EVIDEN

Identity and Access Management

DirX Identity

Provisioning Administration Guide

Version 9.0.2, Edition November 2026



All product names quoted are trademarks or registered trademarks of the manufacturers concerned.

© 2026 Atos Group

All Rights Reserved

Distribution and reproduction not permitted without the consent of Atos Group.

Table of Contents

Copyright ii
Preface 1
DirX Identity Documentation Set 2
Notation Conventions 4
1. Managing Provisioning 6
1.1. Managing Users 6
1.1.1. Maintaining Users 6
1.1.2. Assigning Privileges to Users 7
1.2. Managing User Facets, Personas and Functional Users 7
1.2.1. Managing User Facets 8
1.2.2. Managing Personas 9
1.2.3. Managing Functional Users 9
1.3. Managing Business Objects 9
1.4. Managing Tickets 10
1.5. Managing Policies 10
1.6. Managing Roles, Permissions, and Groups 11
1.7. Managing Request Workflows 11
1.8. Managing Target Systems 11
1.9. Managing Compliance 12
1.9.1. Managing Segregation of Duties (SoD) 12
1.9.2. Managing Certification Campaigns 12
1.9.3. Managing Auditing 13
1.9.4. Managing Status Reporting 13
1.10. Managing Domains 13
1.10.1. Setting up a Domain 14
1.10.2. Managing Domain-Specific Information 14
1.11. Managing the Provisioning System 15
1.12. Managing States 15
1.12.1. User States 16
1.12.2. User Facet and Persona States 20
1.12.3. Functional User States 20
1.12.4. Account States 20
1.12.4.1. Account State Transitions 21
1.12.4.2. Provisioning Workflow State Handling 23
1.12.5. Role States 35
1.12.6. Permission States 36
1.12.7. Group States 36
1.12.7.1. Group State Transitions 36
1.12.7.2. Provisioning Workflow State Handling 37

1.12.8. Assignment States . . . 42
1.12.8.1. Assignment State Transitions . . . 42
1.12.8.2. Provisioning Workflow State Handling . . . 43
1.12.9. How Privilege Resolution Works . . . 46
1.12.9.1. Calculating Account-To-Group Memberships . . . 46
1.12.9.2. Calculating Account States . . . 46
1.12.10. User and Account Lifecycles . . . 47
1.13. Managing Attribute Flow . . . 49
2. Planning for Provisioning . . . 52
2.1. Select the Master Directory or Directories . . . 52
2.2. Identify the Target Systems . . . 52
2.3. Identify the Roles and Permissions . . . 52
2.4. Identify the Target System Groups . . . 53
2.5. Identify the Policies . . . 53
2.6. Define Request Workflows . . . 55
2.7. Decide on Auditing . . . 55
2.8. Identify the Administrators . . . 55
2.9. Set up Important Services . . . 55
2.10. Decide on Customizations . . . 56
3. Managing Users . . . 57
3.1. User Management Pre-Requisites . . . 57
3.2. Managing the Users (DirX Identity is Master) . . . 58
3.2.1. Viewing Users with DirX Identity Manager . . . 58
3.2.2. Adding Users with DirX Identity Manager . . . 59
3.2.2.1. Specifying a User Lifetime . . . 60
3.2.2.2. Creating User Templates . . . 60
3.2.3. Deleting Users with DirX Identity Manager . . . 60
3.2.4. Changing User Attributes with DirX Identity Manager . . . 61
3.2.5. Changing the User Subtree with DirX Identity Manager . . . 62
3.3. Managing the Users (Another Directory is Master) . . . 63
3.4. Assigning Roles to Users . . . 63
3.5. Assigning Permissions to Users . . . 65
3.6. Assigning Groups to Users . . . 66
3.7. Working with User Query Folders . . . 66
3.8. Working with User States . . . 67
3.9. Working with Links at User Entries . . . 67
4. Managing User Facets . . . 70
4.1. Locating User Facets . . . 70
4.2. Working with User Facets . . . 71
4.2.1. Viewing User Facets with DirX Identity Manager . . . 71
4.2.2. Adding User Facets with DirX Identity Manager . . . 71
4.2.2.1. Specifying a User Facet Lifetime . . . 72

4.2.2.2. Working with User Facet Templates . . . 72
4.2.3. Deleting User Facets with DirX Identity Manager . . . 72
4.2.4. Changing User Facet Attributes with DirX Identity Manager . . . 72
4.3. Working with User Facet States . . . 72
4.4. Working with Links at User Facet Entries . . . 73
4.5. Working with User Facet Inheritance . . . 73
4.6. Maintaining User Facets with DirX Identity Workflows . . . 73
5. Managing Personas . . . 74
5.1. Locating Personas . . . 74
5.2. Working with Personas . . . 74
5.2.1. Viewing Personas with DirX Identity Manager . . . 74
5.2.2. Adding Personas with DirX Identity Manager . . . 75
5.2.2.1. Specifying a Persona Lifetime . . . 75
5.2.2.2. Working with Persona Templates . . . 76
5.2.3. Deleting Personas with DirX Identity Manager . . . 76
5.2.4. Changing Persona Attributes with DirX Identity Manager . . . 76
5.3. Working with Persona States . . . 76
5.4. Working with Links at Persona Entries . . . 76
6. Managing Functional Users . . . 77
6.1. Locating Functional Users . . . 77
6.2. Working with Functional Users . . . 77
6.2.1. Viewing Functional Users with DirX Identity Manager . . . 77
6.2.2. Adding Functional Users with DirX Identity Manager . . . 78
6.2.2.1. Specifying a Functional User Lifetime . . . 78
6.2.2.2. Creating Functional User Templates . . . 79
6.2.3. Deleting Functional Users with DirX Identity Manager . . . 79
6.2.4. Changing Functional Users Attributes with DirX Identity Manager . . . 79
6.3. Working with Functional User States . . . 79
6.4. Working with Links at Functional User Entries . . . 80
7. Managing Business Objects . . . 81
7.1. Working with Business Objects . . . 81
7.2. About Business Object Types . . . 82
7.3. Using Business Objects . . . 83
8. Managing Tickets . . . 84
8.1. Understanding Internal Tickets . . . 84
8.1.1. Understanding Internal Ticket Structure . . . 85
8.1.2. Understanding Internal Ticket Folder Structure . . . 87
8.1.3. Understanding Internal Ticket Naming . . . 87
8.1.4. Understanding the Internal Ticket Life-cycle . . . 88
8.1.5. Understanding Internal Ticket States . . . 89
8.1.6. Working with Internal Ticket Objects . . . 90
8.2. About Customer-Specific Tickets . . . 91

9. Managing the Privilege Structure . . . 93
9.1. Working with Privilege Structure Objects . . . 93
9.2. Managing Roles . . . 94
9.2.1. Adding a Role . . . 95
9.2.2. Assigning Role Parameters . . . 95
9.2.3. Creating a Role Hierarchy . . . 97
9.2.4. Assigning Permissions to Roles . . . 97
9.2.5. Saving Changes to a Role . . . 98
9.2.6. Deleting a Role . . . 98
9.2.7. Organizing the Role Subtree . . . 99
9.3. Managing Permissions . . . 100
9.3.1. Adding a Permission . . . 100
9.3.2. Assigning a Matching Rule . . . 100
9.3.3. Managing a Permission's Group Assignments . . . 102
9.3.4. Saving Changes to a Permission . . . 102
9.3.5. Deleting a Permission . . . 102
9.3.6. Organizing the Permission Subtree . . . 103
9.4. Managing Groups . . . 103
9.4.1. Adding a Group . . . 104
9.4.2. Assigning Permission Parameters . . . 104
9.4.3. Viewing a Group's Permission Assignments . . . 104
9.4.4. Saving Changes to a Group . . . 104
9.4.5. Deleting a Group . . . 105
9.5. Working with Privilege Structure Query Folders . . . 105
9.6. Managing the Privilege Resolution Process . . . 106
9.6.1. Handling Role Parameters . . . 107
9.6.2. Handling Permission Parameters . . . 108
10. Managing Policies . . . 109
10.1. Delegated Administration . . . 109
10.1.1. Managing Access Policies . . . 111
10.1.1.1. Understanding Access Policies . . . 112
10.1.1.2. How to Select Subjects . . . 113
10.1.1.2.1. Selecting a List of Direct Users . . . 113
10.1.1.2.2. Selecting a Dynamically Filtered Object List . . . 114
10.1.1.2.3. Selecting a List of Groups by Direct Privilege Assignment . . . 115
10.1.1.2.4. Selecting a List of Groups Controlled by User Attributes . . . 115
10.1.1.2.5. Selecting a List of Groups Controlled by Assignment Parameters . . . 116
10.1.1.2.6. Selecting a List of Groups Controlled by Provisioning Rules . . . 117
10.1.1.2.7. Excluding Specific Persons . . . 118
10.1.1.3. Policies for Object Creation . . . 118
10.1.1.4. Policies for Object Access . . . 118
10.1.1.5. Policies for Grant Operations . . . 119

10.1.1.6. Policies for Approvals . . . 119
10.1.1.7. Policies for Viewing Assignments . . . 122
10.1.1.8. Policies for Hierarchical Role Parameters . . . 123
10.1.1.8.1. Defining the Hierarchical Role Parameter Object . . . 124
10.1.1.8.2. Defining Privileges with Hierarchical Parameters . . . 124
10.1.1.8.3. Defining a Grant Policy for the Role (Role A) . . . 125
10.1.1.9. Policies for Menus . . . 126
10.1.1.10. Guidelines for Access Policy Setup . . . 126
10.1.1.11. Verifying Access Policies . . . 127
10.1.2. Managing Delegations . . . 127
10.1.2.1. Understanding Delegations . . . 128
10.1.2.2. Understanding Access Control . . . 129
10.2. Managing Automatic Provisioning and Consistency Checking . . . 132
10.2.1. Managing Rules . . . 132
10.2.1.1. Understanding Rule Types . . . 133
10.2.1.1.1. Consistency Rules . . . 133
10.2.1.1.2. Provisioning Rules . . . 135
10.2.1.1.3. Validation Rules . . . 136
10.2.1.2. Adding Rules . . . 137
10.2.1.3. Changing Rules . . . 139
10.2.1.4. Deleting Rules . . . 139
10.2.1.5. Organizing Rules . . . 139
10.2.2. Managing Consistency Rule Operations . . . 140
10.2.2.1. About Operation Types . . . 141
10.2.2.1.1. Using Operations Based on Executables . . . 141
10.2.2.1.2. Using Operations Based on Java Methods . . . 142
10.2.2.1.3. Using Operations Based on JavaScripts . . . 144
10.2.2.2. Adding Consistency Rule Operations . . . 146
10.2.2.2.1. Adding an Executable Operation . . . 146
10.2.2.2.2. Adding a Java Operation . . . 147
10.2.2.2.3. Adding a JavaScript Operation . . . 148
10.2.2.3. Changing Operations . . . 148
10.2.2.4. Deleting Operations . . . 149
10.2.2.5. Organizing Operations . . . 149
10.2.3. About Rule Operation . . . 150
10.2.3.1. Running Rules . . . 150
10.2.3.2. How Rules Work . . . 150
10.2.3.2.1. How to Process Provisioning Rules . . . 152
10.2.3.2.2. How to Process Consistency Rules . . . 153
10.2.3.2.3. How to Process Validation Rules . . . 153
10.2.3.3. Optimizing Rule Operation . . . 154
10.2.3.3.1. Separating Rules . . . 154

10.2.3.3.2. Other Hints . . . 155
10.2.3.3.3. Warnings . . . 155
10.3. Managing Object Policies . . . 155
10.3.1. Managing Event Policies . . . 155
10.3.1.1. Adding Event Policies . . . 156
10.3.1.2. Changing Event Policies . . . 157
10.3.1.3. Deleting Event Policies . . . 157
10.3.2. Managing Attribute Policies . . . 157
10.3.2.1. Attribute Policies for Objects . . . 157
10.3.2.2. Adding Attribute Policies . . . 158
10.3.2.3. Changing Attribute Policies . . . 158
10.3.2.4. Deleting Attribute Policies . . . 158
10.3.2.5. Organizing Attribute Policies . . . 158
10.3.3. Managing Deletion Policies . . . 159
10.3.3.1. Adding Delete Policies . . . 159
10.3.3.2. Changing Delete Policies . . . 160
10.3.3.3. Deleting Delete Policies . . . 160
10.4. Managing SoD Policies . . . 161
10.4.1. Adding SoD Policies . . . 161
10.4.2. Changing SoD Policies . . . 161
10.4.3. Deleting SoD Policies . . . 161
10.4.4. Organizing SoD Policies . . . 162
10.5. Managing Password Policies . . . 162
11. Managing Request Workflows . . . 164
11.1. Managing Configuration Objects . . . 165
11.1.1. Adding Configuration Objects . . . 165
11.1.2. Changing Configuration Objects . . . 165
11.1.3. Deleting Configuration Objects . . . 166
11.1.4. Organizing Configuration Objects . . . 166
11.2. Managing Request Workflow Definitions . . . 167
11.2.1. Organizing the Definition Subtree . . . 167
11.2.2. Copying Request Workflow Definitions . . . 167
11.2.3. Adding Request Workflow Definitions . . . 168
11.2.4. Deleting Request Workflow Definitions . . . 168
11.3. Managing Request Workflow Instances . . . 168
11.3.1. Searching Workflow Instances . . . 168
11.3.2. Viewing Workflow Instances . . . 169
11.3.3. Viewing Activity Instances . . . 170
11.3.4. Managing Problems with Workflow Instances . . . 170
11.3.4.1. Changing Participants . . . 171
11.3.4.2. Restarting Workflows . . . 171
11.3.4.2.1. Detailed Procedure . . . 172

11.3.4.2.2. Custom JavaScripts ... 172
12. Managing Target Systems ... 173
12.1. Managing Access Rights ... 173
12.1.1. Managing Access Rights with Group-based Memberships ... 174
12.1.2. Managing Access Rights with Attribute-based Memberships ... 174
12.2. Working with Target System Objects ... 175
12.3. Adding a Target System ... 177
12.3.1. Creating the Target System ... 177
12.3.2. Extending Account and Group Object Descriptions ... 178
12.3.3. Configuring the Initial Load Workflow ... 179
12.3.4. Running the Initial Load Workflow ... 179
12.3.5. Defining Rules to Automate Tasks ... 180
12.3.6. Configuring the Synchronization and Validation Workflows ... 181
12.4. Monitoring Account and Account-Group Assignments ... 181
12.5. Handling the Results of the Validation Workflow ... 182
12.6. Managing Target System Accounts ... 183
12.6.1. Managing Personal Accounts ... 184
12.6.1.1. Adding a Personal Account ... 184
12.6.1.2. Changing the Attributes of an Account ... 185
12.6.1.3. Handling Imported Accounts ... 185
12.6.1.3.1. Association of Accounts to Users ... 186
12.6.1.3.2. Handling Assigned Accounts ... 186
12.6.1.3.3. Handling Unassigned Accounts ... 187
12.6.1.4. Handling Multiple Accounts ... 188
12.6.1.4.1. Full Provisioning ... 189
12.6.1.4.2. Partial Provisioning ... 191
12.6.1.5. Automatic Account Creation ... 192
12.6.1.6. Smooth Account Creation ... 192
12.6.1.7. Managing Accounts Only in the Connected System ... 193
12.6.1.8. Deleting an Account ... 193
12.6.2. Managing Privileged Accounts ... 194
12.6.2.1. Enabling Privileged Account Management ... 195
12.6.2.2. Marking a Privileged Account ... 195
12.6.2.3. Managing Security Tokens ... 196
12.6.2.3.1. Password Management for Privileged Accounts ... 196
12.6.2.3.2. Certificate Management for Privileged Accounts ... 197
12.6.3. Managing Functional Accounts ... 197
12.7. Managing Target System Groups ... 197
12.7.1. Adding a Target System Group ... 198
12.7.2. Changing the Attributes of a Group ... 198
12.7.3. Managing Group Hierarchies ... 198
12.7.4. Managing Imported Groups ... 199

12.7.5. Managing Group Memberships . . . 200
12.7.6. Deleting a Group . . . 201
12.8. Deleting a Target System . . . 201
13. Managing Compliance . . . 203
13.1. Managing Segregation of Duties (SoD) . . . 203
13.1.1. How SoD Works in DirX Identity . . . 204
13.2. Managing Risk . . . 205
13.2.1. How Risk Assessment Works . . . 205
13.2.1.1. Supported Risk Factors . . . 205
13.2.1.2. Computed Values . . . 206
13.2.1.3. How Risk Factors are Computed . . . 206
13.2.1.3.1. Group Memberships . . . 206
13.2.1.3.2. Imported Memberships . . . 207
13.2.1.3.3. Imported Accounts . . . 207
13.2.1.3.4. Privileged Accounts . . . 207
13.2.1.3.5. Sod Violations . . . 207
13.2.1.3.6. Special Handling for Personas, User Facets and Functional Users . . . 207
13.2.2. Configuring Risk Management . . . 207
13.2.2.1. Defining a Risk Policy . . . 207
13.2.2.2. Configuring the Risk Calculation Workflow . . . 208
13.2.2.3. Configuring the Risk Approval Workflow . . . 208
13.2.2.4. Activating Risk Management . . . 209
13.3. Managing Certification Campaigns . . . 209
13.3.1. Enabling the Mail Workflow . . . 209
13.3.2. Setting up the Certification Campaign Controller . . . 209
13.3.3. Creating a Certification Campaign Folder . . . 210
13.4. Managing the Audit Trail . . . 211
13.4.1. How Audit Trail Works . . . 211
13.4.2. Identifying Multiple Causes of Account and Group Changes . . . 213
13.4.2.1. Identifying and Referencing Audit Causes . . . 214
13.4.3. About Audit Trail Format . . . 214
13.4.4. Setting up Audit Trail . . . 218
13.4.4.1. Auditing for the Service Layer . . . 218
13.4.4.2. Auditing for Request Workflows . . . 219
13.4.4.3. Auditing for Real-time Workflows . . . 219
13.4.4.4. Auditing for Web Center Authentications . . . 219
13.4.4.5. Setting up Audit Policies for Custom Object Types . . . 220
13.4.4.6. Setting up Client Signature . . . 220
13.4.4.6.1. Internet Explorer . . . 220
13.4.4.6.2. Firefox . . . 220
13.4.5. Handling Audit Trail Workflows . . . 221
13.4.6. Verifying Signatures . . . 221

13.5. Managing Status Reports . . . 221
13.5.1. Generating Status Reports . . . 221
14. Managing Domains . . . 223
14.1. Configuring a Domain . . . 223
14.1.1. Defining the Administrators . . . 223
14.1.1.1. Pre-configured Administrative Roles and Access Rights . . . 223
14.1.1.2. Domain Super Administrators . . . 225
14.1.2. Customizing Object Descriptions . . . 225
14.1.2.1. Viewing and Editing Template Descriptions . . . 225
14.1.2.2. Upgrading Customized Descriptions . . . 226
14.1.3. Managing Critical Parameters . . . 226
14.1.4. Specifying Permission Parameters . . . 227
14.1.5. Setting up Role Parameters . . . 228
14.1.6. Setting Domain Parameters . . . 229
14.1.7. Setting up the Report Templates . . . 229
14.1.8. Relationship to the Connectivity Domain . . . 230
14.2. Using Object Collections . . . 232
14.3. Monitoring the Workflows . . . 233
15. Managing the Provisioning System . . . 234
15.1. Logging in To DirX Identity Manager . . . 234
15.2. Setting LDAP Client Limits . . . 235
15.2.1. Setting Limits in the Bind Profile . . . 235
15.2.2. Setting Limits in the DirX Identity Manager INI File . . . 236
15.2.3. Setting Limits in the DirX Identity Provisioning Domain . . . 236
15.3. Managing Consistency . . . 236
15.3.1. Using the Consistency Check Workflow . . . 236
15.3.2. Using the Mark Affected Users Workflow . . . 237
15.3.3. Using the User Resolution Workflow . . . 237
15.3.4. Setting up Object Cleanup . . . 238
15.3.4.1. Consistency Checking . . . 238
15.3.4.2. Mark Affected Users . . . 238
15.3.4.3. Object Cleanup with Deletion Consistency Rules . . . 238
15.4. Monitoring DirX Identity Provisioning . . . 239
15.4.1. Detecting and Handling Inconsistencies in DirX Identity Provisioning . . . 239
15.4.1.1. How DirX Identity Provisioning Detects Inconsistencies . . . 239
15.4.1.2. How DirX Identity Manager Detects Inconsistencies . . . 240
15.4.1.3. How the Domain Consistency Workflow Detects Inconsistencies . . . 240
15.4.1.4. How the Consistency and Validation Workflows Detect Inconsistencies . . . 241
15.4.1.5. Using Status Reports to Detect Inconsistencies . . . 241
15.4.1.6. Using Query Folders to Detect Inconsistencies . . . 241
15.4.1.7. Provisioning Object Attributes that Identify Inconsistencies . . . 242
15.4.1.8. Handling Detected Inconsistencies . . . 242

15.4.2. DirX Identity Provisioning Logging . . . 243
15.4.2.1. DirX Identity Manager Logging . . . 243
15.4.2.2. DirX Identity Service Agent Logging . . . 243
15.5. Tuning the Provisioning System . . . 243
15.5.1. Increasing the Number of Resolution Adapter Threads . . . 244
15.5.2. Tuning the User Edit Operation . . . 244
15.5.2.1. Changing the Customer-specific User.xml Object Description . . . 244
15.5.3. Tuning the Tree Operation . . . 245
15.5.4. Managing Long Delays . . . 246
15.5.5. Ignoring Nested Group Searches . . . 247
15.5.6. Activating XXL Groups . . . 247
Appendix A: Context-Sensitive Help . . . 248
A.1. Users View . . . 248
A.1.1. Users . . . 248
A.1.1.1. User - General Properties . . . 249
A.1.1.2. User - Relationships . . . 250
A.1.1.3. User - Operational . . . 251
A.1.1.4. User - Communication . . . 253
A.1.1.5. User - Authentication . . . 254
A.1.1.6. User - Organization . . . 256
A.1.1.7. User - Location . . . 257
A.1.1.8. User - Context . . . 258
A.1.1.9. User - Assigned Roles . . . 259
A.1.1.10. User - Assigned Permissions . . . 260
A.1.1.11. User - Assigned Groups . . . 262
A.1.1.12. User - Accounts . . . 264
A.1.1.13. User - Orders . . . 265
A.1.1.14. User - SoD Exceptions . . . 267
A.1.1.15. User - Risk Parameters . . . 268
A.1.2. User Facets . . . 269
A.1.2.1. User Facet - General Properties . . . 269
A.1.2.2. User Facet - Relationships . . . 271
A.1.2.3. User Facet - Operational Properties . . . 271
A.1.2.4. User Facet - Communication . . . 274
A.1.2.5. User Facet - Authentication . . . 275
A.1.2.6. User Facet - Organization . . . 276
A.1.2.7. User Facet - Location . . . 276
A.1.2.8. User Facet - Context . . . 277
A.1.2.9. User Facet - Assigned Roles . . . 278
A.1.2.10. User Facet - Assigned Permissions . . . 279
A.1.2.11. User Facet - Assigned Groups . . . 281
A.1.2.12. User Facet - Orders . . . 283

A.1.2.13. User Facet - SoD Exceptions . . . 284
A.1.2.14. User Facet - Risk Parameters . . . 285
A.1.3. Persona . . . 286
A.1.3.1. Persona - General Properties . . . 286
A.1.3.2. Persona - Relationships . . . 288
A.1.3.3. Persona - Operational Properties . . . 289
A.1.3.4. Persona - Communication . . . 291
A.1.3.5. Persona - Authentication . . . 292
A.1.3.6. Persona - Organization . . . 293
A.1.3.7. Persona - Location . . . 294
A.1.3.8. Persona - Context . . . 295
A.1.3.9. Persona - Assigned Roles . . . 295
A.1.3.10. Persona - Assigned Permissions . . . 297
A.1.3.11. Persona - Assigned Groups . . . 299
A.1.3.12. Persona - Accounts . . . 301
A.1.3.13. Persona - Orders . . . 302
A.1.3.14. Persona - SoD Exceptions . . . 303
A.1.3.15. Persona - Risk Parameters . . . 305
A.1.4. Functional Users . . . 306
A.1.4.1. Functional User - General Properties . . . 306
A.1.4.2. Functional User - Relationships . . . 307
A.1.4.3. Functional User - Operational Properties . . . 308
A.1.4.4. Functional User - Communication . . . 310
A.1.4.5. Functional User - Authentication . . . 311
A.1.4.6. Functional User - Organization . . . 312
A.1.4.7. Functional User - Location . . . 313
A.1.4.8. Functional User - Context . . . 314
A.1.4.9. Functional User - Assigned Roles . . . 315
A.1.4.10. Functional User - Assigned Permissions . . . 317
A.1.4.11. Functional User - Assigned Groups . . . 318
A.1.4.12. Functional User - Accounts . . . 320
A.1.4.13. Functional User - Orders . . . 322
A.1.4.14. Functional User - SoD Exceptions . . . 323
A.1.4.15. Functional User - Risk Parameters . . . 324
A.1.5. Assignment - General . . . 325
A.1.6. Assignment - RoleParams . . . 326
A.1.7. Country - General Properties . . . 326
A.1.8. Domain Component - General Properties . . . 327
A.1.9. Locality - General Properties . . . 327
A.1.10. Organization - General Properties . . . 327
A.1.11. Organizational Unit - General Properties . . . 328
A.2. Business Objects View . . . 328

A.2.1. Business Objects . . . 328
A.2.2. Context . . . 328
A.2.3. Cost Unit . . . 329
A.2.4. Country . . . 330
A.2.5. Location . . . 330
A.2.6. Organization . . . 331
A.2.7. Organizational Unit . . . 332
A.2.8. Project . . . 332
A.3. Ticket View . . . 333
A.3.1. Ticket Folder . . . 333
A.3.2. Ticket - General Properties . . . 333
A.3.3. Ticket - Status Information . . . 334
A.3.4. Ticket - Object . . . 334
A.3.5. Ticket - Assignments . . . 335
A.4. Privileges View . . . 335
A.4.1. Roles . . . 336
A.4.1.1. Role - General Properties . . . 336
A.4.1.2. Role - Approval . . . 337
A.4.1.3. Role - Re-approval . . . 339
A.4.1.4. Role - Details . . . 340
A.4.1.5. Role - Role Parameters . . . 341
A.4.1.6. Role - Assigned Junior Roles . . . 342
A.4.1.7. Role - Assigned Permissions . . . 343
A.4.1.8. Role - Senior Roles . . . 343
A.4.1.9. Role - Users . . . 344
A.4.1.10. Role - Provisioning Rules . . . 344
A.4.2. Permissions . . . 345
A.4.2.1. Permission - General Properties . . . 345
A.4.2.2. Permission - Approval . . . 347
A.4.2.3. Permission - Re-approval . . . 348
A.4.2.4. Permission - Match Rules . . . 349
A.4.2.5. Permission - Assigned Groups . . . 350
A.4.2.6. Permission - Roles . . . 350
A.4.2.7. Permission - Users . . . 351
A.4.2.8. Permission - Provisioning Rules . . . 351
A.4.3. Groups . . . 352
A.4.3.1. Group - General Properties . . . 353
A.4.3.2. Group - Approval . . . 354
A.4.3.3. Group - Re-approval . . . 355
A.4.3.4. Group - *Target System Specific* . . . 356
A.4.3.5. Group - Operational . . . 359
A.4.3.6. Group - Permission Parameters . . . 360

A.4.3.7. Group - Permissions . . . 361
A.4.3.8. Group - Privileged Members . . . 361
A.4.3.9. Group - Members . . . 362
A.4.3.10. Group - Remote Members . . . 363
A.4.3.11. Group - Member of Group . . . 363
A.4.3.12. Group - Obligations . . . 364
A.4.3.13. Group - Provisioning Rules . . . 365
A.5. Policies View . . . 365
A.5.1. Policies . . . 366
A.5.1.1. Access Policies . . . 366
A.5.1.1.1. Access Policy - General . . . 366
A.5.1.1.2. Access Policy - Subjects . . . 368
A.5.1.1.3. Access Policy - Resources . . . 369
A.5.1.1.4. Access Policy - Attributes Read . . . 369
A.5.1.1.5. Access Policy - Attributes Modify . . . 370
A.5.1.1.6. Access Policy - Rules . . . 370
A.5.1.2. Attribute Policies . . . 372
A.5.1.2.1. Attribute Policies Folder . . . 372
A.5.1.2.2. Attribute Policy - General . . . 372
A.5.1.2.3. Attribute Policy - Configuration . . . 373
A.5.1.3. Delete Policies . . . 373
A.5.1.3.1. Delete Policies Folder . . . 374
A.5.1.3.2. Delete Policy - General . . . 374
A.5.1.3.3. Delete Policy - Configuration . . . 375
A.5.1.4. Event Policies . . . 375
A.5.1.4.1. Event Policies Folder . . . 375
A.5.1.4.2. Event Policy - General . . . 376
A.5.1.4.3. Event Policy - Configuration . . . 376
A.5.1.5. Risk Policies . . . 377
A.5.1.5.1. Risk Policy - General . . . 377
A.5.1.6. Rules . . . 378
A.5.1.6.1. Consistency Rule - General Properties . . . 378
A.5.1.6.2. Consistency Rule - Filter . . . 379
A.5.1.6.3. Provisioning Rule - General Properties . . . 380
A.5.1.6.4. Provisioning Rule - Filter . . . 381
A.5.1.6.5. Provisioning Rule - Privileges . . . 382
A.5.1.6.6. Validation Rule - General Properties . . . 384
A.5.1.6.7. Validation Rule - Accounts . . . 385
A.5.1.6.8. Validation Rule - Groups . . . 385
A.5.1.7. Operations . . . 386
A.5.1.7.1. Executables - General Properties . . . 386
A.5.1.7.2. Java Class - General Properties . . . 387

A.5.1.7.3. Java Method - General Properties . . . 387
A.5.1.7.4. JavaScript - General Properties . . . 388
A.5.1.7.5. JavaScript - Implementation . . . 389
A.5.1.8. Password Policies . . . 389
A.5.1.9. SoD Policies . . . 391
A.5.1.9.1. SoD Policies Folder . . . 392
A.5.1.9.2. SoD Policy - General . . . 392
A.5.1.9.3. SoD Exception . . . 393
A.5.2. Delegations Folder . . . 393
A.5.2.1. Delegations . . . 393
A.5.2.2. Access Rights . . . 394
A.6. Certification Campaign View . . . 395
A.6.1. Campaign . . . 395
A.6.1.1. Campaign Folder . . . 395
A.6.1.2. Campaign - General Properties . . . 395
A.6.1.3. Campaign - Status . . . 396
A.6.1.4. Campaign - User Filter . . . 398
A.6.1.5. Campaign - Privilege Filter . . . 398
A.6.1.6. Campaign - User Hooks . . . 399
A.6.2. Notifications . . . 400
A.6.2.1. Notification - General . . . 400
A.6.2.2. Notification - Mail Body . . . 401
A.6.3. Certifications . . . 402
A.6.3.1. Certification - General . . . 402
A.6.3.2. Certification - Status . . . 402
A.6.3.3. Certification - Approvers . . . 403
A.6.3.4. Certification - Attributed Assignments . . . 404
A.6.3.5. Certification - Simple Assignments . . . 404
A.7. Request Workflows View . . . 405
A.7.1. Configuration . . . 405
A.7.1.1. Activity Types . . . 405
A.7.1.2. Common Activities . . . 405
A.7.1.3. Services Folder . . . 406
A.7.1.4. Service . . . 406
A.7.1.5. Certificate Service . . . 407
A.7.1.6. SMTP Service . . . 408
A.7.1.7. SMS Gateway . . . 410
A.7.2. Definitions . . . 411
A.7.2.1. Workflow - General . . . 411
A.7.2.2. Workflow - Workflow . . . 412
A.7.2.3. Workflow - When Applicable . . . 413
A.7.2.4. Activity - General . . . 414

A.7.2.5. Activity - Activity . . . 415
A.7.2.6. Activity - Parameters . . . 416
A.7.2.7. Activity - Participants . . . 422
A.7.2.8. Activity - Participants Filter and Constraints . . . 425
A.7.2.9. Activity - Notifications . . . 426
A.7.2.10. Activity - Escalations . . . 427
A.7.2.11. Notifications . . . 427
A.7.2.12. SendMail . . . 428
A.7.2.13. Send Text Message . . . 429
A.7.3. Monitor . . . 429
A.7.3.1. Request Workflow Instance . . . 429
A.7.3.2. Request Workflow Activity Instance . . . 432
A.8. Target Systems View . . . 433
A.8.1. Target Systems . . . 434
A.8.1.1. Accounts and Groups - General . . . 434
A.8.1.2. Accounts and Groups - Attributes for the Connected System . . . 434
A.8.1.3. Target System - General Properties . . . 435
A.8.1.4. Target System - Advanced Properties . . . 436
A.8.1.5. Target System - Timing Properties . . . 440
A.8.1.6. Target System - Workflow Configuration . . . 441
A.8.1.7. Target System - Connector Configuration . . . 441
A.8.1.8. Target System - Environment Properties . . . 442
A.8.1.9. Target System - Options . . . 442
A.8.1.10. Account - General Properties . . . 445
A.8.1.11. Account - *Target System Specific* . . . 446
A.8.1.12. Account - Operational . . . 448
A.8.1.13. Account - Authentication . . . 449
A.8.1.14. Account - Member Of . . . 451
A.8.1.15. Account - UsedBy . . . 452
A.8.2. Configuration . . . 452
A.8.2.1. JavaScripts . . . 453
A.8.2.1.1. JavaScript - General Properties . . . 453
A.8.2.1.2. JavaScript - Content . . . 453
A.8.2.2. Object Description . . . 453
A.8.2.2.1. Object Description - General Properties . . . 454
A.8.2.2.2. Object Description - Content . . . 454
A.8.2.3. Obligation . . . 454
A.8.2.3.1. Obligation - General Properties . . . 454
A.8.2.3.2. Obligation - Obligations . . . 455
A.8.2.4. Proposal List . . . 457
A.8.2.4.1. Proposal List - General Properties . . . 457
A.8.2.4.2. Proposal List - Item List . . . 459

A.8.2.5. Proposal List with Language Support . . . 459
A.8.2.5.1. Proposal List with Language Support - General Properties . . . 459
A.8.2.6. Report . . . 460
A.8.2.6.1. Report - General Properties . . . 460
A.8.2.6.2. Report - Description . . . 460
A.8.2.6.3. Report - Format . . . 461
A.8.3. Groups . . . 461
A.8.3.1. Group - General Properties . . . 462
A.8.3.2. Group - Approval . . . 463
A.8.3.3. Group - Re-approval . . . 465
A.8.3.4. Group - *Target System Specific* . . . 466
A.8.3.5. Group - Operational . . . 469
A.8.3.6. Group - Permission Parameters . . . 470
A.8.3.7. Group - Permissions . . . 471
A.8.3.8. Group - Privileged Members . . . 472
A.8.3.9. Group - Members . . . 472
A.8.3.10. Group - Remote Members . . . 473
A.8.3.11. Group - Member of Group . . . 474
A.8.3.12. Group - Obligations . . . 475
A.8.3.13. Group - Provisioning Rules . . . 476
A.8.4. Services . . . 477
A.8.4.1. Policy Agent Optimization . . . 477
A.9. Auditing View . . . 477
A.9.1. Audit Trail . . . 477
A.9.1.1. Audit Policies . . . 478
A.9.1.2. Audit Policy . . . 478
A.9.2. Status Reports . . . 480
A.10. Domain Configuration View . . . 480
A.10.1. Domain - Properties . . . 481
A.10.2. Collection . . . 491
A.10.3. Collections . . . 492
A.10.4. Collection Rule . . . 493
A.10.5. Collection Rules . . . 496
A.10.6. Collection Rule Syntax . . . 496
A.10.7. Role Parameter . . . 499
A.10.7.1. Role Parameters . . . 503
A.11. Miscellaneous . . . 503
A.11.1. Activity Editor . . . 504
A.11.1.1. Viewing . . . 504
A.11.1.2. Editing . . . 504
A.11.1.3. Other Features . . . 505
A.11.2. Assignment Editor . . . 505

A.11.3. Multi-Value Editor . . . 506
A.11.4. Query Folder - General Properties . . . 506
A.11.5. Query Folder - Filter . . . 507
A.11.6. Text Editor . . . 508
A.11.7. Run Report . . . 509
A.11.8. Workflow Editor . . . 510
A.11.8.1. Viewing . . . 510
A.11.8.2. Editing . . . 510
A.11.8.3. Other Features . . . 511
A.12. Target System Wizard - Overview . . . 512
A.12.1. Target System Wizard - Type Selection . . . 513
A.12.2. Target System Wizard - General . . . 513
A.12.3. Target System Wizard - Advanced . . . 514
A.12.4. Target System Wizard - Timing . . . 514
A.12.5. Target System Wizard - Options . . . 515
A.12.6. Target System Wizard - Connectivity Configuration . . . 515
A.12.7. Target System Wizard - Identity Store . . . 516
A.12.8. Target System Wizard - Connectivity Scenario . . . 516
A.12.9. Target System Wizard - Associated Connected Directory . . . 517
A.12.10. Target System Wizard - Connected Directory Configuration . . . 518
A.12.11. Target System Wizard - Provisioning Workflows . . . 520
A.13. Message Item . . . 520
A.13.1. Message Item Folder . . . 521
Legal Remarks . . . 523

Preface

This manual provides an introduction to DirX Identity Provisioning Administration. It consists of the following sections:

- "Managing Provisioning" describes how to manage the Provisioning System.
- "Planning for Provisioning" provides a planning checklist of the issues that you must take into account when deciding how to deploy DirX Identity Provisioning in your environment.
- "Managing Users" describes how to manage DirX Identity Provisioning users.
- "Managing User Facets" describes how to manage DirX Identity Provisioning user facets.
- "Managing Personas" describes how to manage DirX Identity Provisioning personas.
- "Managing Functional Users" describes how to manage DirX Identity Provisioning functional users.
- "Managing Business Objects" describes how to manage business objects.
- "Managing Tickets" provides information how to manage tickets.
- "Managing the Privilege Structure" describes how to use DirX Identity Manager to create and manage the objects associated with a privilege structure.
- "Managing Policies" describes how to use DirX Identity Manager to create and manage the objects associated with a policy structure.
- "Managing Request Workflows" describes how to manage request workflows.
- "Managing Target Systems" describes how to manage target systems.
- "Managing Compliance" describes how to manage auditing and compliance.
- "Managing Domains" describes how to manage domains.
- "Managing the Provisioning System" describes how to manage the DirX Identity Provisioning system.
- "Context-Sensitive Help" presents the context-sensitive help topics that are provided with DirX Identity Provisioning.

DirX Identity Documentation Set

Version: 9.0.2 | Date: 2026-11-02 | Timestamp: 13:57:36 UTC

The DirX Identity document set consists of the following manuals:

- *DirX Identity Introduction*. Use this book to obtain a description of DirX Identity architecture and components.
- *DirX Identity Release Notes*. Use this book to understand the features and limitations of the current release. This document is shipped with the DirX Identity installation as the file **release-notes.pdf**.
- *DirX Identity History of Changes*. Use this book to understand the features of previous releases. This document is shipped with the DirX Identity installation as the file **history-of-changes.pdf**.
- *DirX Identity Tutorial*. Use this book to get familiar quickly with your DirX Identity installation.
- *DirX Identity Provisioning Administration Guide*. Use this book to obtain a description of DirX Identity provisioning architecture and components and to understand the basic tasks of DirX Identity provisioning administration using DirX Identity Manager.
- *DirX Identity Connectivity Administration Guide*. Use this book to obtain a description of DirX Identity connectivity architecture and components and to understand the basic tasks of DirX Identity connectivity administration using DirX Identity Manager.
- *DirX Identity User Interfaces Guide*. Use this book to obtain a description of the user interfaces provided with DirX Identity.
- *DirX Identity Application Development Guide*. Use this book to obtain information how to extend DirX Identity and to use the default applications.
- *DirX Identity Customization Guide*. Use this book to customize your DirX Identity environment.
- *DirX Identity Integration Framework*. Use this book to understand the DirX Identity framework and to obtain a description how to extend DirX Identity.
- *DirX Identity Web Center Reference*. Use this book to obtain reference information about the DirX Identity Web Center.
- *DirX Identity Web Center Customization Guide*. Use this book to obtain information how to customize the DirX Identity Web Center.
- *DirX Identity Resolution Service Setup Guide*. Use this book to set up the DirX Identity Resolution Service as a standalone service.
- *DirX Identity Meta Controller Reference*. Use this book to obtain reference information about the DirX Identity meta controller and its associated command-line programs and files.
- *DirX Identity Connectivity Reference*. Use this book to obtain reference information about the DirX Identity agent programs, scripts, and files.
- *DirX Identity Troubleshooting Guide*. Use this book to track down and solve problems in your DirX Identity installation.

- *DirX Identity Installation Guide*. Use this book to install DirX Identity.
- *DirX Identity Migration Guide*. Use this book to migrate from previous versions.
- *DirX Identity REST Service Guide*. Use this book to migrate from previous versions.

Notation Conventions

Boldface type
In command syntax, bold words and characters represent commands or keywords that must be entered exactly as shown.

In examples, bold words and characters represent user input.

Italic type
In command syntax, italic words and characters represent placeholders for information that you must supply.

[ ]
In command syntax, square braces enclose optional items.

{ }
In command syntax, braces enclose a list from which you must choose one item.

In Tcl syntax, you must actually type in the braces, which will appear in boldface type.

|
In command syntax, the vertical bar separates items in a list of choices.

...
In command syntax, ellipses indicate that the previous item can be repeated.

userID_home_directory
The exact name of the home directory. The default home directory is the home directory of the specified UNIX user, who is logged in on UNIX systems. In this manual, the home pathname is represented by the notation *userID_home_directory*.

install_path
The exact name of the root of the directory where DirX Identity programs and files are installed. The default installation directory is *userID_home_directory*/**DirX Identity** on UNIX systems and **C:\Program Files\DirX\Identity** on Windows systems. During installation the installation directory can be specified. In this manual, the installation-specific portion of pathnames is represented by the notation *install_path*.

dirx_install_path
The exact name of the root of the directory where DirX programs and files are installed. The default installation directory is *userID_home_directory*/**DirX** on UNIX systems and **C:\Program Files\DirX** on Windows systems. During installation the installation directory can be specified. In this manual, the installation-specific portion of pathname is represented by the notation *dirx_install_path*.

dxi_java_home
The exact name of the root directory of the Java environment for DirX Identity. This location is specified while installing the product. For details see the sections "Installation" and "The Java for DirX Identity".

tmp_path

The exact name of the tmp directory. The default tmp directory is /tmp on UNIX systems. In this manual, the tmp pathname is represented by the notation *tmp_path*.

tomcat_install_path
The exact name of the root of the directory where Apache Tomcat programs and files are installed. This location is defined during product installation.

mount_point
The mount point for DVD device (for example, **/cdrom/cdrom0**).

1. Managing Provisioning

Managing the Provisioning system consists of the following tasks:

- Managing users and user templates
- Managing user facets, personas and functional users
- Managing business objects
- Managing tickets
- Managing privileges (roles, permissions, and groups)
- Managing policies (access policies, rules and operations, delegations)
- Managing request workflows
- Managing target systems
- Managing audit and compliance
- Managing certification campaigns
- Managing domains
- Configuring and managing the Provisioning Configuration
- Managing states
- Managing the attribute flow

This chapter gives a brief overview of each of these tasks, with more detailed information provided in the individual chapters on these tasks.

1.1. Managing Users

User management consists of two main tasks:

- Maintaining an accurate and up-to-date directory of the users to be administered with DirX Identity
- Assigning privileges to users (controlling user rights) either by hand or automatically with policies.

Administrators carry out these tasks manually with the DirX Identity Manager or automatically with one of the workflows of the Connectivity view.Alternatively the Web-based interface Web Center can be used to assign privileges to users.

1.1.1. Maintaining Users

User maintenance tasks include:

- Adding users, deleting users, and changing the attributes of users to be administered with DirX Identity, especially the attributes associated with user lifetime and deactivation periods (if DirX Identity masters the user data)
- Synchronizing the DirX Identity user with a corporate directory master (if a corporate

directory masters the user data)

- Creating and maintaining user templates (if DirX Identity masters the user data)

1.1.2. Assigning Privileges to Users

Assigning privileges to users is the main administrative function in DirX Identity.In the DirX Identity Provisioning Configuration, the administrator controls a user's rights by assigning one or more privileges to the user.DirX Identity then resolves the assigned privileges to the appropriate groups in the target systems (privilege resolution).

Next, DirX Identity Connectivity distributes the user-group relationships to the appropriate target systems (privilege distribution).

The administrator can also assign one or more privileges to a user template.In this case, DirX Identity stores the user-privilege assignment in its identity store, but does not perform privilege resolution and distribution until the administrator uses the template to create a real user.

1.2. Managing User Facets, Personas and Functional Users

User facets, personas and functional users are special representations of users.A **user** represents a user in an identity management system.It is the main entry for this user and typically represents the personal accounts for this user.

A **user facet** is an alternative representation of a user that reflects the user's different positions in an organization, for example, a student who works as a teaching assistant and as a tutor.The "tutor" and "teaching assistant" user facets model the privileges that derive from his capacity as a tutor and a teaching assistant, while the "student" user facet models the privileges that derive from the user's position as a student.The main user, meanwhile, models all of these privileges with the resulting accounts.The lifetime of the user object typically comprises the lifetime of all associated user facet objects, and the lifetime of a user facet object can be shorter than the user lifetime.The user facet object offers a way to model different functions for a user within a single organizational unit.User facets allow you to model multiple privilege profiles that share the same accounts as the user.A user facet is related to a user via the owner attribute.In contrast to a functional user, the user facet object normally changes its status with the corresponding user object and is not re-assigned to another user.

A **persona** is an alternative representation of a user that reflects the user's different functions in the company, for example, an administrator or a project manager.The accounts and the access rights for each representation may be quite different and auditing should be able to distinguish between the different representations for the user.The lifetime of the user object typically comprises the lifetime of all associated persona objects, and the lifetime of a persona object can be shorter than the user lifetime.The persona object offers a way to model different functions for a user that works in the same company but for different organizational units.A persona is related to a user via the owner attribute.In contrast to a functional user, the persona object normally changes its status with the corresponding user object and is not re-assigned to another user.

A **functional user** is a resource that is assigned to an identity.Examples are a global or group mailbox, a physical room with a phone or a working student entry.The user either manages these resources or is responsible for them.A functional user is related to a user via the sponsor attribute.In contrast to a persona, the functional user can survive the user and must be re-assigned to another sponsor (user) if the current user is deleted or is no longer responsible for the resource.

The following figure illustrates the relationships between user, user facet persona and functional user objects:

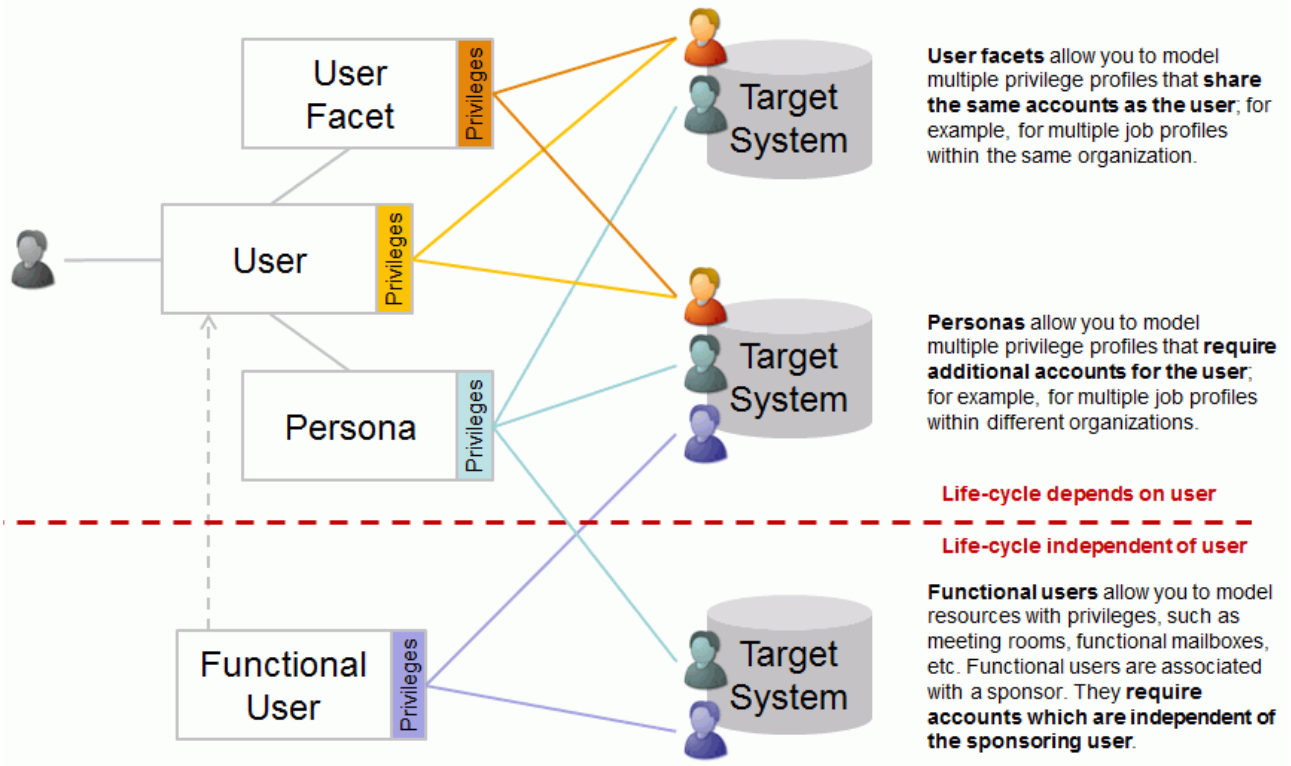


Figure 1. User, User Facet, Persona and Functional User Relationships

1.2.1. Managing User Facets

Since a user facet is a special representation of a user, maintaining user facets is closely related to maintaining users. User facet maintenance tasks include:

- Adding user facets, deleting user facets, and changing the attributes of user facets to be administered with DirX Identity, especially the attributes associated with user facet lifetime and deactivation periods (if DirX Identity masters the user facet data).
- Synchronizing the DirX Identity user facet with a corporate directory master, if the concept of different user representations is also supported by the corporate directory (if a corporate directory masters the user facet data). Otherwise, only the user object is synchronized, and the user facet is maintained in DirX Identity.
- Assigning privileges to user facets, which you perform with same procedures and under the same rules as for user privilege assignment.

1.2.2. Managing Personas

Since a persona is a special representation of a user, maintaining personas is closely related to maintaining users. Persona maintenance tasks include:

- Adding personas, deleting personas, and changing the attributes of personas to be administered with DirX Identity, especially the attributes associated with persona lifetime and deactivation periods (if DirX Identity masters the persona data).
- Synchronizing the DirX Identity persona with a corporate directory master, if the concept of different user representations is also supported by the corporate directory (if a corporate directory masters the persona data). Otherwise, only the user object is synchronized, and the persona is maintained in DirX Identity.
- Assigning privileges to personas, which you perform in the same way under the same rules as privilege assignment to users.

1.2.3. Managing Functional Users

Since a functional user is a special representation of a user, maintaining functional users is closely related to maintaining users.Functional user maintenance tasks include:

- Adding functional users, deleting functional users, and changing the attributes of functional users to be administered with DirX Identity.
- Maintaining the sponsor if the related user for the functional user changes.
- Assigning privileges to functional users, which you perform with same procedures and under the same rules as for user privilege assignment.

1.3. Managing Business Objects

You can use business objects to keep common information in a centrally located individual object rather than in individual user entries, where it can become inconsistent over time.Examples of business objects include organizations, organizational units, countries, locations, and projects.Business objects have many uses:

- You can use a business object to assign roles, permissions and groups referenced from the business object automatically to all linked users.
- You can use business object data such as "address" or "street" in linked user attributes and master this data at the business object.
- You can use business objects as sources for role parameters and proposal lists.

Business object management consists of the following tasks:

- Defining a set of business objects that help you to maintain an accurate and up-to-date directory of central data that can be used by user entries.DirX Identity delivers a set of default business objects that you can customize to your needs, and you can define your own types of business objects as necessary.
- Setting up event-based maintenance workflows that inherit privileges from business objects to users.

- Setting up event-based maintenance workflows that propagate attribute changes from business objects to users.
- (Optional) Setting up workflows to synchronize data from external data sources.

1.4. Managing Tickets

By default, DirX Identity processes updates to its identity management data immediately.For example, when an administrator changes a user's department attribute, the change is applied right away (or when approval is obtained).In some cases, however, administrators may want to delay the application of a change and schedule it for a future date; for example, to process an employee transfer from one department to another.DirX Identity provides a built-in ticket system that allows administrators to specify the changes to an identity management object that should be made at a specified date in the future.The changes are then deferred and saved as temporary information in a ticket until the specified date arrives, at which point the DirX Identity Services layer processes them automatically.DirX Identity also allows for the integration of external ticket systems and provides other means for service management integration.

Ticket management tasks generally include:

- Creating tickets in DirX Identity Web Center or Manager to schedule object changes
- Viewing tickets to check their status and content
- Making changes to tickets to correct errors in ticket data

1.5. Managing Policies

DirX Identity supports different kinds of policies:

- Access policies, to control the access to objects in the Provisioning Configuration and for use for delegation and approval workflows.
- Object policies for object related events, modification and deletion.
- Rules and operations, to automate consistency checks and perform automatic provisioning.
- Delegations, to propagate access rights from one user to another one (optionally with a time restriction).

Policy management for access management and delegation consists of these tasks:

- Adding access policies, deleting access policies or maintaining access policies
- Defining which access policies can be delegated and which ones cannot be delegated.
- Creating and maintaining delegations via the DirX Identity Web Center

Policy management for consistency checking and automatic provisioning consists of these tasks:

- Adding attribute policies, deleting attribute policies or maintaining attribute policies

- Adding event policies, deleting event policies or maintaining event policies
- Adding delete policies, deleting delete policies or maintaining delete policies
- Setting up event-based and request workflows that are triggered by attribute, event and delete policies
- Adding rules, deleting rules or maintaining rules
- Adding operations, deleting operations or maintaining operations
- Running rules against the database to check correctness
- Setting up Tcl-oriented workflows to run these rules regularly.

1.6. Managing Roles, Permissions, and Groups

Managing privileges (roles, permissions, and groups) involves setting up and maintaining the privilege structure; that is, the roles and permissions and the role-permission-group relationships.Role management tasks include:

- Setting up and maintaining the role structure
- Creating, changing, and deleting roles
- Building the role hierarchy (role-to-role relationships)
- Defining role parameters and their application to roles
- Creating, changing, and deleting permissions
- Defining permission parameters and their application to permissions
- Creating, changing, and deleting groups

Administrators carry out these tasks with the DirX Identity Manager.

1.7. Managing Request Workflows

Request workflows permit you to manage the life-cycle of objects in a controlled way including the necessary approval and certification steps.Management tasks include:

- Configuring central parameters and services (for example, setting up e-mail forms for approval or certification requests)
- Setting up and maintaining request workflow templates
- Monitoring and maintaining request workflow instances

1.8. Managing Target Systems

Target system management tasks include:

- Configuring the workflows that:
- Perform the initial load of existing target system data - accounts, groups, and their relationships - into the Provisioning Configuration (this is a set-up task vs. a day-to-day

task)

- Load the Provisioning Configuration data into the target systems to synchronize them (either as event-based or scheduled privilege distribution)
- Periodically reconcile the target system data with the Provisioning Configuration
- Configuring name space rules, for example, naming rules for accounts or passwords
- Adding and deleting target systems to and from the Provisioning Configuration



Provisioning workflows are not needed for managing virtual target systems because the target system data is contained in the Provisioning Configuration.

Administrators use DirX Identity Manager to configure the initial load, event-based privilege distribution, and scheduled validation workflows.

1.9. Managing Compliance

Compliance services provided in DirX Identity include:

- Segregation of Duties (SoD)
- Certification campaigns
- Audit trail
- Status reporting

1.9.1. Managing Segregation of Duties (SoD)

Segregation of duties - often called separation of duties or SoD - is a type of policy that can be applied to user-privilege assignments that constitute conflicts of interest and a mechanism for checking whether the privilege assignments granted to a user comply with that policy. The SoD policies define the conflicting privileges, while the SoD mechanism checks the user-privilege assignments for compliance with the policy and reports any discovered violations.

The section "Managing SoD Policies" describes the management tasks associated with SoD policies, while the section "Managing SoD" describes the internal and external features that DirX Identity provides for SoD policy definition and for SoD violation checking and resolution.

1.9.2. Managing Certification Campaigns

Certification campaigns let you regularly certify user-role assignments. A campaign manager uses Identity Manager to set up a campaign. This task mainly involves defining the set of users to be certified and the start and end of the campaign. DirX Identity's Certification Campaign Controller workflow automatically starts and monitors the campaign.

When the start date is reached, the Certification Campaign Controller workflow creates a

certification task for every user to be certified, determines the approver, and informs the approver via email. The approvers must either accept or reject each role assignment.

When the end date of the campaign is reached, the Certification Campaign Controller workflow removes all rejected assignments and notifies the affected users. The campaign manager can produce a report on the whole campaign.

The section "Managing Certification Campaigns" in the chapter "Managing Compliance" provides more information about this process, while the DirX Identity Use Case Document *User Certification Campaigns* gives a complete description.

1.9.3. Managing Auditing

DirX Identity's audit trail mechanism can track all relevant identity management events, recording information such as the date/time the event occurred, the identity that initiated it, the users who approved it, and whether it was carried out by hand or automatically by a policy.

Auditing management tasks include:

- Configuring domain-wide audit trail parameters.
- Setting up audit policies for each object to be audited. DirX Identity supplies a set of pre-configured audit policies, and you can also define your own audit policies that address your company's requirements.

Audit trails are archived in XML format to a central audit store for centralized visibility and traceability and can be optionally secured with a system-specific digital signature to make them tamper-proof. To extract and view the audit data, you can:

- Configure the audit trail workflow to extract audit data to files, view them with third-party XML tools or move them to your audit database.
- Install DirX Audit, set up the required plug-ins and view the audit data with the DirX Audit Manager.

The section "Managing the Audit Trail" in the chapter "Managing Compliance" provides more information about these tasks.

1.9.4. Managing Status Reporting

Status reporting is part of DirX Identity's audit and compliance service.Status reports provide information about the current state of identity management data; for example, which users have which privileges, which privileges are unused, and which users have been given delegated administrative tasks.Administrators can generate status reports automatically or on demand.The section "Managing Status Reports" provides more details.

1.10. Managing Domains

A domain is a high-level segregation of Provisioning Configuration data that can be used to establish different policies and privilege models in a single Provisioning system.Domain

management tasks include:

- Setting up a domain
- Setting up domain-specific parameters and flags
- Generating information about a domain (reports)
- Setting up a corresponding Connectivity domain
- Maintaining Identity Store consistency by scheduling workflows to run all relevant rules

DirX Identity allows you to create multiple domains, which can be useful when a service provider runs one installation of DirX Identity that manages different customers; for example, company PQR and company XYZ.You can create separate domains for each customer, which allows the service provider to maintain a clear separation of each customer's access rights and data.Use the Configuration Wizard to create domains.

You can also set up multiple tenants - like company PQR and company XYZ - within a single domain; the DirX Identity Use Case Document *Using Domains* provides more information about this configuration.

1.10.1. Setting up a Domain

Setting up a domain consists of the following tasks:

- Changing the directory schema (using standard directory tools) and, as necessary, changing the object descriptions and property page descriptions of one or more of the following Provisioning objects: user, account, group, permission, role, business objects and sub-tree container
- Defining the administrators for all objects and configuring their access rights using access policies using DirX Identity Manager
- Adding the users to the DirX Identity Store (Provisioning Configuration) by hand with Web Center or DirX Identity Manager. Alternatively you can configure the workflows in the Connectivity domain to add and maintain them automatically (if you are going to import users from an external source)
- Configuring the Identity Store workflows using DirX Identity Manager
- Configuring the status report workflows using DirX Identity Manager

1.10.2. Managing Domain-Specific Information

Information management tasks consist of getting information from the Provisioning system.These tasks include:

- Monitoring the Identity Store workflows with DirX Identity Manager and the Web Admin
- Evaluating generated status reports
- Adjusting the schema and administrator responsibilities and access rights as necessary

1.11. Managing the Provisioning System

Managing the Provisioning System consists of the tasks that are necessary to provide provisioning.These tasks include:

- Creating a domain and specifying a domain administrator
- Configuring access control for the Provisioning Configuration via access policies
- Establishing general security settings, such as authentication requirements and settings for remote access
- Defining domain-specific configuration parameters

1.12. Managing States

To understand DirX Identity, it is important to understand the concept of "state" for the various objects (users, accounts, groups) and their interrelationship (account-to-group memberships).

This section describes:

- **User states** - the life-cycle and the possible states of users.
- **Persona states** - the life-cycle and the possible states of personas.
- **Functional users states** - the life-cycle and the possible states of functional users.
- **Account states** - the life-cycle and the possible states of account objects.
- **Role states** - the life-cycle and the possible states of role objects.
- **Permission states** - the life-cycle and the possible states of permission objects.
- **Group states** - the life-cycle and the possible states of group objects.
- **Assignment states** - the life-cycle and the possible states of account-to-group assignments.
- **How privilege resolution works**.
- The relationship between the user and the account life-cycle.

The descriptions of DirX Identity states given in this section use the following notation conventions to keep the state diagrams as simple as possible:

- The **dxr** prefixes are omitted from the attribute names: we use StartDate instead of dxrStartDate and EndDate instead of dxrEndDate.
- Virtual states are used. For example, if a user with a passed StartDate and a passed DisableStartDate is entered into the Identity Store, the user shortly reaches the ENABLED state due to the StartDate. Because the DisableStartDate is also passed, the state switches immediately from ENABLED to DISABLED. In this case, the ENABLED state is not stable, thus the result state is the DISABLED state.

DirX Identity supports domain-specific timing parameters that control the setting of individual timing parameters at user, account and group objects. See the section "Timing"

in the chapter "Domain Properties" of the context-sensitive help for more information about these timing parameters.

Several types of actors trigger the state transitions we describe in the next sections:

- **Administrators** - Administrators perform actions via graphical user interfaces (DirX Identity Manager, DirX Identity Web Center). These actions result in requests to a built-in services layer that performs both simple and complex operations. An example of a simple operation is the change of a status attribute at a single object; an example of a complex action is a privilege resolution that may affect hundreds of objects.
- **Services** - Services are either triggered by administrators, by schedules or by an event. An example of a service is the User Resolution workflow. It checks and resolves users, which may result in state changes of the user object and other related objects. Another example is a policy workflow, which performs consistency checks on some objects that can result in state changes. As described earlier, services can be part of a user interface operation.
- **Provisioning workflows** - Accounts and groups are representations of the real objects in the connected systems. The initial load / validation and synchronization workflows adjust the TSState attributes.



The creation workflows should not set and handle user state attributes directly. Instead, they should set the timing attributes like StartDate, EndDate, DisableStartDate, DisableEndDate and the To Be Analyzed (TBA) flag. The TBA flag triggers services that evaluate the user object and set the user states. This procedure guarantees that the state machine is always the same and not dependent on a special creation workflow logic.

1.12.1. User States

User objects represent active identity objects. Templates are inactive users that are not yet resolved. A user can have the following states:

TEMPLATE

the settings made for this user are used as a template for new users. Privilege assignments and attribute settings are not resolved for these user objects.

NEW

a new user that can be used for 'in advance' provisioning. The user does not have any rights in the domain because all accounts are disabled with no EndDate set. All necessary group assignments exist for all privilege assignments that are valid, but they are not activated due to the disabled accounts.



'In advance' provisioning means that the administrators in the connected systems can perform all necessary additional preparative steps so that the user can work immediately when state ENABLED is reached. For a detailed discussion of this topic, see the section "How Privilege Resolution Works".

ENABLED

the user is established; the corresponding user object is up to date and valid, the privilege assignments are active.
All corresponding accounts are set to state ENABLED.
All necessary group assignments exist and are activated for all privileges that are currently valid.

DISABLED

the privilege assignments for this user are currently disabled, the user does not have any rights in the domain because all accounts are disabled. All necessary group assignments exist for all privilege assignments that are valid, but they are not activated due to the disabled accounts.

TBDEL

the user object shall be deleted from the DirX Identity Provisioning system. This is done, when all assigned accounts in the connected systems are deleted or the end date for deletion is reached.



we chose to name this state TBDEL (to-be-deleted) instead of Deleted to indicate that the entry is assumed to be deleted but it still exists.

The following user attributes are related to these states (the LDAP attribute is shown in parentheses):

StartDate **(dxrStartDate)**

the date at which the user enters the company (activation of the user).

EndDate **(dxrEndDate)**

the date at which the user leaves the company or when he is scheduled to leave (deactivation of the user).

DisableStartDate **(dxrDisableStartDate)**

the start date of a temporary deactivation phase (for example, a maternity leave).

DisableEndDate **(dxrDisableEndDate)**

the end date of the temporary deactivation phase.

DeleteDate **(dxrDeleteDate)**

the date at which a user in the TBDEL state is physically removed (the LDAP entry is deleted).



entries that still contain audit trail information are not deleted until this audit trail information is moved to an audit database.

The following figure illustrates the user states and how they function in conjunction with the user attributes described here:

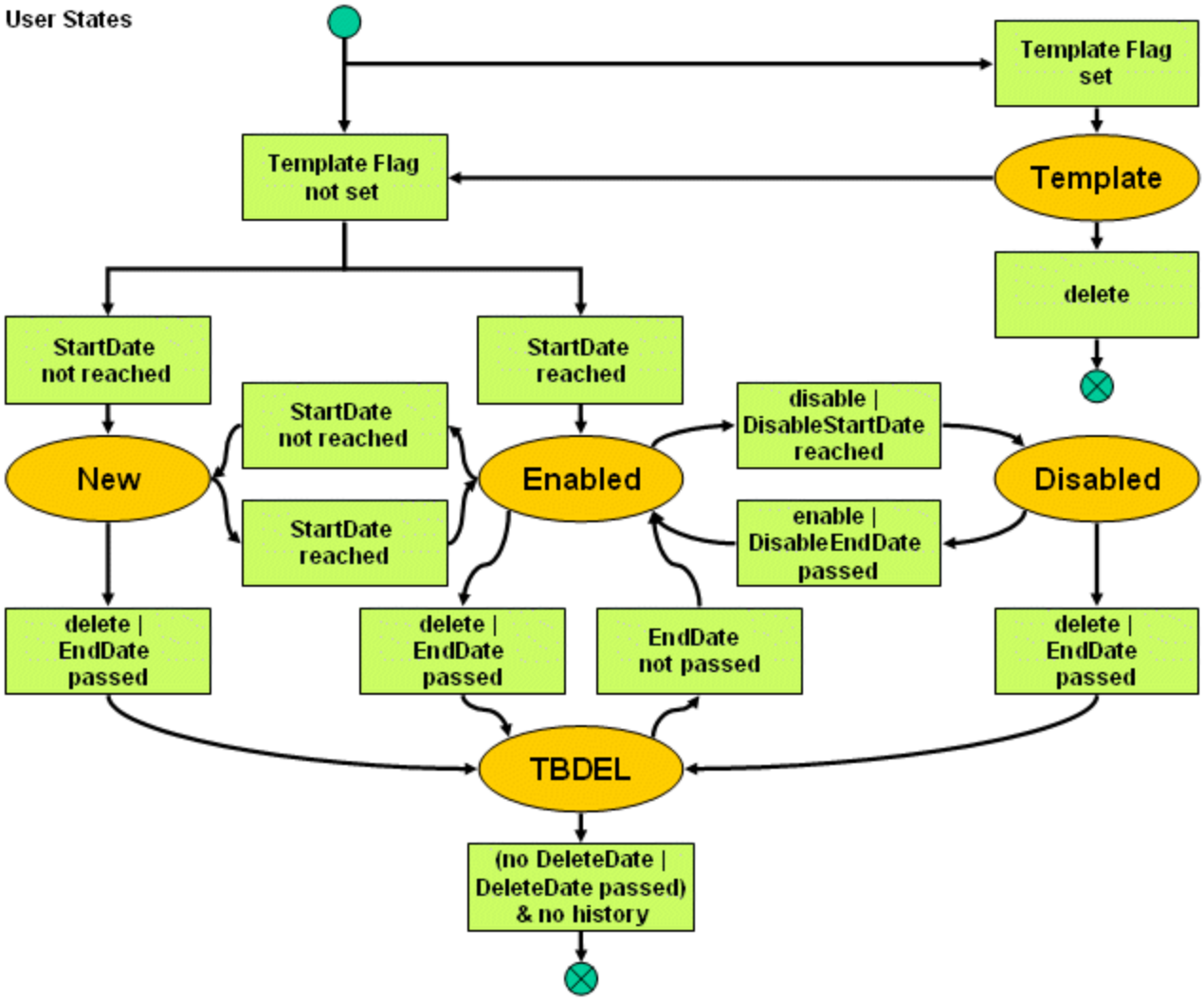


Figure 2. User State Transitions

As shown in the figure, the possible transitions are as follows:

(does not exist) → TEMPLATE - a new user object created with the "template" flag set results in the state TEMPLATE. No privilege resolution is performed; that is, privileges that are assigned are not yet resolved.

(does not exist) or TEMPLATE → NEW - a new user object or a template whose StartDate is not yet reached enters the state NEW. A privilege resolution is performed but all newly created accounts enter the state DISABLED. All necessary group assignments are made for all privilege assignments that are valid.

(does not exist) or TEMPLATE → ENABLED - a new user object or a template with StartDate reached enters the state ENABLED. A privilege resolution is performed and all newly created accounts enter state ENABLED. All necessary group assignments are made for all privilege assignments that are valid.

TEMPLATE → (does not exist) - you can simply delete templates.

NEW → ENABLED - when the StartDate is reached, the state changes from NEW to ENABLED. The state of all relevant accounts is changed to ENABLED.

NEW → TBDEL - when the EndDate is passed or an administrator deletes the entry with the user interface, the state changes from NEW to TBDEL. The DeleteDate of the user is set to 'today + maximumTimeToDeletaAnObject'. All accounts are set to state DISABLED, which results in inactive group assignments. The EndDate of all accounts are set to 'today + maximumTimeToDeleteAnObject'.

ENABLED → NEW - this state change occurs if the StartDate is not yet reached. The state of all relevant accounts is changed to DISABLED.

ENABLED → DISABLED - this state change occurs if either the DisableStartDate is reached or the entry is disabled by an administrator via the user interface. The DisableStartDate is set to today if not yet defined.
The state of all relevant accounts is changed to DISABLED. The EndDate is not set.

ENABLED → TBDEL - when the EndDate is passed or an administrator deletes the entry with the user interface, the state changes from ENABLED to TBDEL. The DeleteDate of the user is set to 'today + maximumTimeToDeleteAnObject'. All accounts are set to state DISABLED, which results in inactive group assignments. The EndDate of all accounts are set to 'today + maximumTimeToDeleteAnObject'.

DISABLED → ENABLED - this state change occurs if either the DisableEndDate is passed or an administrator enables the entry with the user interface. The state of all relevant accounts is changed to ENABLED.

DISABLED → TBDEL - when the EndDate is passed or an administrator deletes the entry via the user interface, the state changes from DISABLED to TBDEL.
The DeleteDate of the user is set to 'today + maximumTimeToDeleteAnObject'. All accounts are set to state DISABLED. This results also in inactive group assignments. The EndDate of all accounts is set to 'today + maximumTimeToDeleteAnObject'.

TBDEL → ENABLED - an entry in the TBDEL state can be reactivated by setting the EndDate to a date past the current date. DeleteDate is reset.

TBDEL → (does not exist) - this state change occurs if the DeleteDate is passed (or this attribute does not exist) and the entry does not contain any audit trail information.
All privilege assignments are removed.



A note about user and account lifetimes:

Accounts can exist for longer than their associated user. When the user reaches the TBDEL state, all the accounts change to the DISABLED state with an end date 'today + maximumTimeToDisableAnObject'. When the user's DeleteDate has passed, the DirX Identity system physically removes it. When an account's EndDate has passed, it switches to state DELETED with a new EndDate 'today + maximumTimeToDeleteAnObject'. DirX Identity only removes it from the DirX Identity Store when the corresponding account in the connected system disappears or its grace period defined by the EndDate is passed.

1.12.2. User Facet and Persona States

User facets and personas have the same states and state transitions as users. Since personas belong to the identity represented by the user, their life-cycle is related to the user's life-cycle:

- A user facet or a persona cannot outlive its related user.
- Disabling a user also disables his related user facets and personas.

The consequence for the user facet and persona state handling is as follows:

- As long as the related user is in the state **DISABLED**, the user facet and persona are also **DISABLED**, unless their state is **TBDEL**.
- If a user's state changes to **TBDEL**, the states of his user facets and personas also change to **TBDEL**, and their DeleteDates/account states are handled the same way as for the user.



You can customize the state handling of user facets and personas by implementing your own state machine for them. See the section "Custom Status Modules" in the chapter "Customizing Program Logic" in the *DirX Identity Customization Guide* for details.

1.12.3. Functional User States

Functional users have the same states and state transitions as users. In contrast to a persona, a functional user's life-cycle does not depend on his sponsor's life-cycle. If its sponsor's state changes to **TBDEL**, the functional user's sponsor attribute is cleared and a request workflow **Relink Functional User** is started to assign a new sponsor to the functional user.



You can customize the state handling of functional users by implementing your own state machine for functional users: See the section "Custom Status Modules" in the chapter "Customizing Program Logic" in the *DirX Identity Customization Guide* for details.

1.12.4. Account States

Accounts represent identities in connected systems. Accounts may have an associated user. Account states depend on the states of the associated user and on the accounts in the connected system. An account can have these states:

Imported
: the account was created / exists in the connected system, but is not yet requested (handled) by DirX Identity Provisioning.

Enabled
: the account is to exist, be created in the connected system and be enabled.

Disabled

the account is to exist, be created in the connected system and be disabled.

Deleted

the account is to be deleted in the connected system.

Reserved

the account is reserved with a specific name for a user. This is an intermediate state that occurs during assignment of a privilege.

The following account attributes are related to these states (the LDAP attribute is shown in parentheses):

***EndDate* (dxrEndDate)**

depends on the account's state: If it is DISABLED, it defines the date at which the account's state is changed to DELETED. If it is DELETED, it defines the date at which the account will be deleted.

***State in connected system* (dxrTSState)**

the state of the account in the connected system. Possible states are:

ENABLED

the account exists in the connected system and is enabled.

DISABLED

the account exists in the connected system and is disabled.

DELETED

the object has been deleted in the connected system without DirX Identity Provisioning having requested it! This state is only set by the validation workflow together with a message in the ToDo (dxrToDo) attribute that reminds the administrator to inspect this situation.

NONE

set when the object is created by the DirX Identity services.

1.12.4.1. Account State Transitions

The following figure illustrates the account states and how they function in conjunction with the account attributes just described:

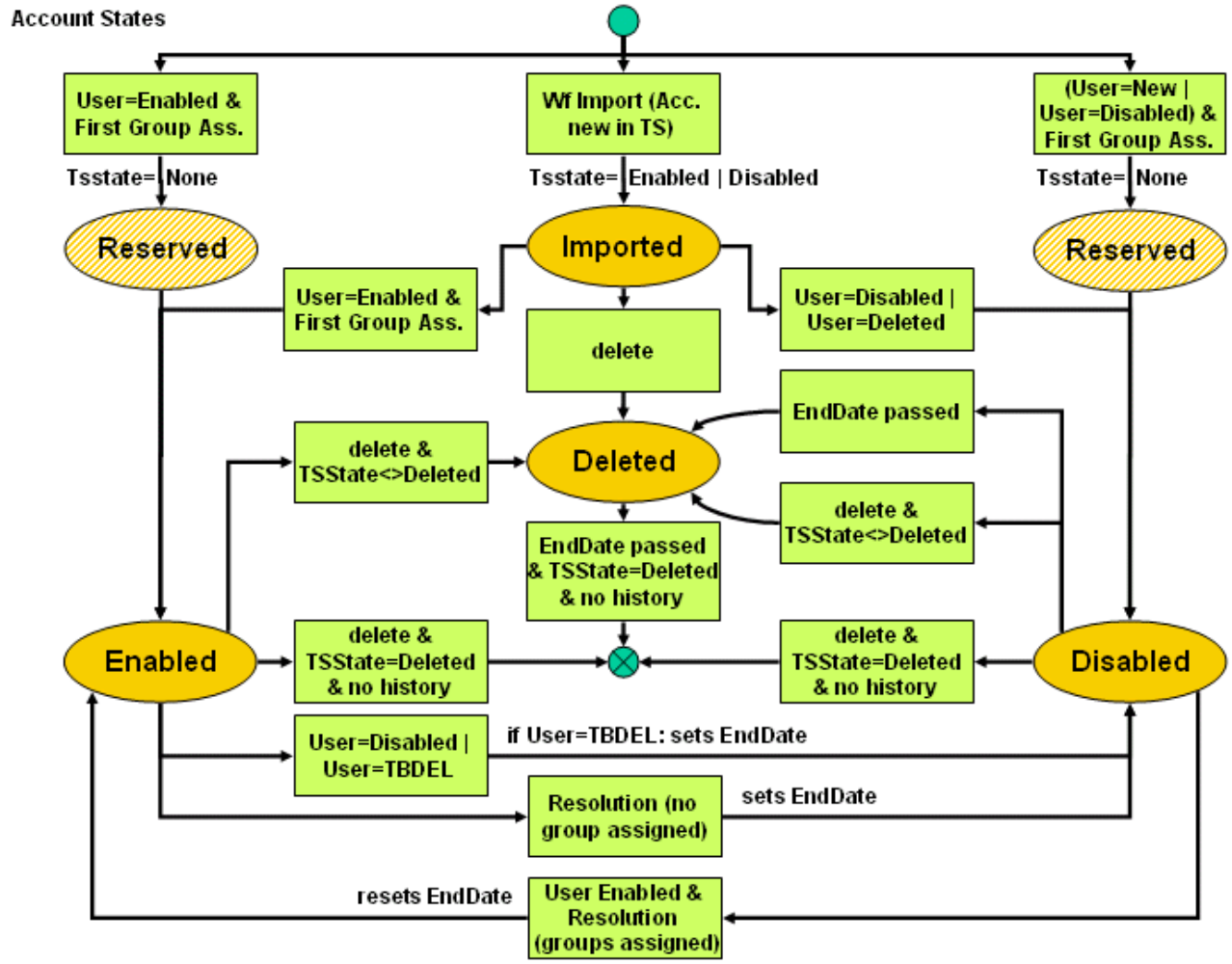


Figure 3. Account State Transitions

As shown in the figure, the possible transitions of the account state are as follows:

(does not exist) → IMPORTED - the initial load or validation workflows create new accounts in the target system in state IMPORTED. The TSState is set according to the state in the connected system.

(does not exist) → RESERVED - if the first group is assigned in the DirX Identity Manager, an account is created in the temporary state RESERVED to prevent the same account from being created in another application in parallel. If the group assignment is aborted, the account is deleted. If the group assignment succeeds, the state changes either to ENABLED if the user is ENABLED or to DISABLED if the user is in state NEW or DISABLED.



In DirX Identity Manager, the account state is always shown as ENABLED or DISABLED even if it is still in state RESERVED. In other applications (for example, Web Center), the state is shown as RESERVED.



Due to a network error or an error in the account object description, the state can remain RESERVED. To solve this problem, find and correct the cause of the error and then re-run the privilege resolution service on the user.

(does not exist) or **RESERVED → ENABLED** - if the user is in state ENABLED and the first group is assigned, the account enters the ENABLED state, too.

(does not exist) or **RESERVED → DISABLED** - if the user is in state NEW or DISABLED and the first group is assigned, the account enters the DISABLED state.

IMPORTED → ENABLED - if an account exists in the target system, the user is in state ENABLED and the first group is assigned, the account enters the ENABLED state.

IMPORTED → DISABLED - if an account exists in the target system and the user is in state DISABLED or DELETED, the account enters the DISABLED state.

IMPORTED → DELETED - if the account is deleted, it enters the state DELETED.

ENABLED → DISABLED - if a user is in state DISABLED or TBDEL, the account enters the DISABLED state (if the user is in state TBDEL, the EndDate of the account is set to 'today + maximumTimeToDisableAnObject'). If the resolution of an enabled user results in no assigned groups, the account is set to state DISABLED (the EndDate of the account is set).

ENABLED → DELETED - if the account is to be deleted and the TSState is not DELETED, the account enters state DELETED.

ENABLED → (does not exist) - if the account is to be deleted and the TSState is DELETED and there is no audit trail information present, the account is simply removed from the Identity Store.

DISABLED → ENABLED - if the user is ENABLED and the privilege resolution assigns the first group, the account enters state ENABLED.

DISABLED → DELETED - if the EndDate has passed, the account enters state DELETED. Alternatively, this state is reached if the account is deleted and the TSState is not DELETED.

DISABLED → (does not exist) - if the account is to be deleted and the TSState is DELETED and there is no audit trail information present, the account is deleted from the Identity Store.

DELETED → (does not exist) - the account is deleted from the Identity Store if the EndDate has passed, TSState is DELETED and no audit trail information is still present.

1.12.4.2. Provisioning Workflow State Handling

The provisioning workflows control the account TSState attribute: they change its value to the current situation in the connected system as follows:

- The **initial load workflow** reads all entries from the connected system and imports them into the Identity Store. The state is set to IMPORTED and the TSState is set to ENABLED or DISABLED depending on the state in the connected system.
- The **validation workflow** reads all entries from the connected system and imports the changes to the Identity Store.



Te initial load and the validation workflows are almost identical. The only

> difference is that the validation workflow writes a ToDo message in cases where the administrator must be notified.

- The **synchronization workflow** reads all necessary actions from the Identity Store, updates the connected system accordingly and then changes the TSState in the Identity Store. The synchronization workflows in the direction "connected system to Identity Store" usually operate on a delta-based scheme and therefore do not detect any unsolicited object deletions. Nevertheless, depending on the features of the connected system, the workflows may detect and acknowledge requested deletions.

The following figures and tables show all account state combinations (State, TSState) and the possible transitions that can occur during the import operation at the Identity Store.

- In the figures, dashed lines indicate transitions that should not occur under normal circumstances but which can occur due to (illegal) administrator actions or other malfunctions. DirX Identity uses the dashed-line transition to solve the problem.
- In the tables, bold states in the Resulting state / TSState column indicate that the state is changed.
- In the tables, the following columns indicate the following information:
- **Previous State / TSState** - represents the State/TSState in the Identity Store before the workflow is run.
- **Connected system account** - shows the state of the account in the connected system before the initial load / validation workflow is started or the TargetSystem → IdentityStore part of the synchronization workflow runs.
- **Resulting State / TSState** - represents the State/TSState in the Identity Store after the workflow completes.

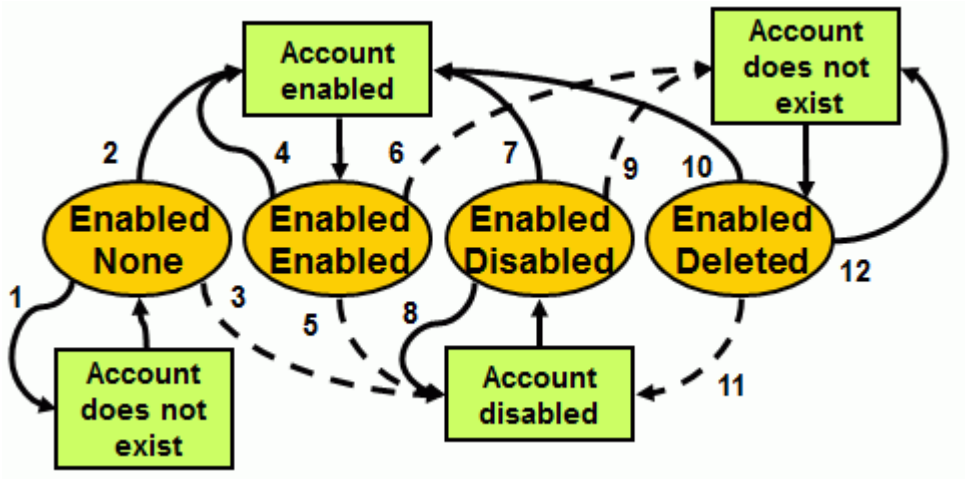


Figure 4. Account Import State Transitions (Account is Enabled)

Here is an example of how to read this diagram: the synchronization workflow reads all objects that are in state ENABLED and TSState NONE and then creates the corresponding accounts in the connected system. After successful creation, the TSState in the Identity Store is set to ENABLED, shown in the figure with the path ENABLED / NONE → Account enabled → ENABLED / ENABLED.

The following table describes these transitions (the account in the Identity Store is ENABLED):

No	Previous State / TSState	Connected system account	Resulting State / TSState	Comments
1	ENABLED NONE	does not exist	ENABLED NONE	The account has not yet been created in the connected system. The Identity Store and the connected system are synchronized.
2	ENABLED NONE	is enabled	ENABLED **ENABLED**	The account exists or has been created by the synchronization workflow in the connected system and is enabled. The TSState in DirX Identity is set accordingly.
3	ENABLED NONE	is disabled	ENABLED **DISABLED**	The account does not exist in the connected system; a connected system administrator created the account and then disabled it. The initial load / validation workflows detect the situation and set the ToDo field of the account to "Created in target system". The next synchronization workflow will enable the account.
4	ENABLED ENABLED	is enabled	ENABLED ENABLED	The Identity Store and the connected system are synchronized.
5	ENABLED ENABLED	is disabled	ENABLED **DISABLED**	The account exists in the connected system; a connected system administrator disabled it. The initial load / validation workflows detect the situation and set the TSState accordingly. The next synchronization workflow will enable the account.
6	ENABLED ENABLED	does not exist	ENABLED **DELETED**	A connected system administrator has deleted the account in the connected system. The initial load / validation workflows detect the situation, set the ToDo field to "Deleted in target system" and set the TSState accordingly. An administrator should examine the situation and resolve it (for example, delete the object in the Identity Store).

No	Previous State / TSState	Connected system account	Resulting State / TSState	Comments
7	ENABLED DISABLED	is enabled	ENABLED **ENABLED**	A connected system administrator has disabled the account in the connected system. The synchronization workflow has enabled the account again and set the TSState accordingly. DirX Identity and the connected system are synchronized again.
8	ENABLED DISABLED	is disabled	ENABLED DISABLED	The initial load / validation workflows have verified the situation but have not made any changes because the Identity Store and the connected system are synchronized. The next synchronization workflow will enable the account.
9	ENABLED DISABLED	does not exist	ENABLED **DELETED**	A connected system administrator has deleted the disabled account in the connected system by. The initial load / validation workflows detect the situation, set the ToDo field to "Deleted in target system" and set the TSState accordingly. An administrator should examine the situation and resolve it (for example, delete the object in the Identity Store).
10	ENABLED DELETED	is enabled	ENABLED **ENABLED**	A connected system administrator deleted the account in the connected system and then re-created it in the state ENABLED. The initial load / validation workflows detect the situation, set the ToDo field of the account to "Recreated in target system" and set the TSState accordingly. DirX Identity and the connected system are synchronized again.

No	Previous State / TSState	Connected system account	Resulting State / TSState	Comments
11	ENABLED DELETED	is disabled	ENABLED **DISABLED**	A connected system administrator deleted the account in the connected system and then re-created it in the state DISABLED. The initial load / validation workflows detect the situation, set the ToDo field of the account to "Recreated in target system" and set the TSState accordingly. The next synchronization workflow will re-enable the account.
12	ENABLED DELETED	does not exist	ENABLED DELETED	The account is deleted in the connected system and the synchronization workflow will *never* recreate it. Instead, the administrator must create this account in DirX Identity (either manually or by policies). A subsequent privilege resolution will create a new account.

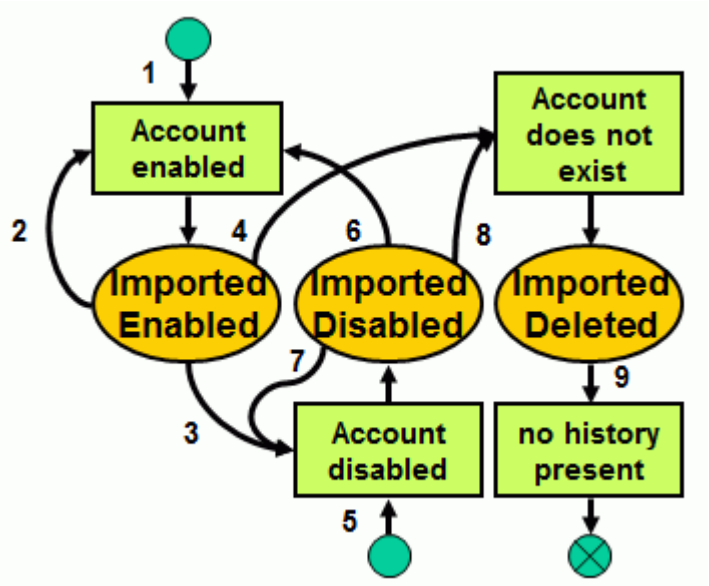


Figure 5. Account Import State Transitions (Imported or Does Not Exist)

The following table describes these transitions (the account in the Identity Store is IMPORTED or does not exist):

No	Previous State / TSState	Connected system account	Resulting State / TSState	Comments
1	(does not exist)	is enabled	**IMPORTED ENABLED**	Accounts that exist in the connected system but not yet in the Identity Store are created by an initial load / validation workflow. The TSState is set to ENABLED and the ToDo field is set to "Created in target system".
2	IMPORTED ENABLED	is enabled	IMPORTED ENABLED	Identity Store and connected system are synchronized.
3	IMPORTED ENABLED	is disabled	IMPORTED **DISABLED**	The connected system administrator has disabled the account. The TSState in DirX Identity is set accordingly.
4	IMPORTED ENABLED	does not exist	IMPORTED **DELETED**	The connected system administrator has deleted the account. The initial load / validation workflows detect the situation and set the TSState in the Identity Store accordingly. The account will be physically deleted by the corresponding consistency rule after all audit trail is removed.
5	(does not exist)	is disabled	**IMPORTED DISABLED**	Accounts that exist in the connected system but not yet in the Identity Store are created by an initial load / validation workflow. The TSState is set to DISABLED and the ToDo field is set to "Created in target system".
6	IMPORTED DISABLED	is enabled	IMPORTED **ENABLED**	The connected system administrator has enabled the account. The TSState in DirX Identity is set accordingly.
7	IMPORTED DISABLED	is disabled	IMPORTED DISABLED	The Identity Store and the connected system are synchronized.
8	IMPORTED DISABLED	does not exist	IMPORTED **DELETED**	The connected system administrator has deleted the account. The initial load / validation workflows detect the situation and set the TSState in the Identity Store accordingly. A consistency rule will delete the account after all audit trails are removed.

No	Previous State / TSState	Connected system account	Resulting State / TSState	Comments
9	IMPORTED DELETED	does not exist	(does not exist)	A consistency rule will delete the account after all audit trails are removed.

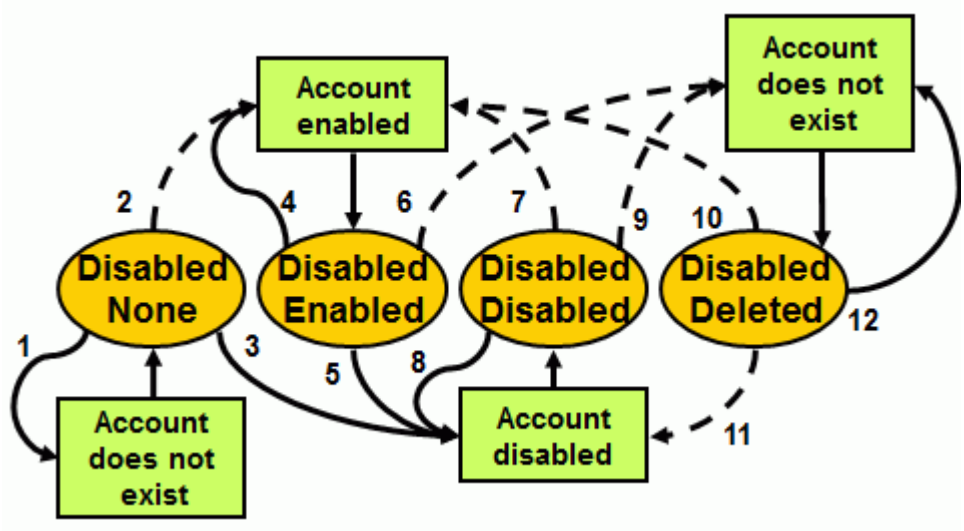


Figure 6. Account Import State Transitions (Account is Disabled)

The following table describes these transitions (the account in the Identity Store is DISABLED):

No	Previous State / TSState	Connected system account	Resulting State / TSState	Comments
1	DISABLED NONE	does not exist	DISABLED NONE	The account has not yet been created in the connected system. The Identity Store and the connected system are synchronized.
2	DISABLED NONE	is enabled	DISABLED **ENABLED**	The account did not exist in the connected system: a connected system administrator created the account and then enabled it. The initial load / validation workflows detect the situation, set the TSState to ENABLED and the ToDo field of the account to "Created in target system". The next synchronization workflow will disable the account.

No	Previous State / TSState	Connected system account	Resulting State / TSState	Comments
3	DISABLED NONE	is disabled	DISABLED **DISABLED**	The account exists or has been created by the synchronization workflow in the connected system and is disabled. The TSState in DirX Identity is set accordingly.
4	DISABLED ENABLED	is enabled	DISABLED ENABLED	The account exists in the connected system and is ENABLED. The initial load / validation workflow verifies the situation. The next synchronization workflow will disable the account.
5	DISABLED ENABLED	is enabled	DISABLED DISABLED	The account is enabled in the connected system. The synchronization workflow disables the account and sets the TSState accordingly. The Identity Store and the connected system are synchronized again.
6	DISABLED ENABLED	does not exist	DISABLED **DELETED**	A connected system administrator has deleted the account in the connected system. The initial load / validation workflows detect the situation, set the ToDo field to "Deleted in target system" and set the TSState accordingly. The synchronization workflow will never recreate the account. Instead, the administrator must delete this account in DirX Identity (either manually or by policies). A subsequent privilege resolution will create a new account.
7	DISABLED DISABLED	is enabled	DISABLED **ENABLED**	A connected system administrator has enabled the account and it exists in the connected system. The initial load / validation workflows detect the situation and set the TSState accordingly. The next synchronization workflow will disable the account.
8	DISABLED DISABLED	is disabled	DISABLED DISABLED	The Identity Store and the connected system are synchronized.

No	Previous State / TSState	Connected system account	Resulting State / TSState	Comments
9	DISABLED DISABLED	does not exist	DISABLED **DELETED**	A connected system administrator has deleted the account in the connected system. The initial load / validation workflows detected the situation, set the ToDo field to "Deleted in target system" and set the TSState accordingly. An administrator should inspect the situation and resolve it (for example, delete the object in the Identity Store).
10	DISABLED DELETED	is enabled	DISABLED **ENABLED**	The account was deleted in the connected system and then re-created afterwards by a connected system administrator in the state ENABLED. The initial load / validation workflows detected the situation, set the ToDo field of the account to "Recreated in target system" and set the TSState accordingly. The next synchronization workflow will disable the account again
11	DISABLED DELETED	is disabled	DISABLED **DISABLED**	The account was deleted in the connected system and then re-created afterwards by a connected system administrator in the state DISABLED. The initial load / validation workflows detected the situation, set the ToDo field of the account to "Recreated in target system" and set the TSState accordingly. DirX Identity and the connected system are now synchronized.
12	DISABLED DELETED	does not exist	DISABLED DELETED	The Identity Store and the connected system are synchronized.

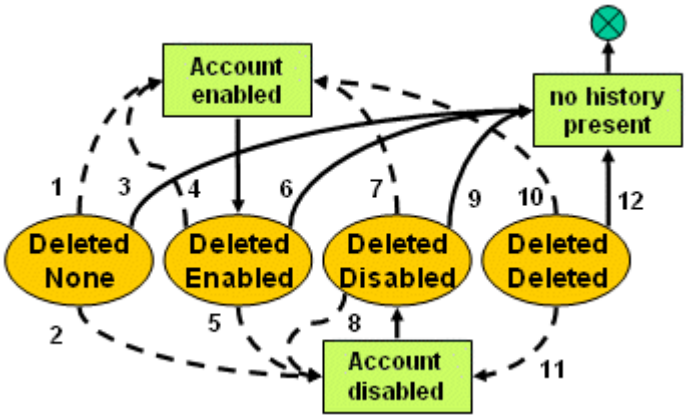


Figure 7. Account Import State Transitions (Account is Deleted)

The following table describes these transitions (the account in the Identity Store is DELETED):

No	Previous State / TSState	Connected system account	Resulting State / TSState	Comments
1	DELETED NONE	is enabled	DELETED **ENABLED**	The account was deleted in DirX Identity and does not exist in the connected system: an administrator in the connected system created it and set it to ENABLED. The initial load / validation workflows detect this situation, set the TSState accordingly and the ToDo field to "Recreated in target system". The next synchronization workflow will delete the account in the connected system.
2	DELETED NONE	is disabled	DELETED **DISABLED**	The account was deleted in DirX Identity and did not exist in the connected system: an administrator in the connected system created it and set it to DISABLED. The initial load / validation workflows detected this situation, set the TSState accordingly and the ToDo field to "Recreated in target system". The next synchronization workflow will delete the account in the connected system.

No	Previous State / TSState	Connected system account	Resulting State / TSState	Comments
3	DELETED NONE	does not exist	(does not exist)	The account has been deleted in the connected system before or has never been created in the connected system. The workflows do not delete the account in the Identity Store. Instead, the corresponding consistency rule will delete the account after all audit trail information is removed.
4	DELETED ENABLED	is enabled	DELETED ENABLED	The account is in state DELETED but still exists and is enabled in the connected system. The initial load / validation workflows detect this situation but do not change it. The next synchronization workflow will delete the account in the connected system.
5	DELETED ENABLED	is disabled	DELETED **DISABLED**	The account is in state DELETED but still exists and was disabled by an administrator in the connected system. The initial load / validation workflows detect this situation and set the TSState accordingly. The next synchronization workflow will delete the account in the connected system.
6	DELETED ENABLED	does not exist	DELETED **DELETED**	The account is in state DELETED and existed in the connected system. The synchronization workflow deletes it and sets the TSState accordingly. It does not delete the account in the Identity Store. Instead, the corresponding consistency rule will delete the account after all audit trail information is removed.

No	Previous State / TSState	Connected system account	Resulting State / TSState	Comments
7	DELETED DISABLED	is enabled	DELETED **ENABLED**	The account is in state DELETED but still exists and was enabled by an administrator in the connected system. The initial load / validation workflows detect this situation and set the TSState accordingly. The next synchronization workflow will delete the account in the connected system.
8	DELETED DISABLED	is disabled	DELETED DISABLED	The account is in state DELETED but still exists and is disabled in the connected system. The initial load / validation workflows detect this situation but do not change it. The next synchronization workflow will delete the account in the connected system.
9	DELETED DISABLED	does not exist	DELETED **DELETED**	The account is in state DELETED and existed in the connected system but was disabled. The synchronization workflow deletes it and sets the TSState accordingly. It does not delete the account in the Identity Store. Instead, the corresponding consistency rule will delete the account after all audit trail information is removed.

No	Previous State / TSState	Connected system account	Resulting State / TSState	Comments
10	DELETED DELETED	is enabled	DELETED **ENABLED**	The account was deleted under control of DirX Identity and was then re-created by an administrator in the connected system and set to ENABLED. In this case, the ToDo field of the account is set to "Recreated in target system" and the TSState is set accordingly. The next synchronization workflow will delete the account in the connected system again.
11	DELETED DELETED	is disabled	DELETED **DISABLED**	The account was deleted under control of DirX Identity and was then re-created by an administrator in the connected system and set to DISABLED. In this case, the ToDo field of the account is set to "Recreated in target system" and the TSState is set accordingly. The next synchronization workflow will delete the account in the connected system.
12	DELETED DELETED	does not exist	(does not exist)	The Identity Store and the connected system are synchronized. The initial load / validation workflows detect this situation and do not change it. The corresponding consistency rule will delete the account in the Identity Store after all audit trails are removed.

1.12.5. Role States

A role can have these states:

- **<empty>** - the role is enabled and grants the related access rights.
- **Deleted** - the role will be deleted. The state is used to store the role's delete history in the role object. Once the History Agent removes its history, the role can be deleted manually or by a consistency rule.

- **Tbdel** - the role is marked to be deleted with the next run of the Service Agent in Resolution or CheckConsistency mode. The state is used to delete the role in the Service Agent if deleting the object online is too time-consuming.

1.12.6. Permission States

A permission can have these states:

<empty> - the permission is enabled and grants the related access rights.

Deleted - the permission will be deleted. The state is used to store the permission's delete history in the permission object. Once the History Agent removes its history is removed, the permission can be deleted manually or by a consistency rule.

Tbdel - the permission is marked to be deleted with the next run of the Service Agent in Resolution or CheckConsistency mode. The state is used to delete the permission in the Service Agent if deleting the object online is too time-consuming.

1.12.7. Group States

Groups represent access rights in connected systems. A group can have these states:

Enabled - the group shall exist and be created in the connected system.

Deleted - the group shall be deleted in the connected system.

Tbdel - the group is marked to be deleted with the next run of the Service Agent in Resolution or CheckConsistency mode. As a result, the group's state will change to **Deleted**.

The following group attributes are related to these states (the LDAP attribute is shown in parentheses):

EndDate (dxrEndDate) - the date when the group will be deleted.

State in connected system (dxrTSState) - The state of the group in the connected system. Possible states are:

ENABLED - the group exists in the connected system.

DELETED - the object has been deleted in the connected system without DirX Identity Provisioning having requested it! This state is only set by the validation workflow together with a message in the ToDo (dxrToDo) attribute that reminds the administrator to examine this situation.

NONE - set when the object is created by the DirX Identity services.

1.12.7.1. Group State Transitions

The following figure illustrates the group states and how they function in conjunction with the group attributes just described:

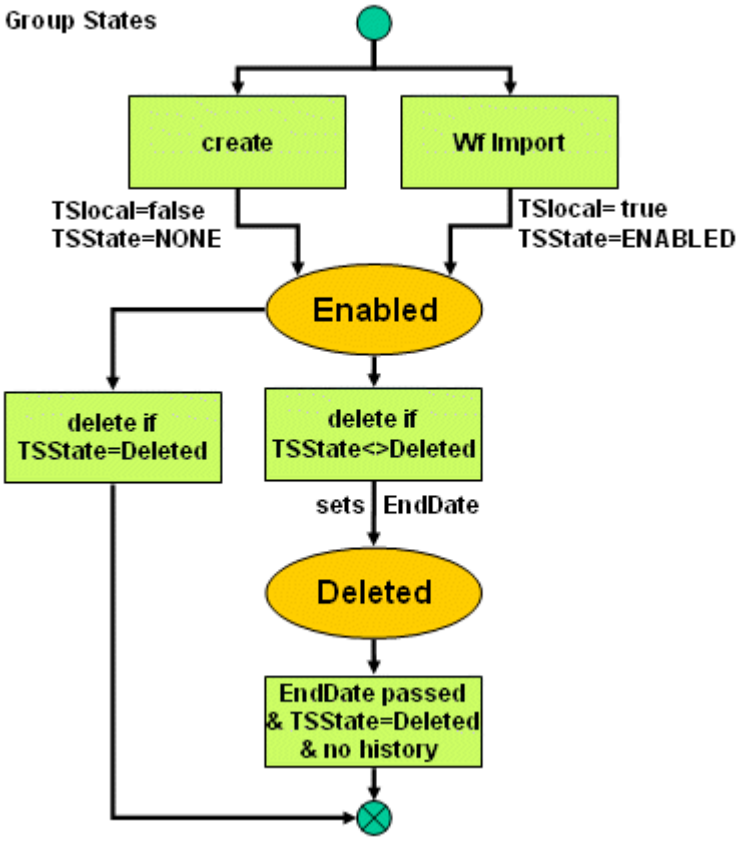


Figure 8. Group State Transitions

As shown in the figure, the possible transitions are as follows:

(does not exist) → ENABLED - the initial load or validation workflows create new groups in the target system in the state ENABLED, set TSState depending on the state in the connected system and set the TsLocal Attribute is to true. Alternatively, an administrator can create a group by hand. In this case, the TSState is set to NONE and the TsLocal flag is reset.

ENABLED → DELETED - administrators can delete groups with TSState unequal to DELETED. The resulting state is DELETED, and the EndDate is set accordingly.

ENABLED → (does not exist) - if an administrator deletes a group with TSState DELETED and there is no audit trail information available at this group entry, the group is physically removed from the Identity Store.

DELETED → (does not exist) - if the EndDate has passed, TSState is DELETED and there is no audit trail information available at this group entry, the group is physically removed from the Identity Store.

1.12.7.2. Provisioning Workflow State Handling

The provisioning workflows control the TSState attribute of groups according to the description given in the section "Account States".

The following figures and tables show all state combinations (State, TSState) and the possible transitions:

- In the figures, dashed lines indicate transitions that should not occur under normal circumstances but which can occur due to (illegal) administrator actions or other malfunctions. DirX Identity uses the dashed-line transition to solve the problem.
- In the tables, bold states in the Resulting state / TSState column indicate that the state is changed.
- The meaning of the columns is as follows:
- **Previous State / TSState** - the State/TSState in the Identity Store before the workflow is run.
- **Connected system group** - the state of the group in the connected system before the initial load / validation workflow is started or the TargetSystem → IdentityStore part of the synchronization workflow runs.
- **Resulting State / TSState** - the State/TSState in the Identity Store after the workflow completes.

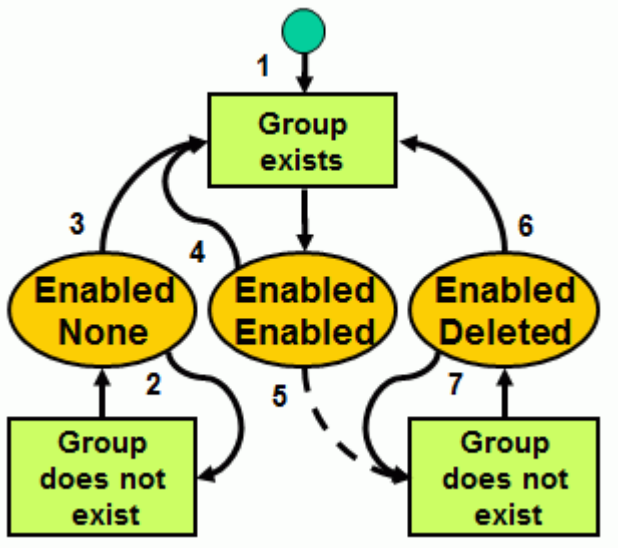


Figure 9. Group Import State Transitions (Group is Enabled)

Here is an example of how to read this diagram: the synchronization workflow reads all groups that are in state ENABLED and TSState NONE and creates the corresponding group in the connected system. After successful creation, the TSState in the Identity Store is set to ENABLED. This is shown in the figure with the path ENABLED / NONE → group does not exist → ENABLED / ENABLED.

The following table describes these transitions (the group in the Identity Store is ENABLED or does not exist):

No	Previous State / TSState	Connected system group	Resulting State / TSState	Comments
1	(does not exist)	exists	**ENABLED** **ENABLED**	Groups that exist in the connected system but not yet in the Identity Store are created by an initial load / validation workflow. The state and TSState are set to ENABLED, the TSlocal flag is set to TRUE and the ToDo field is set to "Created in target system".
2	ENABLED NONE	does not exist	ENABLED NONE	The group has not yet been created in the connected system. The Identity Store and the connected system are synchronized.
3	ENABLED NONE	exists	ENABLED **ENABLED**	The group exists or has been created by the synchronization workflow in the connected system and is enabled. The TSState in DirX Identity is set accordingly.
4	ENABLED ENABLED	exists	ENABLED ENABLED	The Identity Store and the connected system are synchronized.
5	ENABLED ENABLED	does not exist	ENABLED **DELETED**	A connected system administrator has deleted the group in the connected system. The initial load / validation workflows detect the situation, set the ToDo field to "Deleted in target system" and set the TSState accordingly. The synchronization workflow will *never* recreate the group. Instead, an administrator must delete this group in DirX Identity (either manually or by policies). A subsequent privilege resolution will create a new group.
6	ENABLED DELETED	exists	ENABLED **ENABLED**	The group was deleted in the connected system and then re-created by the connected system administrator. The workflows set the TSState accordingly. DirX Identity and the connected system are synchronized again.

No	Previous State / TSState	Connected system group	Resulting State / TSState	Comments
7	ENABLED DELETED	does not exist	ENABLED DELETED	The group is deleted in the connected system. The initial load / validation target system workflows verify this situation but do not change anything. The synchronization workflow will *never* recreate this group. Instead, the administrator must delete this group in DirX Identity (either manually or by policies). A subsequent privilege resolution will create a new group.

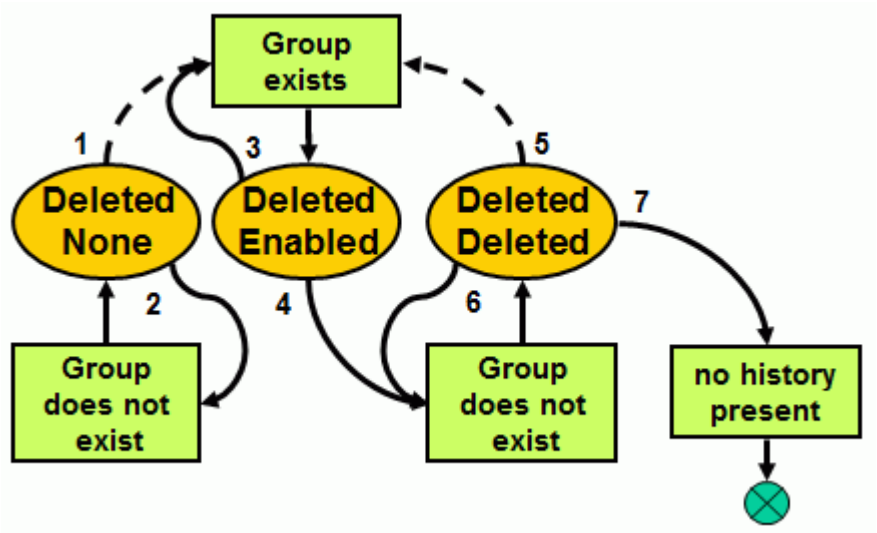


Figure 10. Group Import State Transitions (Group is Deleted)

The following table describes these transitions (the group in the Identity Store is DELETED):

No	Previous State / TSState	Connected system group	Resulting State / TSState	Comments
1	DELETED NONE	exists	DELETED **ENABLED**	The group was deleted in DirX Identity and did not exist in the connected system. It was created by an administrator in the connected system. The initial load / validation workflows detect this situation, set the TSState accordingly and the ToDo field to "Created in target system"'. The next synchronization workflow will delete the group in the connected system.

No	Previous State / TSState	Connected system group	Resulting State / TSState	Comments
2	DELETED NONE	does not exist	DELETED NONE	The group has never been created in the connected system. The initial load / validation workflows detect this situation but do not change anything. They do not delete the group in the Identity Store. Instead, a consistency rule will delete it after all audit trail information is removed.
3	DELETED ENABLED	exists	DELETED ENABLED	The group is in state DELETED but still exists in the connected system. The initial load / validation workflows detect this situation but do not change it. The next synchronization workflow will delete the group in the connected system.
4	DELETED ENABLED	does not exist	DELETED **DELETED**	The synchronization workflow deletes the group from the connected system. It sets the TSState accordingly. The corresponding consistency rule will delete the group in the Identity Store after all audit trail is removed.
5	DELETED DELETED	exists	DELETED **ENABLED**	The group was deleted in the connected system. A connected system administrator created it in the connected system. The initial load / validation workflows detect this situation, set the TSState accordingly and the ToDo field to "Created in target system". The next synchronization workflow will delete the group in the connected system again.
6	DELETED DELETED	does not exist	DELETED DELETED	The Identity Store and the connected system are synchronized. The initial load / validation workflows detect this situation and do not change it. The corresponding consistency rule will delete the group in the Identity Store after all audit trail information is removed.
7	DELETED DELETED	does not exist	(does not exist)	The corresponding consistency rule will delete the group after all audit trail information is removed.

1.12.8. Assignment States

The account-group membership is controlled by the DirX Identity resolution services and by the provisioning workflows. An assignment can have the following states:

ADD - the membership is created in DirX Identity, but does not yet exist in the connected system.

IMPORTED - the membership exists in the connected system, but is not requested by DirX Identity.

ENABLED – the membership is established both in DirX Identity and the connected system.

DELETED – DirX Identity requested to delete an existing membership. It still exists in the connected system.

IGNORE – a DirX Identity administrator explicitly accepted an imported membership. It is still not yet required by DirX Identity.

The graphical user interfaces may display an additional (virtual) state for groups: **INACTIVE**. Reasons for this state are as follows:

- The end date of this assignment has passed.
- The privilege is already assigned (for example, by the policy execution service), but a privilege resolution has not yet run. Run the privilege resolution to solve this problem.

The reason for this state is that the dxrGroupLink at the user is already populated, but the member attributes at the group or account are not yet populated.

1.12.8.1. Assignment State Transitions

The following figure illustrates the assignment state transitions:

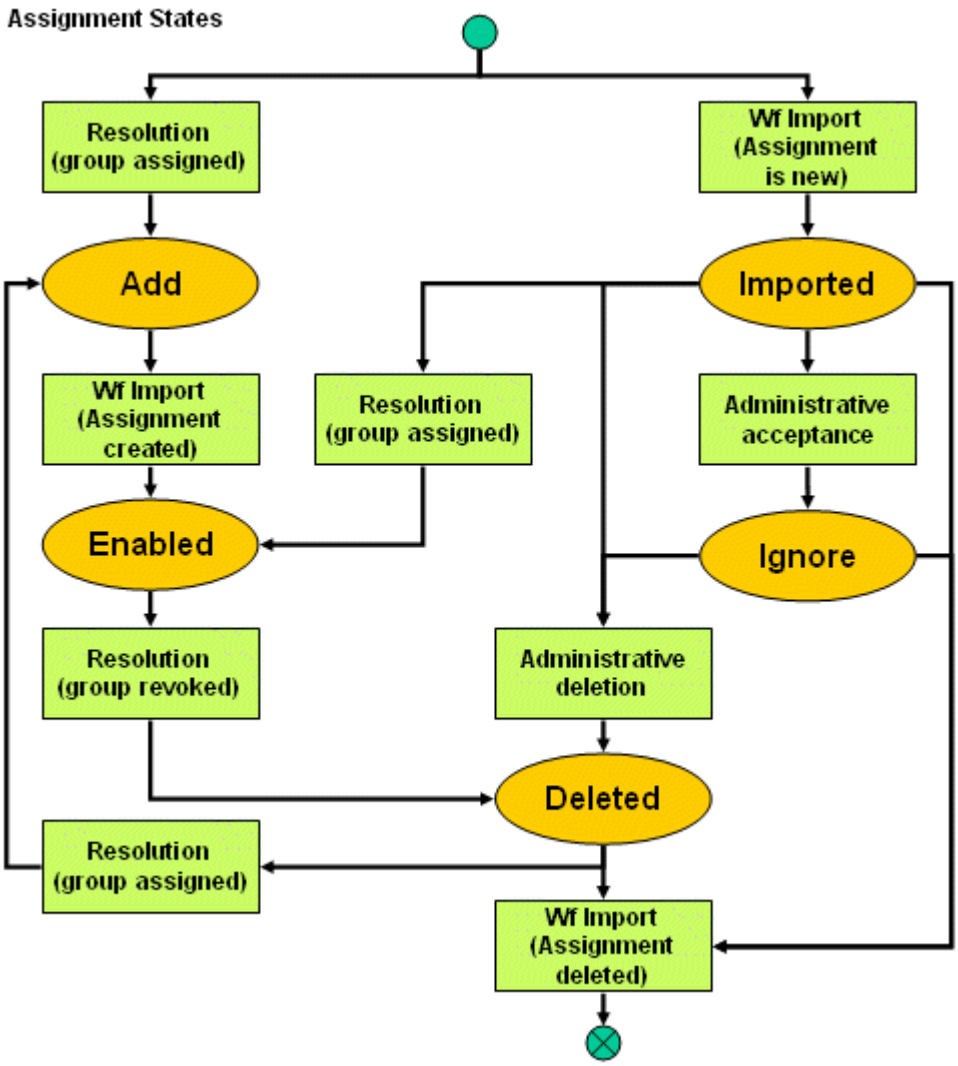


Figure 11. Assignment State Transitions

1.12.8.2. Provisioning Workflow State Handling

The provisioning workflows, in cooperation with the resolution services, control the membership between accounts and groups and change the value to the real situation in the connected system:

- The **initial load workflow** reads all memberships from the connected system and imports them into the Identity Store. The state is as shown in the figure and table below.
- The **validation workflow** reads all memberships from the connected system and imports the changes to the Identity Store.



Initial load and the validation workflows are almost identical. The only difference is that the validation workflow writes a ToDo message in cases where the administrator needs to be notified.

- The **synchronization workflow** first reads all memberships in state ADD and DELETE from the Identity Store and updates the connected system memberships accordingly. It then updates the Identity Store accordingly: it moves the ADD memberships to ENABLED and deletes the DELETE memberships).

The next figure shows the possible states of memberships and the possible transitions that can occur during the import operation at the Identity Store.

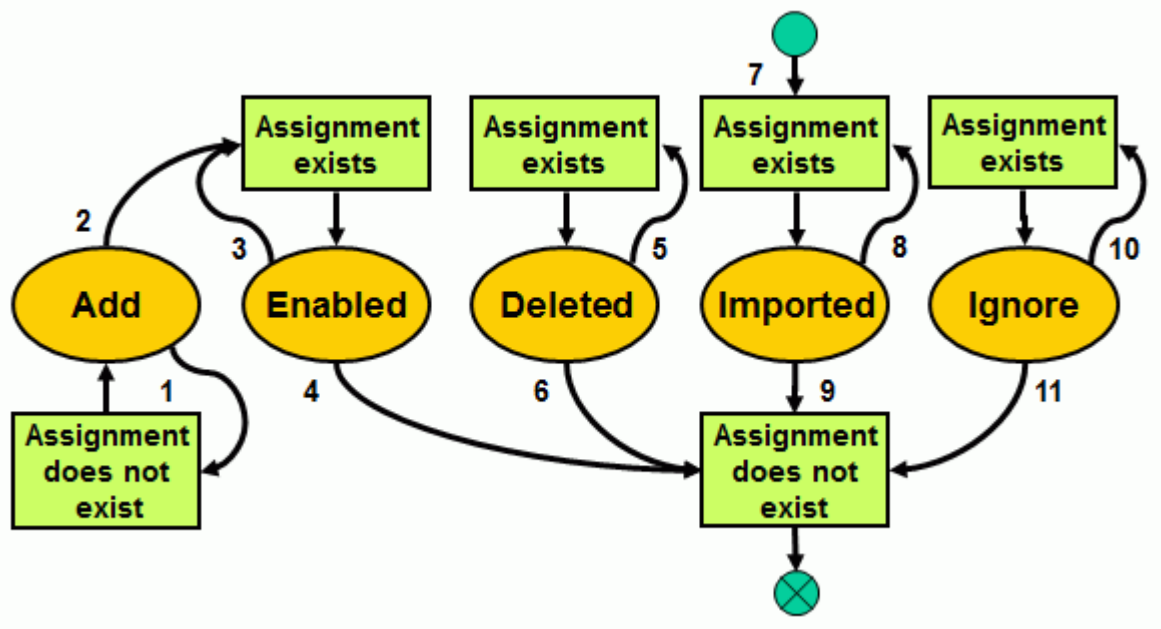


Figure 12. Assignment Import State Transitions (Account is Enabled)

Here is an example of how to read this diagram: the synchronization workflow reads all memberships that are in state ADD and adds these memberships in the connected system. After a successful add operation (the assignment now exists), the state is set to ENABLED. This is shown in the figure with the path ADD → Assignment exists → ENABLED.

The following table describes these transitions (the account in the Identity Store is ENABLED):

No	Previous State	Connected system membership	Resulting State	Comments
1	ADD	does not exist	ADD	The membership does not yet exist.
2	ADD	exists	**ENABLED**	The membership existed or was created by the synchronization workflow. The state is set accordingly. Identity Store and connected system are synchronized.
3	ENABLED	exists	ENABLED	The membership exists and is controlled by the Identity Store. The Identity Store and the connected system are synchronized.

No	Previous State	Connected system membership	Resulting State	Comments
4	ENABLED	does not exist	(does not exist)	A connected system administrator deleted the membership. The synchronization workflow deletes the membership. The next privilege resolution will create the membership again (in status ADD).
5	DELETED	exists	DELETED	DirX Identity requested to delete the membership. The next synchronization workflow will try to delete the membership again.
6	DELETED	does not exist	(does not exist)	The membership is deleted from the Identity Store.
7	(does not exist)	exists	**IMPORTED**	The membership does not exist in the Identity Store but it exists in the connected system. It is created with state IMPORTED within the Identity Store.
8	IMPORTED	exists	IMPORTED	The membership exists but is not controlled by the Identity Store. Identity Store and connected system are synchronized.
9	IMPORTED	does not exist	(does not exist)	The membership is deleted from the Identity Store.
10	IGNORE	does not exist	(does not exist)	The membership is deleted from the Identity Store.

In the previous table:

- Bold states in the Resulting state / TSState column indicate that the state is changed.
- The meaning of the columns is as follows:
- **Previous State / TSState** - the State/TSState in the Identity Store before the workflow is run.
- **Connected system membership** - the state of the membership in the connected system before the initial load / validation workflow is started or the TargetSystem -> IdentityStore part of the synchronization workflow runs.
- **Resulting State / TSState** - the State/TSState in the Identity Store after the workflow completes.

1.12.9. How Privilege Resolution Works

You can assign privileges either by hand or via provisioning rules. You can also set a start date and an end date for a manual assignment to restrict its duration. After new privileges are assigned, (and if an optional approval is granted) DirX Identity performs a privilege resolution. The privilege resolution process collects the assignments from associated user facets and then calculates the actual memberships and the necessary accounts for the user depending on these assignments and their attributes.

1.12.9.1. Calculating Account-To-Group Memberships

Account-to-group memberships can result from several sources, for example, a time-restricted user-to-group assignment or a role or permission assignment. The privilege resolution process calculates the duration period for each assignment. It can be defined for the future (period is FUTURE), it can be active (period is ENABLED) or it may have already passed (period is PASSED).

The privilege resolution process calculates the resulting period as a sum of all periods, where the priority is ENABLED → FUTURE → PASSED. It calculates the resolution result for the current time and sets the memberships accordingly (see below):

- The resolution result is ENABLED if one or more of the assignments are in the ENABLED state.
- The resolution result is in the FUTURE state if none of the assignments is in the ENABLED state but one or more assignments is in the FUTURE state.
- Otherwise, the resolution result is in the DELETED state.



These intermediate duration periods are not visible at the user interface. Only the resolution result is visible as the state of the account-to-group membership.

1.12.9.2. Calculating Account States

The sum of the account-to-group memberships and the user's state define the state of the account. DirX Identity supports "in advance" provisioning; that is, privileges can be assigned for a future period or the user can still be in the NEW state. With "in advance" provisioning, DirX Identity provisions the connected systems with all the necessary accounts (but these accounts remain in the DISABLED state) and all the necessary account-to-group memberships to enable the connected system administrators to perform advanced additional provisioning tasks if necessary.

The calculation procedure runs as follows:

- If a user is in the NEW state, all accounts are set to the DISABLED state and all account-to-group memberships are set to state ADD.
- If a user is in the ENABLED state and has only "in advance" account-to-group memberships (period is FUTURE), the account is set to the DISABLED state. The account-to-group assignments are set to the ADD state.
- If a user is in the ENABLED state and has mixed account-to-group assignments (some

are in period FUTURE and some in period ENABLED), the account is set to the ENABLED state and only memberships with period ENABLED are generated in state ADD. All memberships with period FUTURE are not generated in state ADD.

- If a user is in the ENABLED state, has only "in advance" account-to-group memberships (period is FUTURE) and one or more of these memberships enters the period ENABLED, the account is set to the ENABLED state. All account-to-group memberships in period FUTURE are set to the DELETED state and only the memberships in period ENABLED are set to state ADD.

1.12.10. User and Account Lifecycles

The lifecycle of users and accounts is closely related. This section explains a typical user lifecycle and the associated account lifecycle. The following figure shows a typical user lifecycle:

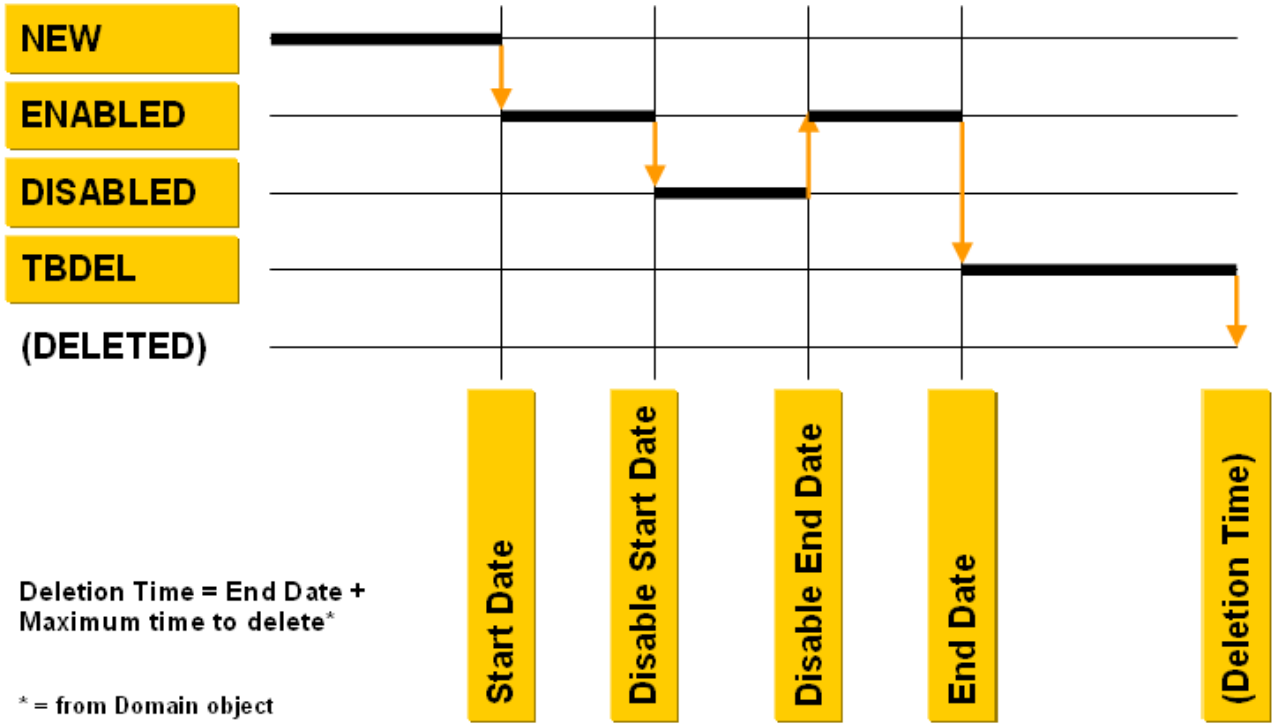


Figure 13. Typical User Lifecycle

As shown in the figure:

- After creation of the user, the **Start Date** is not yet reached. The user is in state **NEW**. In this phase privileges can be assigned that are not yet active in the connected systems.
- Reaching the **Start Date**, the DirX Identity services change the status automatically to **ENABLED**. Now all accounts and group memberships are enabled if the corresponding privileges are active.
- For some period defined by a **Disable Start Date** and a **Disable End Date**, the user is in state **DISABLED**. This means that the corresponding privileges and accounts are not active in the connected systems.
- Reaching the **Disable End Date**, the DirX Identity services change the status automatically to **ENABLED** again. Now all accounts and group memberships are

enabled again if the corresponding privileges are active.

- If the **End Date** is reached (the user leaves the company), the DirX Identity services change the status automatically to **TBDEL** (to be deleted). A complex procedure that can last months or years is performed that is explained in the next section in more detail. This time is defined by the **End Date** + **Maximum time to delete** (defined at the domain object).

The account lifecycle and the user lifecycle are closely related, as shown in the following figure:

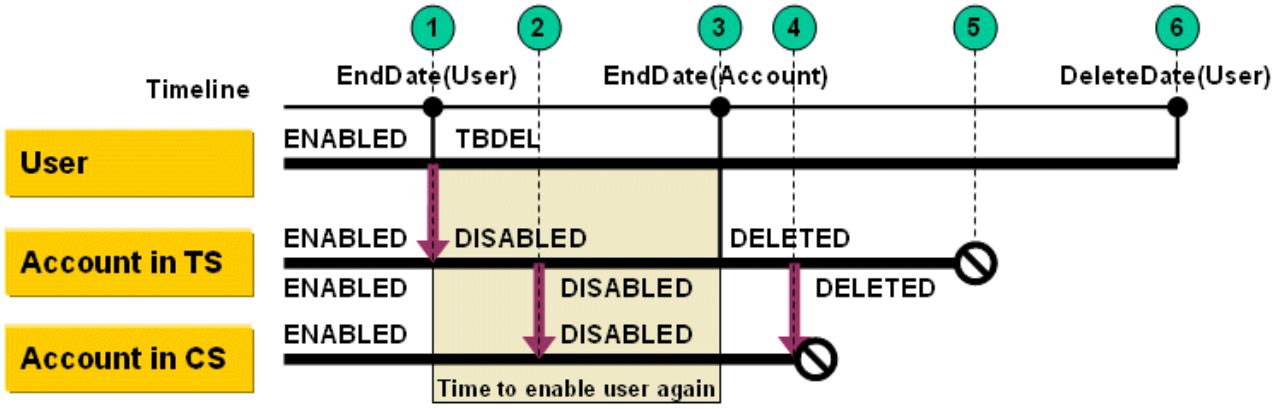


Figure 14. Relationship between User and Account Lifecycles

Until the user reaches the TBDEL status, the user is in status ENABLED. All associated accounts are in status ENABLED in the target systems of the Identity Store. The TS status in the target systems is also in ENABLED status. All related accounts in the connected system are also in ENABLED status.

The following steps are:

1. If the **End Date** is reached for the user entry (the user leaves the company), the DirX Identity services change the user status automatically to **TBDEL** (to be deleted). The **Delete Date (User)** is set to **End Date (User) + Maximum time to delete (Domain)**.



The **Maximum time to delete** is taken from the domain object. This results in immediate disabling of all accounts (state of accounts goes to **DISABLED**). The TS state is still in **ENABLED** state. The End Date of all accounts is set to **End Date (User) + Maximum time to disable (TS or Domain)**.



The **Maximum time to disable** is taken from the target system and if not available from the domain object.

2. Provisioning (synchronization) workflows are run that perform the disabling in the target system (state of the accounts goes to **DISABLED**). If this operation is successful, the TS state is in the target system is set to **DISABLED** to reflect the state in the connected system. If these workflows are run scheduled, this can last some time. If the workflows are run event-based, the change happens almost immediately.

Now the user is in state **TBDEL** and all account states are set to **DISABLED**, which means that the user is no longer active but not yet really deleted.This feature allows the user to be reactivated until the first account **End Date** is reached. The reason for this feature is that deleting the accounts instead of disabling them would mean that a reactivation of the user would require creation of the deleted accounts. In many systems, accounts have unique identifiers and security identifiers that are established by the connected system (for example, Active Directory). Creating an account again means that these identifiers are different. The user no longer has access to the resources he had before. So it is very important to keep the accounts in disabled state instead of deleting them.

3. If the End Date of an account is reached, the DirX Identity services set the state to **DELETED**.In the target system the TS state is still in state **DISABLED**.
4. Next, a provisioning (synchronization) is run that deletes the account in the connected system physically. If successful, the TS state in the target system is set to **DELETED**.

   The user is still in state **TBDEL** and after the defined time (**Maximum time to disable** for accounts) all account states are set to **DELETED**.
5. Cleanup workflows (policy execution service with consistency rules) are run from time to time that try to delete accounts that are in state **DELETED**. If auditing is enabled, the deletion is only performed if all history information is removed from the entries.
6. Cleanup workflows (policy execution service with consistency rules) are run from time to time that try to delete users that are in state **DELETED**. If auditing is enabled, the deletion is only performed if all history information is removed from the entries.



The user deletion is performed even if not all accounts are deleted.

1.13. Managing Attribute Flow



One of the most important aspects of identity management is synchronization of attributes to provide a consistent view of identities all over the identity domain. The next figure shows how attribute flow is achieved through the identity system.

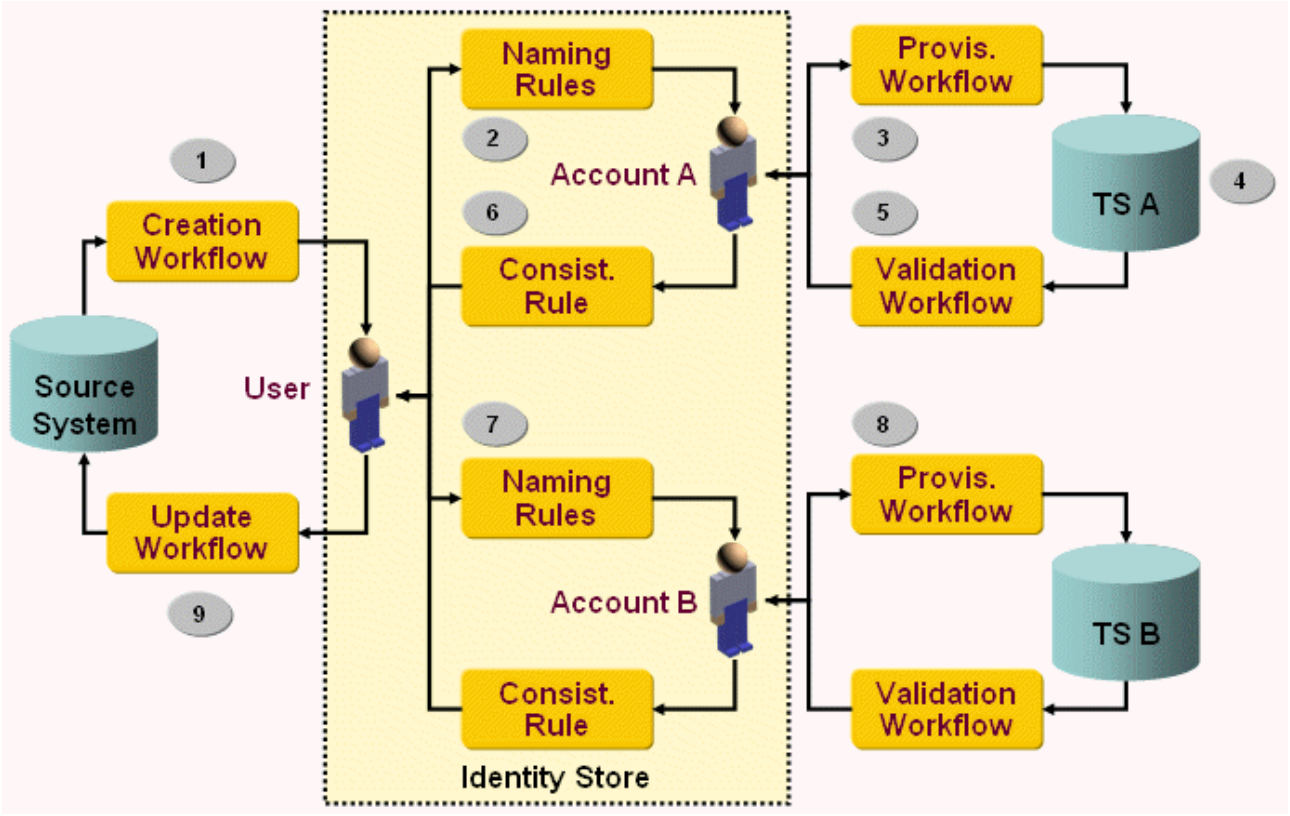


Figure 15. Attribute Flow

In this example, identity data for a user comes from a source system (for example, an HR system).As illustrated in the figure, the attribute flow steps are as follows:

1. The **Creation Workflow** extracts the identity and its attributes from the Source System and transfers it to the Identity Store, where it creates the user entry. If attributes change, the Creation Workflow performs an update of these attributes. The Source System masters these attributes. For information on how to configure creation workflows, see the section "Configuring the Source Workflows" in the *DirX Identity Application Development Guide*.
2. If privileges are assigned to this user, accounts are created in target systems. In this example, Account A is created. You can use **Naming Rules** to transfer attributes from the user entry to the account. Naming rules are definitions within the attribute definition in the account object description. See the section "Properties and Property Elements" in the *DirX Identity Customization Guide* for more information.
3. **Provisioning Workflows** synchronize the account attributes to the corresponding account objects in the target systems. For information on how to configure Provisioning Workflows, see the section "Configuring the Target System Workflows" in the *DirX Identity Application Guide* for Java-based real-time provisioning and the section "Configuring the Provisioning Workflows" in the *DirX Identity Application Development Guide* for Tcl-based provisioning workflows.
4. In this example, attributes that are created in target systems need to be distributed to other parts of the identity domain. A good example is an email address or a telephone number. The target system masters these attributes.
5. **Validation Workflows** or **Provisioning Workflows** transfer these attributes to the representation of the account in the Identity Store. Validation Workflows update all

accounts, whereas Provisioning Workflows only transfer information about specific entries that are handled during this synchronization. See the section "Configuring the Provisioning Workflows" in the *DirX Identity Application Development Guide* for more information.

6. Use **Consistency Rules** to copy the changed attributes from the account entries to the user entries in the Identity Store. For information on how to configure these rules, see the section "Managing Automatic Provisioning".



The event-based workflows for example the EventBasedAccountProcessing workflow provide alternative means to handle such tasks.

7. As in Step 2, you can use **Naming Rules** to transfer these attributes to the accounts in other target systems.
8. **Provisioning Workflows** synchronize the account attributes to the corresponding account objects in these target systems.
9. **Update Workflows** can synchronize this information back to Source Systems.

This example describes attribute mastering from source and target systems. Attributes can also be mastered directly from user entries in the Identity Store. In this case, you use the Web Center or the DirX Identity Manager to maintain these attributes. The update processes are analogous to the steps described here.

2. Planning for Provisioning

These topics make up a planning checklist of the issues that you must take into account when deciding how to deploy DirX Identity Provisioning in your environment.

Planning tasks include:

- Selecting the master directory or directories
- Identifying the target systems
- Identifying the roles and permissions
- Identifying the target system groups
- Identifying the policies
- Defining the request workflows
- Deciding on auditing
- Identifying the administrators
- Setting up important services
- Deciding on customizations

2.1. Select the Master Directory or Directories

- What is the user master directory: DirX Identity or others?
- Is there only one, or are there several? For example, there could be a master corporate directory, a Human Resources directory that contains full-time employee data, another directory for external employees such as consultants, a customer database, and so on.

2.2. Identify the Target Systems

- What are the target systems that must be integrated?
- Do these target systems already exist?
- What are the characteristics of the target systems in terms of authentication and authorization?
- How should these target systems be migrated into the DirX Identity Provisioning environment?Can there be an initial load of accounts and/or groups into DirX Identity?
- Is reconciliation a requirement?
- Is there a need for a virtual target system?

2.3. Identify the Roles and Permissions

Permissions represent technical roles that can contain parameters.

- What permissions are needed?

- Does it make sense to use permission parameters? If so, how should they be applied to permissions?
- Do the permissions - as generalized access rights - fit the existing groups?
- Are you using permission parameters? If so, you must assign values for them to the groups
- How should permissions be assigned to target system groups?
- Have approval workflows been established for the assignment of the permission to users or to roles?Approvals must be established and the workflows must be set up.

If you intend to use roles, you must take into account additional issues:

- Have all the necessary roles been defined?
- What should the role structure be? That is, how should the roles be structured in the role subtree presented by DirX Identity Manager, as organized by role folders?
- Are there any role hierarchies? If so, what is the aggregation of access rights for the hierarchy?
- Which roles shall have role parameters and corresponding match rules?
- How should roles be assigned to permissions?
- Have approval workflows been established for assignment of the role to users or to other roles?Approvals must be established and the workflows must be set up.
- Are the roles delivered with the DirX Identity role catalog sufficient, or does an integrator workflow need to be developed to load customer-specific role definitions into the directory?

2.4. Identify the Target System Groups

- Are all the pre-existing groups assigned to at least one permission?
- Do you need additional groups? (Look at the permissions to determine the need)
- Have the groups for new target systems been defined?
- Have approval workflows been established for the assignment of the group to users or to permissions?Approvals must be established and the workflows must be set up

2.5. Identify the Policies

To automate your provisioning environment as far as possible and in a consistent way, think about the necessary policies.

Defining Access Policies

- Which access policies are necessary for read and modify access rights?
- Should the read / modify policies be applied on attribute level?
- Which access policies for object creation and deletion are necessary?

- Are there access policies for grant operations to be built?
- Are access policies for request workflow execution and control required?
- Which access policies for approval are necessary?
- Which access policies for delegation make sense?
- Which access policies shall be delegatable?
- Do you intend to use menu policies?

Defining Attribute Policies

- Which attributes to protect against not approved changes?

Defining Event Policies

- Are there objects where changes shall result in event notification?
- Which event-based workflows are necessary to process these change events?

Defining Delete Policies

- Are there objects where deletions shall result in event notification?
- Which event-based workflows are necessary to process these deletion events?

Defining Provisioning Rules:

- Which groups should be assigned directly to users?

If you intend to use roles you must take into account additional issues:

- How should permissions be assigned to users?

How should roles be assigned to users?

Defining Consistency Rules:

- Can you use any of the delivered default policies in your environment?
- Do you need any other rules to maintain or document your environment?

Defining Validation Rules:

- How should groups and accounts be cleansed after initial load from the target systems?
- Which rules are necessary after validation with target systems?
- Do you need any other purify rules to keep your provisioning environment up to date and correct?

Defining Password Policies

- Which types of password policies are necessary (for administrators, users, services)?
- Must your password policies be compliant with the Microsoft Windows password

policies?

In general, you can check if it makes sense to reuse rules from the sample domain.See the chapter "Reusing the Sample Domain" in the *DirX Identity Tutorial*.

2.6. Define Request Workflows

If you intend to use request workflows:

- Which privileges must be approved before assignment to a user?
- Which privileges require regular or scheduled certification campaigns of the users or privileges that they are assigned to?
- Which structural links between privileges have to be approved?
- Which attribute changes have to be approved?
- Which object creation workflows do you need?
- Which object deletion workflows do you need?
- Are copies of the pre-configured request workflow templates sufficient or do you need additional ones?

In general, you can check to see if it makes sense to re-use request workflows provided with the sample domain.See the chapter "Reusing the Sample Domain" in the *DirX Identity Tutorial* for details and think about using the transport mechanism to copy samples from other domains to your domain.

2.7. Decide on Auditing

If you intend on using the audit trail:

- Is the standard configuration of the DirX Identity audit trail mechanism sufficient for your needs?
- Is the tracing configuration of the target system workflows sufficient for your needs?

2.8. Identify the Administrators

- Who will administer DirX Identity Provisioning?
- What are the access rights to be assigned to these administrators (use access policies)?

2.9. Set up Important Services

Correct administration of an identity management system requires the setup of some services.You need to:

- Set up consistency checks
- Set up privilege resolution

- Set up processes for object cleanup
- Set up event-based processing workflows
- Set up certification campaigns

2.10. Decide on Customizations

- Do the Provisioning objects (the relevant part of the directory schema and DirX Identity Manager property pages) need to be customized?
- Do the DirX Identity Manager property pages for the target system accounts and groups need to be customized?
- Do the user integration workflows (the workflows that synchronize the master entries into DirX Identity) need to be customized?
- Do the target system initial load (the workflows that export account, group, and account-group data from the target systems and import it into the Identity Store) and validation workflows (the workflows that detect deviations between the target systems' accounts and groups and their representations in the Identity Store) need to be customized?
- Do the target system synchronization workflows (the workflows that synchronize accounts and groups to the target systems) need to be customized?
- Are any other workflow types necessary (for example for password synchronization or restore)?
- What are the naming rules or JavaScripts that apply for creating target system accounts and default passwords?
- Are all necessary policies set up?
- Have you customized the necessary request workflows?
- What about customized certification campaigns and their related processes with certification campaign user hooks?
- Does Web Center need to be customized? Are menu policies required to restrict access to certain functions for specific user types (end users, administrators, and so on)
- Have you customized the Identity Store and status report workflows, especially their schedules? Make sure that you perform this task.
- Have you optimized the workflow schedules? Make sure that you perform this task.
- Have you thought about using nested workflows to run synchronization or validation tasks in parallel or in sequence? It is a good idea to consider this methodology.
- Have you customized audit policies and set up the connection to your audit system?

3. Managing Users

This section provides information about the pre-requisites for user management. User management consists of the following tasks:

- Maintaining an accurate and up-to-date directory of the users to be administered with DirX Identity. This task depends on whether DirX Identity is the master or whether a corporate user directory is the master.
- Assigning privileges to users (roles, permissions, groups) either by hand or fully automatically via policies (controlling access rights).

This section also provides information about:

- How to work with user query folders to search for users.
- How to work with user states.
- How to work with links at user entries.

3.1. User Management Pre-Requisites

The following tasks need to be performed before you can manage DirX Identity users:

- The existing account and group data of all target systems for which user access rights are to be granted must be imported into the Identity Store. Use the initial load workflows to perform this task.
- The accounts must be assigned to users (the initial load workflow usually performs this step automatically). You can run a consistency rule at any time to check for unassigned accounts and to assign the correct users. Assignment by hand is the third option to perform user-account assignment.
- If there are multiple accounts assigned to a user within one target system, only one primary account is managed by DirX Identity: its state and its assigned groups are calculated by the privilege resolution of the user. You can apply a consistency rule to create personas for non-primary assigned accounts. These accounts are then re-assigned to the persona and become primary accounts that are managed by the privilege resolution of their persona.
- You can apply a consistency rule to create functional users for unassigned accounts. These accounts are then assigned to the functional users and are also managed by DirX Identity. Be careful to determine whether it makes sense to create a functional user for the account or assign it to an existing user.
- If customer-specific user attributes and default values are required, the necessary object descriptions, their presentation in the DirX Identity Manager, and the underlying LDAP schema must be extended to support these attributes and defaults.

If you use the Role Package (which requires a separate license), you should consider these tasks:

- The set of roles that can be applied to users must be defined.

- The privilege structure must be defined, including the role-to-permission and permission-to-group relationships.

In addition, the following Provisioning system integrator workflows must be configured:

- The workflow that periodically imports users and their attributes from the corporate user directory into the Identity Store
- The workflows that synchronize user accounts, groups, and account-group assignments in the Identity Store with the target systems
- The consistency workflows that maintain the integrity of the user database according to the lifetimes set for user-to-role assignments, user-to-group assignments, and for the users themselves

3.2. Managing the Users (DirX Identity is Master)

Managing the users when DirX Identity is the master consists of the following tasks:

- Viewing user properties
- Adding users to the store
- Deleting users from the store
- Changing the attributes of existing users
- Changing the structure of the user subtree

When the DirX Identity masters the user data, you use DirX Identity Manager to perform these tasks by hand, or you can delegate these tasks to other administrators or user managers who can use the DirX Identity Web applications Web Center or Business User Interface (BUI) to perform them. This document describes how to work with DirX Identity Manager.

3.2.1. Viewing Users with DirX Identity Manager

When you log into DirX Identity Manager and select **Users** from the view bar, DirX Identity Manager displays a hierarchical tree of the users that you are allowed to manage in the left-hand pane.

To view the properties of a user, click its entry in the tree. DirX Identity Manager displays a property dialog with a set of tabs that you can use to view (and edit) the user's attributes, assigned roles, assigned permissions, assigned groups, and target system accounts. Click the tabs in the property dialog to move between the different property categories. In the tree, users are displayed with a combination of their surname and their given name.

The current status of a user is displayed in brackets at each user entry if the user is not in the ENABLED state. For more information about user states, see the section "User States" in the section "Managing States" in the chapter "Managing Provisioning" and the section "Working with User States".

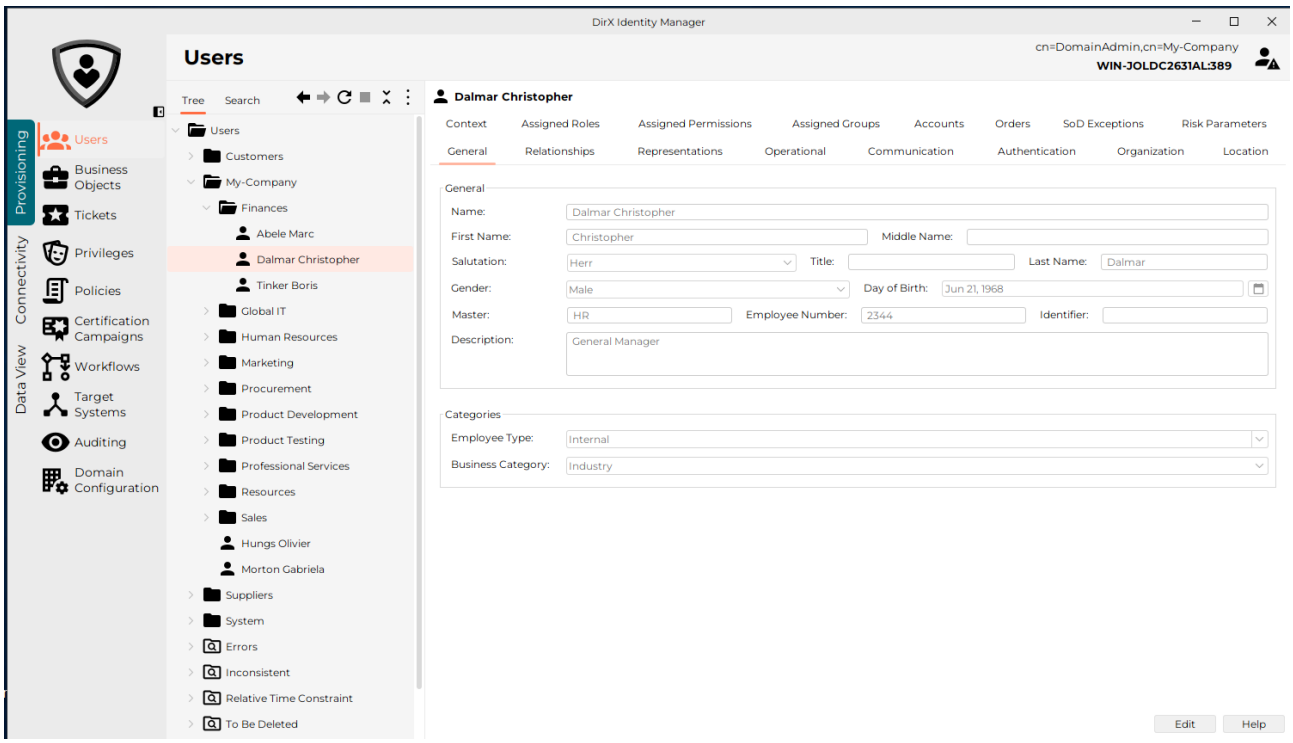


Figure 16. Managing Users with DirX Identity Manager

DirX Identity Manager also provides a search dialog that you can use to select and display a subset of users.



Personas, facets, and functional users are also assigned the dxrUser object class. As a result, searching for users with **objectClass=dxrUser** also returns personas, facets, and functional users. If you want to exclude them from the search result, you must add an AND filter with **(!(objectClass=dxrPersona))** / **(!(objectClass=dxrUserFacet))** / **(!(objectClass=dxrFunctionalUser))**. This is also true for provisioning rules, consistency rules and other components that make use of user searches.

3.2.2. Adding Users with DirX Identity Manager

To add a new user:

1. Click a user folder in the subtree or click the top-level User folder.
2. Select **New** → **User** in the context menu. The General tab is displayed for editing, and the mandatory attributes for a user (the user's common name (cn) and surname (sn)) are shown in red.

When you add a new user, you can:

- Specify a user lifetime (a start date and an end date)
- Create it as a user template

3.2.2.1. Specifying a User Lifetime

When you add a new user, you can specify the date at which the user is to become active; for example, to correspond to the date at which a new employee is to begin work. DirX Identity assigns the user state NEW to newly created users whose start dates have not yet occurred. If you do not specify a start date, DirX Identity immediately activates the user and assigns it the user state ENABLED.

You can also specify the date at which the user is to be removed when you first create the user; for example, to correspond to a contract end date for an outside consultant.

3.2.2.2. Creating User Templates

When you create a new user, you can create it as a user template by checking Use as Template. A user template is a user whose role and group assignments will never be resolved. You can create user templates and assign lifetimes and roles to them, and then use these templates to create real users. DirX Identity assigns the state TEMPLATE to user templates.



User templates can only be created when a user is initially added to the user subtree. Converting an existing user into a user template is not supported.

In the user subtree, user templates appear identical to regular users. The Use as *Template* checkbox is the sole indicator that an entry is a user template. To avoid confusion, user templates should be maintained in a separate folder within the subtree.

To create a real user from a user template, you can copy the user template, open it for editing, and uncheck Use as Template. You can also change a user template to a real user by selecting it, and then unchecking Use as Template.

3.2.3. Deleting Users with DirX Identity Manager

To delete a user or a user template, click it and then select Delete in the menu bar or context menu.

When you delete a user template, DirX Identity removes it from the Identity Store.

When you delete a user, DirX Identity Manager:

- Marks the user as "to be deleted" (user state TBDEL) and sets a delete date for the user according to the domain configuration parameter "maximum time to delete an object"
- Stores this information in the DirX Identity store

A subsequent privilege resolution – either by DirX identity Manager or by a backend process:

- Removes all role and group assignments for the user from the Identity Store
- Marks each of the user's target system accounts as DISABLED and sets an end date for the account according to the domain configuration parameter "maximum time to

disable an object"

The Provisioning workflows to the target system disable the accounts the next time they run. When the end date of the accounts is reached, the target system integrator attempts to delete the accounts and their memberships in the groups. If it successfully performs these operations in the target systems, it deletes the user entry in DirX Identity. If an operation fails in one target system (perhaps an account cannot be deleted) then the user is not deleted. In this case, if one of the accounts still exists after "maximum time to delete an object" has expired, the workflows that maintain the Identity Store (Provisioning Configuration) will mark the account as DELETED and set a delete date for the account according to the domain configuration parameter "maximum time to delete an object". The maintenance workflow will delete the user and the accounts if their delete dates have arrived. The maintenance workflow will also delete a user if all its accounts have been removed.

Disabling accounts during this user grace period allows you to undelete a user and re-enable its accounts and their group memberships. You can undelete a user by setting an end date that lies in the future. DirX Identity then marks the user as ENABLED.

3.2.4. Changing User Attributes with DirX Identity Manager

When DirX Identity is the master of the user store, keeping user attributes up-to-date is a primary administrative task.

To change a user's attributes with DirX Identity Manager:

1. Select the user. You can select the user by browsing the tree or you can use the Manager's user search dialog.
2. Click the General tab to display the user's attributes.
3. Click **Edit** to change one or more attributes.



Changing an attribute can result in changes to access rights (permission parameters).

You can use a user's attributes to:

- Set the user's end date; for example, the date at which an employee is to leave the company
- Set a date at which the user is to become inactive; for example, the date at which an employee is to start a leave of absence. When this date occurs, DirX Identity assigns the user the DISABLED state.
- Set a date at which the user is to be re-activated; for example, the date at which an employee is to return from a leave of absence.
- Change a user template to a real user. When you make this change, DirX Identity changes the user's state from TEMPLATE to ENABLED (if you do not specify a start date) or to NEW (if you specify a future start date).

Manager keeps the changes you make in its internal cache until you use the **Save** function.

When you change tabs, Manager resolves the new access rights in its internal cache. As long as you do not use **Save**, you can use the Manager's **Reset** function to restore the settings to their old values (the values that are stored in the DirX Identity store). When you select the **Save** function, Manager first stores the changes (attributes and changed assignments) in the Identity Store and then tries to resolve the new access rights if the user is not currently resolved by another process.

If a resolution error occurs, you can use **Cancel**, and then modify the attributes or assignments to correct the problem. In some cases, however, it is possible that you cannot fix the error. If you choose to store your changes despite the resolution error, Manager:

- Stores only the current attribute values
- Retains the old access rights
- Marks the user entry as inconsistent
- Stores a description of the error with the user entry in the Identity Store
- Sets an end date after which the access rights will be calculated and stored regardless of the error status

3.2.5. Changing the User Subtree with DirX Identity Manager

The basic structure of the user subtree is established during DirX Identity Provisioning system setup and often reflects a corporate user directory (or mirrors it, if the corporate directory is the master). Changes to the user subtree should be considered carefully. For example, suppose the user subtree has been set up to reflect administrative responsibilities or organizational responsibilities. You might need to change it to accommodate changes to administrative tasks or organizational entities.

In the hierarchical tree view of the users, you can:

- Add a new user folder
- Copy a user or a user folder
- Move a user or a user folder
- Delete a user or a user folder
- Rename a user or a user folder

The user subtree can contain the following folder types:

- Country
- Organization
- Organizational unit
- Location
- Domain component

To add a new folder:

1. Click the user root or one of the folders below it, and then select **New** → *folder_type* from

the context menu. The folder types that Manager offers for creation depend on your current position in the subtree.

2. Select a folder type. Manager prompts you to enter the unique name for the folder.

To delete a user, a user template, or a folder, click it, and then select **Delete** from the context menu or menu bar.

To rename a user or a folder, click it, then select **Rename** from the context menu or menu bar.

To copy a user, a user template, or a folder, click it, then select **Copy Object** from the context menu or menu bar.

To move a user, a user template, or a folder, click it, then select **Move Object** from the context menu or menu bar.

3.3. Managing the Users (Another Directory is Master)

Managing the DirX Identity user store (Provisioning Configuration) when another directory is the master consists of one required task:

- Configuring the relevant creation workflow that periodically synchronizes the DirX Identity user store (Provisioning Configuration) with the state of the other user master directory

Configuring the creation workflow is a domain configuration task. It is described in more detail in the chapter "Managing Domains". Once the workflow has been configured, it runs at the specified schedule and automatically updates the Identity Store with changes from the other user master directory.

Optional user management tasks when a corporate directory is master can include:

- Creating user attributes that are specific to DirX Identity Provisioning and managing them from within DirX Identity. DirX Identity masters these attributes and the administrator uses DirX Identity Manager to maintain them.
- Setting up a workflow to synchronize the DirX Identity-specific user attributes with the other directory.
- Structuring the user subtree in DirX Identity to mirror the organization in the other user directory. The administrator uses DirX Identity Manager to accomplish this task.

3.4. Assigning Roles to Users

To assign roles to users and user templates, you can use DirX Identity Manager, DirX Identity Business User Interface (BUI), or DirX Identity Web Center. The user interfaces for these applications differ from each other but provide nearly the same functionality.

To manage user-to-role assignment with DirX Identity Manager:

1. Select a user, either by browsing the user tree or using the Manager's user search dialog.
2. Click the Assigned Roles tab to display the user's current role assignments.



The role assignment function (Assigned Roles tab) is only visible when the Pro Suite is installed (requires an extra license).

Now you can:

- Select a role and add it to the user's assigned roles. You can use the search dialog to reduce the number of assignable roles that are displayed. If a role has role parameters, you must set these parameters during the assignment process.
- Select an assigned role and remove it from the list of assigned roles.
- Specify or change a lifetime for a role assignment (a start date and an end date)

When you click **Save**, DirX Identity Manager:

- Checks if SoD is enabled at the domain configuration object. If SoD is enabled, it checks if one of the roles is subject of an SoD policy and if so, it warns the user that this assignment will start an SoD approval workflow. The user can cancel this assignment or request the approval process.
- Starts approval workflows if a role is marked with **Requires approval**.



Privilege assignments that are still in approval are not resolved until successful approval. These assignment lines are displayed in grey until they are approved.

When a privilege does not require approval, DirX Identity Manager stores the assignment. To avoid inconsistencies, DirX Identity Manager tries to lock the user. If this is not possible because another process is currently storing access right relevant changes, DirX Identity Manager informs you and you can try later.

Next, DirX Identity Manager tries to perform privilege resolution. If the user is locked because the Resolution Adapter is currently resolving the same user, DirX Identity Manager sends a resolve message and the Resolution Adapter will resolve the user as soon as possible.



Neither the Business User Interface nor DirX Identity Web Center resolve the user themselves. They just send the resolve message.

Privilege resolution by the DirX Identity Manager or the Resolution Adapter:

- Resolves the user-group relationships, where it evaluates the role hierarchy from parent and junior roles to permissions to the target system groups and applies matching rules for role parameters and permissions.
- Creates the necessary new target system accounts. DirX Identity names each account according to the naming rule of the object description for the target system's accounts.

(Each target system has an object description for its accounts that is related to domain specific object descriptions and naming rules - see the import statement in the object description.)

- Updates the access rights, which means it adds or removes account - group memberships.
- Stores the changes in the DirX Identity Store.
- For target systems for which realtime provisioning is enabled, it triggers the Provisioning workflows that immediately apply the necessary changes in the connected systems.

If a resolution error occurs, the resolution process:

- Retains the old access rights
- Marks the user entry as inconsistent
- Stores a description of the error with the user entry in the Identity Store.

When a role assignment's start date is in the future, the resolution creates the necessary accounts in the target system(s) with state DISABLED and creates the account – group memberships. Advance account creation for a future role assignment is designed to give the target system administrator time to perform any local administrative tasks associated with the accounts; for example, creating a mailbox server or establishing a share.When the role assignment's start date occurs, the regularly started consistency workflow enables the target system accounts (changes their state from DISABLED to ENABLED). With realtime provisioning enabled, it sends the message to start the Provisioning workflow and apply the changes to the connected system. Otherwise, the changes will be applied at the next scheduled run of the workflow.

3.5. Assigning Permissions to Users

Assigning a permission to a user is analogous to a role assignment.

To manage direct user-to-permission assignment with Manager:

1. Select a user, either by browsing the user tree or using the Manager's user search dialog.
2. Click the Assigned Permissions tab to display the user's current permission assignments.

Now you can use Manager to:

- Select a permission and add it to the user's assigned permissions.
- Select an assigned permission and remove it from the user's assigned permissions.
- Specify or change a lifetime for the permission assignment (a start date and an end date)

When you use **Save** or change to another tab, Manager performs the tasks described in "Assigning Roles to Users".



Permissions can also be assigned automatically (for example, by rules) or inherited from a role.You cannot remove these permissions manually.

3.6. Assigning Groups to Users

Assigning a group to a user is analogous to a role assignment.

You can make a direct assignment of a group to a user.

To manage direct user-to-group assignment with Manager:

1. Select a user, either by browsing the user tree or using the Manager's user search dialog.
2. Click the Assigned Groups tab to display the user's current group assignments.

Now you can use Manager to:

- Select a group and add it to the user's assigned groups.
- Select an assigned group and remove it from the user's assigned groups.
- Specify or change a lifetime for the group assignment (a start date and an end date).

When you use **Save** or change to another tab, Manager performs the tasks described in "Assigning Roles to Users".



Groups can also be assigned automatically (for example, by rules) or inherited from a permission.You cannot remove these groups manually.

3.7. Working with User Query Folders

Query folders are a tool that you can use to search for and identify users that need some administrative action to be taken, such as users that need to be assigned privileges, or to find users that have similar properties, for example, users located in Munich. When you open a query folder, it immediately performs a search on the DirX Identity store (Provisioning Configuration) according to the values in its Filter tab and displays the results as objects underneath the query folder. In the users view, you can:

- Add a new query folder
- Copy a query folder (and then change it to your requirements)
- Delete a query folder
- Rename a query folder

DirX Identity provides the following default query folders in the users view:

- Error - identifies all users who have messages in their **Error** field
- To Do - identifies all users who have messages in their **To do** field
- Inconsistent - identifies all users in an inconsistent state

- To Be Deleted - identifies all users in the TBDEL state

The topic "Using Query Folders to Detect Inconsistencies" provides more information on how to use the default query folders.

3.8. Working with User States

The DirX Identity Provisioning system recognizes the following user states, which it uses to maintain the consistency of the DirX Identity store (Provisioning Configuration):

- NEW - the user has been added to the user subtree but is not yet activated.
- TEMPLATE - the user has been created as a user template.
- ENABLED - the user has been activated (its start date has arrived).
- DISABLED - the user has been deactivated.All of the accounts associated with the user are also disabled.
- TBDEL - the user's end date has arrived, and DirX Identity has marked the user for deletion

The **Status** field in the Operational tab for a user displays the user's current state. In the tree view, the current status of a user is displayed in brackets at each user entry if the user is not in the ENABLED state. For detailed information about user states, see the section "Managing States" in the chapter "Managing Provisioning".

3.9. Working with Links at User Entries

You can link user entries to various objects in the Identity Store. The DirX Identity services also use some of the links to keep specific information. This section describes the most important default links of a user object, their purpose and how you can use them.

The following links can be used to calculate approvers or to send mail to these related users. If you need more user links, extend the schema accordingly.

dxrRepresentative
: the representative for a user.

dxrSponsor
: the person who is responsible for another person, for example, a contractor.Use dxrSponsor to link a functional user to its responsible user.

manager
: the user's manager.

owner
: the owner of another user entry; for example, a functional account that is treated as a user entry and not as a privileged account. Use owner to link a persona to its related user.

secretary

the user's secretary.

Links to privileges and accounts include:

dxrAssignedAccount

points to all accounts of this user.

dxrGroupLink

keeps all manual (direct) assignments of groups (the source for the dxrPrivilegesGrantedLink).

dxrPermissionLink

keeps all manual (direct) assignments of permissions (the source for the dxrPrivilegesGrantedLink).

dxrRoleLink

keeps all manual (direct) assignments of roles (the source for the dxrPrivilegesGrantedLink).

dxrInheritedPrivilegeLink

keeps all inherited assignments related to business objects (the source for the dxrPrivilegesGrantedLink)

dxrPrivilegeLink

keeps all privileges that were assigned by a rule (the source for the dxrPrivilegesGrantedLink).

dxrPrivilegesGrantedLink

keeps the sum of all approved privileges the user currently has regardless of time constraints. These are the privileges assigned by hand, by rule, from business objects or inherited from privilege resolution (junior roles, permissions, groups). This attribute has been designed primarily for SoD checking. It also contains privileges whose assignment start date is not reached, otherwise approaching a start date would require an SoD check.



For privileges whose start date is not yet reached, either group memberships are still created but the account is in state DISABLED or (if the account is used for other group memberships) the account is ENABLED but the memberships are not yet created.



Privilege assignments that are still in approval do not appear in any of the above-mentioned attributes. This information is kept only in the corresponding request workflow instances. Virtual attributes make the consolidated information available through the service layer.

Links to business objects include:

dxrContextLink
links to customer-specific business objects.

dxrCostUnitLink
links to cost unit objects.

dxrLocationLink
the main location of the user.

dxrOrganizationLink
the main organization of the user.

dxrOULink
the main organizational unit of the user.

dxrSecOULink
additional organizational units to which this user is attached.

dxrSecOrganizationLink
additional organizations to which this user is attached.

dxrSecLocationLink
additional locations to which this user is assigned.

Other links include:

dxrPwdPolicyLink
a link to a specific password policy that this user must follow.

The following links are used by the deprecated approval workflow engine and are no longer in use. Migration routines have been used to move the information from these links to the valid links described above.

- dxrAssignedGroups
- dxrAssignedPermissions
- dxrAssignedRoles
- dxrInheritedPermissions
- dxrNotApprovedPrivilege

4. Managing User Facets

User facets are special representations of users, so we recommend reading the section "Managing Users" to become familiar with the user management tasks that also apply to user facets.

User facets have the following basic features:

- User facets can have privileges assigned to them (roles, permissions, groups) that are maintained by the same processes as for users.
- The privilege resolution process does not apply to user facets. User facets do not have accounts and group assignments.
- Privileges assigned to a user facet are inherited by the user.
- User facets have the same states as users, but their life-cycle is controlled by the related user's life-cycle.

User facets have the following limitations:

- There is no support for parameterized roles.
- There is no SOD checking.
- There is no support for user facet - user switching.
- There is no rule for creating a user facet from an account.

A user facet is a special kind of DirX Identity user and thus generally functions in the same way as a DirX Identity user. This chapter highlights the differences between a user facet and a DirX Identity user and describes the features that apply only to a user facet. The following sections describe special aspects of user facet management, including:

- Where to locate user facets in the user tree
- How to view, add and delete user facet entries and change their attributes
- How to work with user facet states
- How to work with links at user facet entries
- How to work with user facet inheritance
- How to maintain user facets with DirX Identity workflows

For all other aspects of a user facet, the information about DirX Identity users given in "Managing Users" (and elsewhere in the DirX Identity documentation) applies.

4.1. Locating User Facets

User facet objects are located within the **cn=Users** subtree, together with user objects and functional users. Although placement of a user facet object anywhere in the subtree is permitted, it is recommended that the object be stored in the same folder as its corresponding user object, because it is tightly bound to that user and cannot exist independently.

4.2. Working with User Facets

Working with user facets consists of the following tasks:

- Viewing user facet properties
- Adding user facets to the Identity Store
- Deleting user facets from the Identity Store
- Changing the attributes of existing user facets

When DirX Identity masters the user facet data, you use DirX Identity Manager to perform these tasks by hand.

4.2.1. Viewing User Facets with DirX Identity Manager

When you log into DirX Identity Manager and then select Users from the view bar, DirX Identity Manager displays a hierarchical tree of the users, user facets and functional users that you are allowed to manage in the left-hand pane.

Users, user facets, personas and functional users are distinguished in the user tree by their different icons, as shown in the following figure:

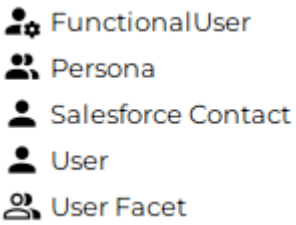


Figure 17. Functional User, Persona, User and User Facet Icons

To view the properties of a user facet, click its entry in the tree. It is displayed in the same tabs as a user.

If a user facet is not in the ENABLED state, its current status is displayed in brackets at its entry.



User facets have the same states as users. For more information about user states, see the section "User States" in the section "Managing States" in the chapter "Managing Provisioning" and the section "Working with User Facet States".

4.2.2. Adding User Facets with DirX Identity Manager

To add a new user facet with DirX Identity Manager:

1. Click a user folder in the **Users** subtree or click the top-level **Users** folder. When adding a user facet, we recommend adding it to the folder that contains the user to which the user facet belongs.
2. Select **New** -> **User facet** in the context menu. The **General** tab is displayed for editing,

and the mandatory attributes for a user (the user's common name (cn) and surname (sn)) are displayed in red.

3. Click the **Relationships** tab and then select the user facet's owner (this is the user that is connected with the user facet).

You can also use Web Center to create a user facet. In Web Center, user facet creation is performed by a request workflow with an activity that uses the related user as a template for creating the user facet. You can configure the attributes to be copied from the owner and the location at which to create the user facet. See the section "Using the Users Menu" in the chapter "Using DirX Identity Web Center" in the *DirX Identity User Interfaces Guide* for details.

4.2.2.1. Specifying a User Facet Lifetime

When you add a new user facet, you can specify a user facet lifetime: a start and end date. Define the start and end dates for the user facet as you would for users. However, remember that the user facet's life-cycle is related to the user's life-cycle, so that:

- Disabling the user facet's owner also disables the user facet, unless it is in state TBDEL.
- If the user's state changes to TBDEL, all his related user facets also change their states to TBDEL and their delete dates are set.
- If you delete a user facet, it does not affect the user's state.

4.2.2.2. Working with User Facet Templates

Working with user facet templates is not recommended.

4.2.3. Deleting User Facets with DirX Identity Manager

To delete a user facet, click it and then select **Delete** from the menu bar or context menu. The delete process for user facets is the same as for users.See the section "Deleting Users with DirX Identity Manager" for details.

4.2.4. Changing User Facet Attributes with DirX Identity Manager

You change a user facet's attributes as you would a user's attributes, using the same available tabs. See the section "Changing User Attributes with DirX Identity Manager" for details.

A set of user facet attributes is mastered from its owner, the related user. You can't edit these attributes at the user facet, but they are automatically updated to the user's values when the user facet is saved. If an attribute that is mastered to the user facet is changed during a user edit or if a state change occurs, the related user facets are updated with the new values.

4.3. Working with User Facet States

The DirX Identity Provisioning system recognizes the same states for user facets as for users. For detailed information about user facet states, see the section "Managing States" in

the chapter "Managing Provisioning".

4.4. Working with Links at User Facet Entries

You can link user facet entries to the same objects as user entries. See the section "Working with Links at User Entries" in the chapter "Managing Users" for details.

The owner link is of special importance for a user facet: it contains the reference to its related user.

4.5. Working with User Facet Inheritance

Privileges assigned to a user facet are inherited by the user. These privilege assignments are marked with the notation **UF** in Web Center and DirX Identity Manager. If the user is disabled, the privilege assignments inherited from its user facets remain.Only its associated accounts are disabled.

User facet inheritance only applies to user facets in the ENABLED state, and only those privilege assignments in the ENABLED state are inherited by the user. Disabling a user facet revokes the privilege assignments unless the user is also disabled.

Creating a user facet with a startDate in the future results in the state NEW. No privilege inheritance takes place. When the startDate is reached and a maintenance workflow for this user facet runs, the user facet is ENABLED and its privileges are inherited by the user.

Privileges that needed to be approved during their assignment to the user facet do not need to be re-approved during user facet inheritance.

Permission match rules are applied to the user, not to the user facet. Attributes of the user facet do not influence evaluation of permission match rules.

4.6. Maintaining User Facets with DirX Identity Workflows

New DirX Identity maintenance workflows for user facets need to be created and configured. For details, see the DirX Identity Use Case Document *Configuring the Maintenance Workflows for User Facets*.

5. Managing Personas

Personas are special representations of users, so we recommend reading the section "Managing Users" to become familiar with the user management tasks that also apply to personas. The following user functionality also applies to personas:

- Personas may have privileges (roles, permissions, groups) assigned that are maintained by the same processes as for users.
- The privilege resolution process also applies to personas, resulting in creation of accounts and group assignments.
- Personas have the same states as users, but their life-cycle is controlled by the related user's life-cycle.

The following sections describe special aspects of persona management, including:

- Where to locate personas in the user tree.
- How to manage personas.
- How to work with persona states.
- How to work with links at persona entries.

5.1. Locating Personas

Persona objects are stored under the **cn=Users subtree**, where they coexist with user objects and functional users. While persona objects may technically be placed anywhere within the **cn=Users** subtree, we recommend storing each persona in the same folder as its corresponding user object. This is because a persona is tightly coupled to its related user and cannot exist independently.

5.2. Working with Personas

Working with personas consists of the following tasks:

- Viewing persona properties
- Adding personas to the Identity Store
- Deleting personas from the Identity Store
- Changing the attributes of existing personas

When DirX Identity masters the persona data, you use DirX Identity Manager to perform these tasks by hand.

5.2.1. Viewing Personas with DirX Identity Manager

When you log into DirX Identity Manager and then select **Users** from the view bar, DirX Identity Manager displays a hierarchical tree of the users, personas and functional users that you are allowed to manage in the left-hand pane.

Users, personas and functional users are distinguished in the user tree by their different icons, as shown in the following figure:

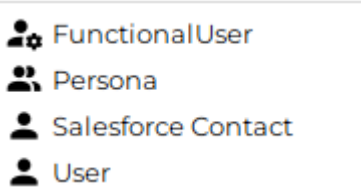


Figure 18. Functional User, Persona and User Icons

To view the properties of a persona, click its entry in the tree. It is displayed in the same tabs as a user.

If a persona is not in the ENABLED state, its current status is displayed in brackets at its entry. For more information about persona states, see the section "Persona States" in the section "Managing States" in the chapter "Managing Provisioning" and the section "Working with Persona States".

5.2.2. Adding Personas with DirX Identity Manager

To add a new persona with DirX Identity Manager:

1. Click a user folder in the **Users** subtree or click the top-level **Users** folder. When adding a persona, we recommend adding it to the folder that contains the user to which the persona belongs.
2. Select **New** -> **Persona** in the context menu. The **General** tab is displayed for editing, and the mandatory attributes for a user (the user's common name (cn) and surname (sn)) are displayed in red.
3. Click the **Relationships** tab and then select the persona's **owner** (this is the user that is connected with the persona).

You can also use Web Center to create a persona. In Web Center, persona creation is performed by a request workflow with an activity that uses the related user as a template for creating the persona. You can configure the attributes to be copied from the owner and the location at which to create the persona. See the section "Using the Users Menu" in the chapter "Using DirX Identity Web Center" in the *DirX Identity User Interfaces Guide* for details.

5.2.2.1. Specifying a Persona Lifetime

When you add a new persona, you can specify a persona lifetime: a start and end date.Define the start and end dates for the persona as you would for users.However, remember that the persona's life-cycle is related to the user's life-cycle, so that:

- Disabling the persona's owner also disables the persona, unless it is in state TBDEL.
- If the user's state changes to TBDEL, all his related personas also change their states to TBDEL and their delete dates are set.
- If you delete a persona, it does not affect the user's state.

5.2.2.2. Working with Persona Templates

Working with persona templates is not recommended.

5.2.3. Deleting Personas with DirX Identity Manager

To delete a persona, click it and then select Delete from the menu bar or context menu. The delete process for personas is the same as for users.See the section "Deleting Users with DirX Identity Manager" for details.

5.2.4. Changing Persona Attributes with DirX Identity Manager

You change a persona's attributes as you would a user's attributes, using the same available tabs. See the section "Changing User Attributes with DirX Identity Manager" for details.

A set of persona attributes is mastered from its owner, the related user. You can't edit these attributes at the persona, but they are automatically updated to the user's values when the persona is saved. If an attribute that is mastered to the persona is changed during a user edit or if a state change occurs, the related personas are updated with the new values.

5.3. Working with Persona States

The DirX Identity Provisioning system recognizes the same states for personas as for users:

- NEW - the persona has been added to the user subtree but is not yet activated.
- TEMPLATE - the persona has been created as a persona template.
- ENABLED - the persona has been activated (its start date has arrived).
- DISABLED - the persona has been deactivated. All of the accounts associated with the persona are also disabled.
- TBDEL - the persona end date has arrived, and DirX Identity has marked the persona for deletion.

The **Status** field in the **Operational** tab for a persona displays the persona's current state. In the tree view, the current status of a persona is displayed in brackets at each persona entry if the persona is not in the ENABLED state. For detailed information about persona states, see the section "Managing States" in the chapter "Managing Provisioning".

5.4. Working with Links at Persona Entries

You can link persona entries to the same objects as user entries. See the section "Working with Links at User Entries" in the chapter "Managing Users" for details.

The owner link is of special importance for a persona: it contains the reference to its related user.

6. Managing Functional Users

Functional users are special representations of users, so we recommend reading the section "Managing Users" to become familiar with the user management tasks that also apply to functional users. The following user functionality also applies to functional users:

- Functional users can have privileges (roles, permissions, groups) assigned that are maintained by the same processes as for users.
- The privilege resolution process also applies to functional users, resulting in the creation of accounts and group assignments.
- Functional users have the same life-cycle and the same states as users.

The following sections describe special aspects of functional users, including:

- Where to locate functional users in the user tree.
- How to manage functional users.
- How to work with functional users states.
- How to work with links at functional user entries.

6.1. Locating Functional Users

Functional User objects reside under the **cn=Users** sub tree, where they are mixed with user objects and persona objects that also populate this tree.

Functional users are independent objects that are loosely coupled to user objects via their sponsor links.If the user no longer exists or is no longer active, it makes sense to re-link the functional user object to another sponsor.As a result, functional users can be created anywhere in the subtree.Re-linking a sponsor object to a functional user does not require the functional user to be moved, but it is of course possible if necessary.We recommend that you locate functional users in their own folder; for example, **ou=Resources**.

6.2. Working with Functional Users

Working with functional users includes:

- Viewing functional user properties
- Adding functional users to the Identity Store
- Deleting functional users from the Identity Store
- Changing the attributes of existing functional users

You use DirX Identity Manager to perform these tasks by hand.

6.2.1. Viewing Functional Users with DirX Identity Manager

When you log into DirX Identity Manager and select **Users** from the view bar, DirX Identity Manager displays a hierarchical tree of the users, personas and functional users that you

are allowed to manage in the left-hand pane. Users, personas and functional users are distinguished in the user tree by their icons, as shown in the following figure:

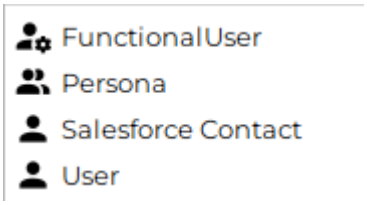


Figure 19. Functional User, Persona and User Icons

To view the properties of a functional user, click its entry in the tree. It is displayed in the same tabs as a user.

If a functional user is not in the ENABLED state, its current status is displayed in brackets at its entry. For more information about functional user states, see the section "Functional User States" in the section "Managing States" in the chapter "Managing Provisioning" and the section "Working with Functional Users States".

6.2.2. Adding Functional Users with DirX Identity Manager

To add a new functional user with DirX Identity Manager:

1. Click a user folder in the subtree or click the top-level User folder. We recommend using a specific folder - for example, **ou=Resources** - to store the functional users separately from users and personas.
2. Select **New** → **Functional User** in the context menu. The **General** tab is displayed for editing, and the mandatory attributes for a user (the user's common name (cn) and surname (sn)) are displayed in red.
3. Change to the **Relationships** tab and then select the functional user's **sponsor** (that's the user who is responsible for the functional user).

You can also use Web Center to create a functional user is to create it in Web Center; in fact, this is the recommended way to create them. Creating a functional user in Web Center is performed by a request workflow with an activity that uses the selected sponsor as a template for creating the functional user. Its parent folder and the attributes to be copied from the sponsor are configured in the workflow definition. See the section "Using the Users Menu" in the chapter "Using DirX Identity Web Center" in the *DirX Identity User Interfaces Guide* for details.

When you add a functional user, you can:

- Specify a functional user lifetime (a start and an end date)
- Create a functional user template

6.2.2.1. Specifying a Functional User Lifetime

You can specify a functional user lifetime by defining start and end dates for the functional user as you can for users. In contrast to personas, the functional user's life-cycle is not related to the life-cycle of the sponsor. If the sponsor's state changes to TBDEL, DirX

Identity removes the sponsor link is from all related functional users and starts request workflows for assigning a new sponsor.

6.2.2.2. Creating Functional User Templates

When you create a new functional user, you can create it as a functional user template by checking **Use as Template**. Note that you must provide a sponsor for the functional user template since the sponsor is a mandatory attribute. If you do not want to define a sponsor for the template, just remove the template's dxrSponsor attribute in the Data View.

6.2.3. Deleting Functional Users with DirX Identity Manager

To delete a functional user, click it and then select **Delete** from the menu bar or context menu.The delete process for functional users is the same as for users.See the section "Deleting Users with DirX Identity Manager" for details.

6.2.4. Changing Functional Users Attributes with DirX Identity Manager

You change a functional user's attributes as you would a user's attributes; the same tabs are provided.See the section "Changing User Attributes with DirX Identity Manager" for details.

A set of functional user attributes can be mastered from its sponsor.These attributes cannot be edited at the functional user but are automatically updated to the user's values when the functional user is saved.

In addition, if an attribute that is mastered to the functional user is changed during a user edit, the related functional users are updated with the new values.

6.3. Working with Functional User States

The DirX Identity Provisioning system recognizes the same states for functional users as for users:

- NEW - the functional user has been added to the user subtree but is not yet activated.
- TEMPLATE - the functional user has been created as a functional user template.
- ENABLED - the functional user has been activated (its start date has arrived).
- DISABLED - the functional user has been deactivated.All of the accounts associated with the functional user are also disabled.
- TBDEL - the functional user end date has arrived, and DirX Identity has marked the functional user for deletion.

The **Status** field in the **Operational** tab for a functional user displays the functional user's current state.In the tree view, the current status of a functional user is displayed in brackets at each functional user entry if the functional user is not in the ENABLED state.For detailed information about functional user states, see the section "Managing States" in the chapter "Managing Provisioning".

6.4. Working with Links at Functional User Entries

You can link functional user entries to the same objects as user entries.See the section "Working with Links at User Entries" in the chapter "Managing Users" for details.

The **sponsor** link is of special importance for a functional user: it contains the reference to its responsible user.

7. Managing Business Objects

Business objects can be used to keep common information in a centrally located individual object. Examples of business objects include organizations, organizational units, countries, locations, and projects. Each business object has related information like a description, an address, and central communication data. A lot of this data can be part of user entries, and you can use business objects to help maintain this data consistently.

The source of business objects can be external data stores. You can use DirX Identity's synchronization features to keep the data synchronized.

Business object management consists of the following tasks:

- Maintaining an accurate and up-to-date directory of central data that can be used by user entries.
- Optionally setting up workflows to synchronize data from external data sources.
- Setting up event-based maintenance workflows that inherit privileges from business objects to users. See the chapter "Entry Change Workflows" in the *DirX Identity Application Development Guide* for more information.
- Setting up event-based maintenance workflows that propagate attribute changes from business objects to users. See the chapter "Entry Change Workflows" in the *DirX Identity Application Development Guide* for more information.

7.1. Working with Business Objects

When you log into DirX Identity Provisioning and select **Business Objects** from the view bar, DirX Identity displays hierarchical trees of the business objects that you are allowed to manage in the left-hand pane. The following figure provides an example of this hierarchical tree.

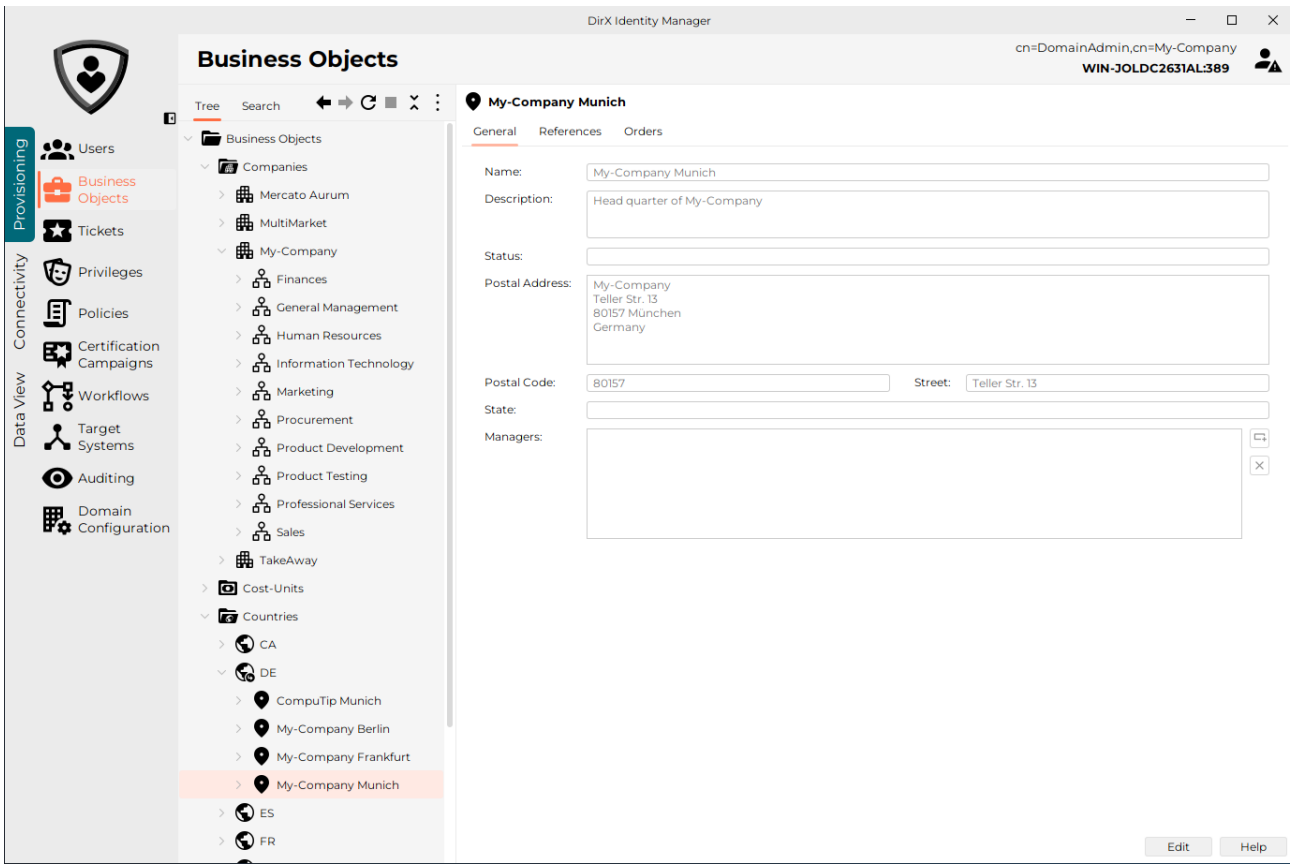


Figure 20. Viewing the Business Object Structure with DirX Identity Manager

By default, the hierarchical tree represented in the DirX Identity sample domain provides trees for companies, cost-units, countries, and projects.You can create your own custom objects under the tree node **Custom**.

To view the properties of a business object, click its entry in the tree. DirX Identity also provides a search dialog that you can use to select and display a subset of these objects. When you select a business object, DirX Identity displays a property dialog for the object. The dialog typically consists of a set of tabs that you can use to view the object's properties. Click the tabs in the property dialog to move between the different property categories.

7.2. About Business Object Types

DirX Identity comes with a set of standard business object types:

Companies - this tree can contain company (**organization**) objects including **organizational unit** objects. Use it to model your company's organizational structure.

Cost-Units - this tree can contain a company's cost center information.Use it to model your company's cost-unit structure.

Countries - this tree can contain **country** and **location** objects.Use it to model your company's regional distribution.

Projects - this tree can contain company **project** objects.Use it to define the various projects running in your organization.

You can use neutral folder objects to structure your object trees and you can change the look of the predefined business objects and add or remove attributes and links.

Use the **Custom** folder to define additional business objects of any type. Define the relevant object descriptions and create the instances in this subtree. Be sure to use the structural object type **dxrContext**, otherwise you cannot use the event-based workflows and the Web services for business objects. For custom attributes, set up auxiliary object classes.

7.3. Using Business Objects

Business objects help to simplify role assignment and reduce redundancy in your Identity Store.Information is kept once in one business object and is linked to (mastered by) other objects. Changing a business object triggers an immediate change of the mastered privileges and attributes.

Business objects typically come with a set of attributes (for example, the street and postal code for a locality). They also have links to other business objects or to privileges, which allows you to create complex linked structures of objects.

You can use business objects in several ways:

You can use a business object to automatically assign roles, permissions and groups referenced from the business object to all users linked to it. As a sample scenario, you may add the reference to a role to an organizational unit. This change triggers the appropriate event-based maintenance workflow, which automatically assigns this role to all users linked to the organizational unit. Working the other way, if you remove this role from the organizational unit, the same real-time workflow removes the role from all associated users.See the chapter "Entry Change Workflows" in the *DirX Identity Application Development Guide* for more information.



You cannot use roles with role parameters or privileges that are marked for approval.Web Center prohibits assignment of such privileges to business objects. DirX Identity Manager and **metacp** do not protect assigning such privileges.

You can use business object data in user attributes, and master data such as "address" or "street" from the business object. If you change the user object, this data is updated via DirX Identity's master mechanism from the linked object. If you change the business object, an event-based workflow can help to synchronize the changed attribute to all linked user objects. See the chapter "Entry Change Workflows" in the *DirX Identity Application Development Guide* for more information.

You can use business objects as a source for role parameters. To use a business object as a role parameter, define it as a DN or a Hierarchical DN type.

You can use business objects as a source for proposal lists.

8. Managing Tickets

Tickets support the concept of scheduled change management: scheduling a change to an identity management object for some time in the future instead of performing the change right away. A ticket system also provides a way to automate and manage cascading changes to inter-related identity management objects that must be carried out as a result of a change to one of the related objects.

Ticket management tasks generally include:

- Creating new tickets for changes to identity management objects
- Viewing tickets to check their status and content
- Making modifications to tickets; for example, to correct errors in ticket data that are hindering ticket processing
- Processing tickets
- Deleting tickets

DirX Identity provides a built-in ticket system that can manage tickets for all standard objects - users, business objects, privileges and so on - and also supports the integration of external ticket systems and target system service management scenarios.

The next sections provide conceptual information about DirX Identity's internal ticket system not covered in other DirX Identity documents and outlines DirX Identity support for customer-specific tickets.

For detailed information about DirX Identity's internal ticket system and how to integrate service management scenarios into DirX Identity, see the DirX Identity Use Case Document *Service Management*.

8.1. Understanding Internal Tickets

DirX Identity's internal service management component is its built-in ticket system, which allows you to specify the changes to be made to an object and then specify a due date for when the change is to occur. The ticket object provides temporary storage for the change data to be processed at the specified date, and the DirX Identity services process the change data automatically when the specified due date arrives.

This section describes the internal ticket system's design and operation, including:

- Ticket structure
- Ticket folder structure
- Ticket naming scheme
- Ticket life-cycle
- Ticket states

The section also describes how to view and manage internal ticket objects in DirX Identity

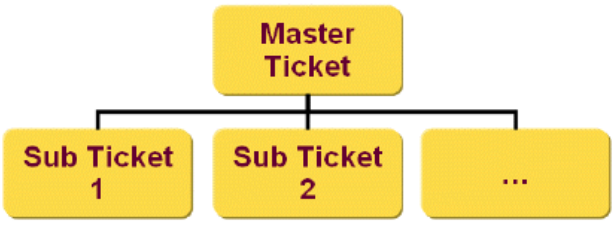
Manager's Provisioning view.

8.1.1. Understanding Internal Ticket Structure

DirX Identity's internal ticket system supports a two-tier ticket structure:

- The **master ticket** contains all changes to be directly applied at a specific date (the due date). If no direct changes are available, the master ticket acts as an informational ticket that connects the lower-level sub-tickets.
- The **sub-ticket** is linked to the master ticket and represents one or more attributes to be approved or an assignment to be approved.

The following figure shows the two-tier ticket structure and the relationship between object actions and ticket structure.



	Object Action				Ticket Structure					
No.	Attribute	Attribute with Approval	Assignment	Assignment with Approval	Master Ticket	Approval Workflow	Sub Ticket 1	Approval Workflow	Sub Ticket 2	Approval Workflow
1	X				Att					
2	X	X			Att		Att	X		
3	X		X		Att					
4	X			X	Att		Ass	X		
5		X			Att	X				
6		X	X		Ass		Att	X		
7		X		X	Info		Att	X	Ass	X
8			X		Ass					
9			X	X	Ass		Ass	X		
10				X	Ass	X				

Att = Attribute Order
Ass = Assignment Order
Info = Master ticket with no order

Figure 21. Ticket Structure

As shown in the table, the first four columns represent the four possible actions that can be performed on a complex object (for example, a user or a privilege):

- Attribute changes (one or more attributes) that do not require approval.
- Attribute changes (one or more attributes) that require approval. If the attributes changes require different approval workflows, the corresponding number of sub-tickets is created.
- Assignment changes (one assignment) that does not require approval.
- Assignment changes (one assignment) that requires approval. Each assignment change creates an individual approval workflow, which results in the corresponding number of sub-tickets.

The internal ticket system creates a master ticket and one or more sub-tickets depending on the combination of these actions (marked with an X in the corresponding columns). The table shows only a maximum of two sub-tickets. More are possible.

Some interesting cases are:

1 - Attribute change without approval: only creates a master ticket that is resolved at the due date.

6 - Attribute change with approval combined with an assignment without approval: the master ticket contains the assignment that is applied at the due date. The sub-ticket contains the attribute change that is to be approved first before it can be applied to the object.

7 - Attribute change with approval combined with an assignment with approval: in this case, an empty (or so called info) master ticket is created. It connects the two sub-tickets that each contains one of the required approvals.

10 - Assignment with approval: in this case, the sub-ticket is merged with the master ticket. This is only possible if there are no changes that need no approval and if there is only one sub-ticket available. Case 5 represents the same situation, but for an attribute approval.

As mentioned earlier in this section, a complex object change can consist of four possible actions, as shown in the following figure:

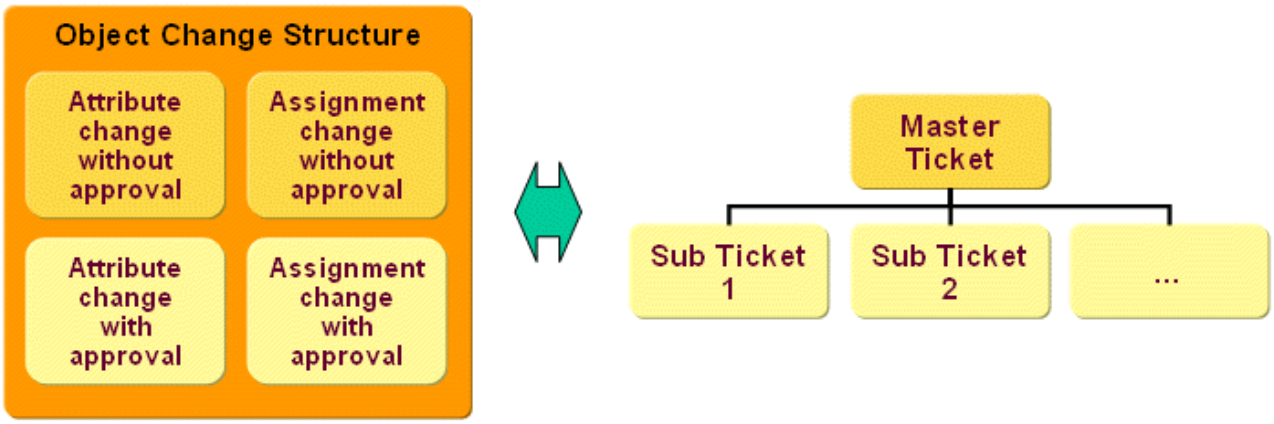


Figure 22. Relationship between Object Changes and Ticket Structure

The master ticket contains all changes without approval. The sub-tickets contain either an attribute change or an assignment to approve.

If changes without approval do not exist and there is only one potential sub-ticket, the resulting structure is a simple ticket with only one action to approve. The following figure illustrates this structure:

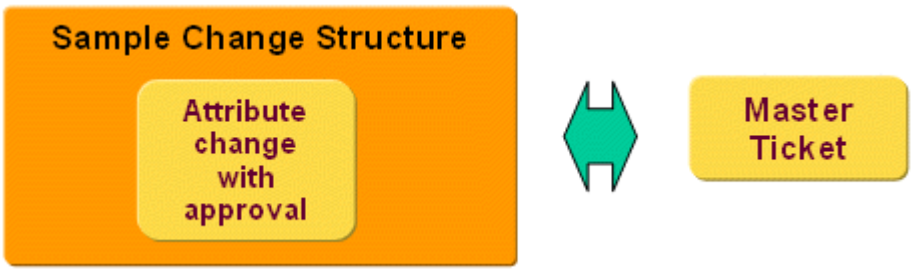


Figure 23. Simple Ticket

As shown in the figure, the sub-ticket is merged with the master ticket, which results in a simpler ticket structure.

8.1.2. Understanding Internal Ticket Folder Structure

Internal tickets are organized into the following folder structure in DirX Identity Provisioning's **Tickets** tree view:

- The tree for internal tickets starts at **Tickets** -> **Internal**.
- A type folder exists for each object type; for example, **Users**, **Roles**, and so on.
- Date folders that reflect the due date are created under each type folder; for example, **2025-09-05**.
- The date folder contains the ticket objects.

Here is an example:

Tickets → **Internal** → **Users** → **2025-09-05**

8.1.3. Understanding Internal Ticket Naming

DirX Identity's internal tickets are organized into master tickets and sub-tickets, as described in "Understanding Internal Ticket Structure".

DirX Identity names a master ticket according to the following scheme:

- The first part is the object name; for example, **Tinker Boris** for a user.
- The second part is created dynamically and contains the creation time; for example, **13:45:12**. This scheme allows for creating several tickets for the same object within the same day.
- If two tickets for the same object are created within the same second, DirX Identity automatically adds an extension; for example, **-1**.
- The ticket status is displayed in the tree and as a header in the property page (Manager only); for example, **Input.Completed**.

Here is an example of master ticket naming:

Tinker Boris 13:45:12 (Input.Completed)

Tinker Boris 13:45:12 -1 (Input.Completed)

Master tickets can contain sub-tickets. DirX Identity names a sub-ticket according to the following scheme:

- The first part is the operation; for example, **ADD**.
- The second part is the name of the resource; for example, **Analyst Reports**.

The two-tier naming scheme allows the master ticket to represent the subject, while the resources that are to approve are visible below it.

Here is an example of a sub-ticket name:

ADD Analyst Reports

8.1.4. Understanding the Internal Ticket Life-cycle

Administrators create tickets by supplying a **Due Date** value for the change in Web Center or DirX Identity Manager. DirX Identity processes - for example, request workflow activities - can also create tickets as required. You can find more information about how to create tickets at the user level in the following locations:

- The section "Working with Due Dates" in the chapter "Using the Web Center" in the *DirX Identity User Interfaces Guide*
- The section "Using the Provisioning View" in the chapter "Using DirX Identity Manager" in the *DirX Identity User Interfaces Guide*
- The section "Working with Internal Tickets" in the "Follow-on Tutorials" chapter of the *DirX Identity Tutorial*
- The context-sensitive help for objects displayed in DirX Identity Manager and Web Center

DirX Identity's Services layer creates a ticket automatically if a change order contains a due date. Internal tickets can be created for three operation types:

- **Object creation** - if object creation works with a request workflow, the ticket is created from the workflow's **ApplyChange** activity. Before this point, the object exists only as an order in the workflow.
- **Object modification** - the ticket is created immediately if a due date is present.
- **Object deletion** - the ticket is created immediately if a due date is present.

The scheduled **Process Internal Tickets** workflow regularly checks the state and due date of all tickets and processes tickets that are ready to process. You can find more information about this workflow in the following locations:

- The section "Working with Internal Tickets" in the *DirX Identity Tutorial* "Follow-on Tutorials" chapter, which demonstrates how to run this workflow by hand to initiate ticket processing.
- The "Internal Service Management "use case described in the DirX Identity Use Case Document *Service Management*, which describes how to set up and run this workflow as part of setting up the built-in DirX Identity ticket system.

- The section "Process Tickets Internal Workflow" in the chapter "Using the Maintenance Workflows" in the *DirX Identity Application Development Guide*, which provides configuration and operation information about this workflow.

A ticket is marked for deletion when a DirX Identity process sets its DELETED state and provides an expiration date. The DirX Identity Services layer uses the Cleanup Objects workflow to remove all tickets that are in state DELETED, that have an empty history and whose expiration dates have been reached or are blank. The Cleanup Workflow is a Tcl workflow that uses the Policy Agent to run consistency rules for object cleanup. It can be called explicitly or periodically by the scheduler. You can find more information about this workflow in the section "Cleanup Objects Workflow" in the chapter "Using the Maintenance Workflows" in the *DirX Identity Application Development Guide*.

8.1.5. Understanding Internal Ticket States

The following figure shows the possible ticket states and the state change flow.

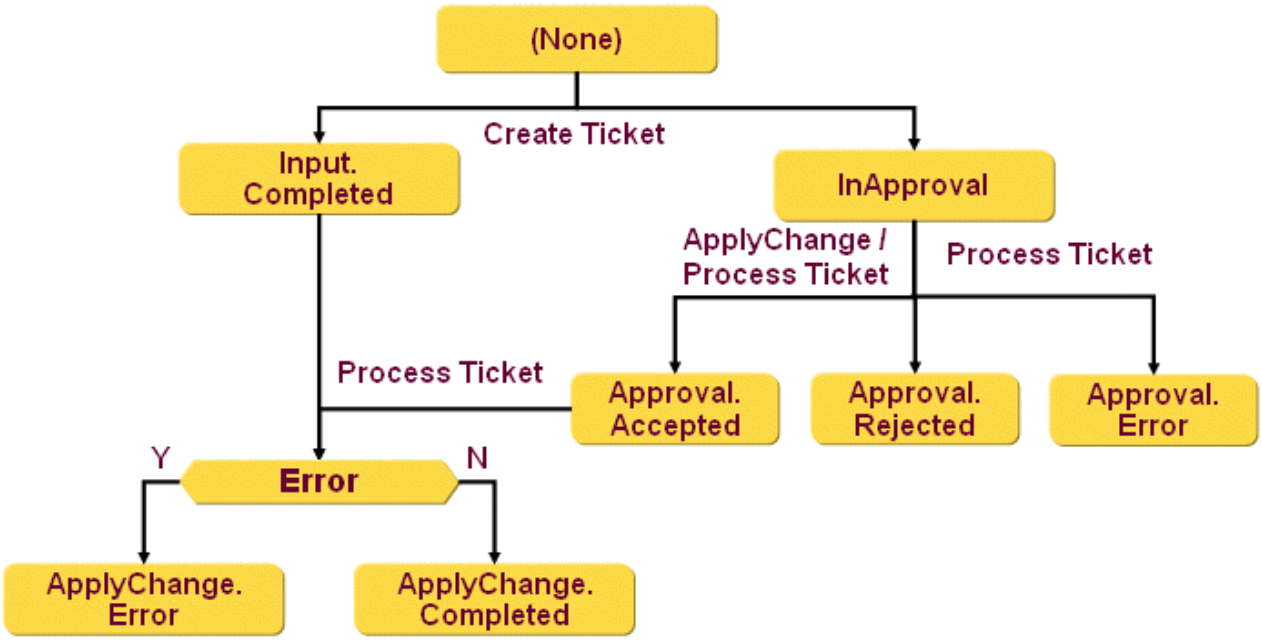


Figure 24. Ticket States

The ticket states are:

(None) - the service layer (or any service that uses it) creates a ticket either in **Input.Completed** or **InApproval** status. **Input.Completed** is used if no approval is necessary before the ticket is applied. **InApproval** is used if an approval is necessary. In this case, the ticket is linked with the corresponding approval workflow.

Input.Completed - the ticket is ready to be processed by the **Process Internal Tickets** service.

InApproval - the ticket requires the corresponding approval workflow to complete. If the approval is successful, the **ApplyChange** activity runs. This activity sets the ticket state to **Approval.Accepted** if the approval is successful (accepted). If the activity doesn't work correctly or the approval is not successful (rejected), the activity does not propagate this

situation. During the next run of the **Process Internal Tickets** service, the situation is detected and the state is set appropriately.

Approval.Accepted - the ticket state after an accepted approval. This ticket is processed again at the due date by the **Process Internal Tickets** service.

Approval.Rejected - the ticket state after a rejected approval. This is an end state.

Approval.Error - the ticket state after a failed approval workflow. This is an end state.

ApplyChange.Completed - the ticket state after successful ticket processing by the **Process Internal Tickets** service at the due date. All completed orders for attributes and assignments are resolved. This is an end state.

ApplyChange.Error - the ticket state after unsuccessful processing by the **Process Internal Tickets** service at the due date. This is an end state.

Use the default query folders provided for internal tickets in the **Tickets** → **Internal** → **_Queries** subtree (see "Working with Internal Ticket Objects" for details) to explore tickets of a specific state, especially tickets in the **ApplyChange.Error** state. Check the reason and then try to solve the problem.

8.1.6. Working with Internal Ticket Objects

When you log into DirX Identity Provisioning and select **Tickets** from the view bar, DirX Identity displays a hierarchical tree of tickets that you are allowed to manage in the left-hand pane. The following figure shows the Tickets view.

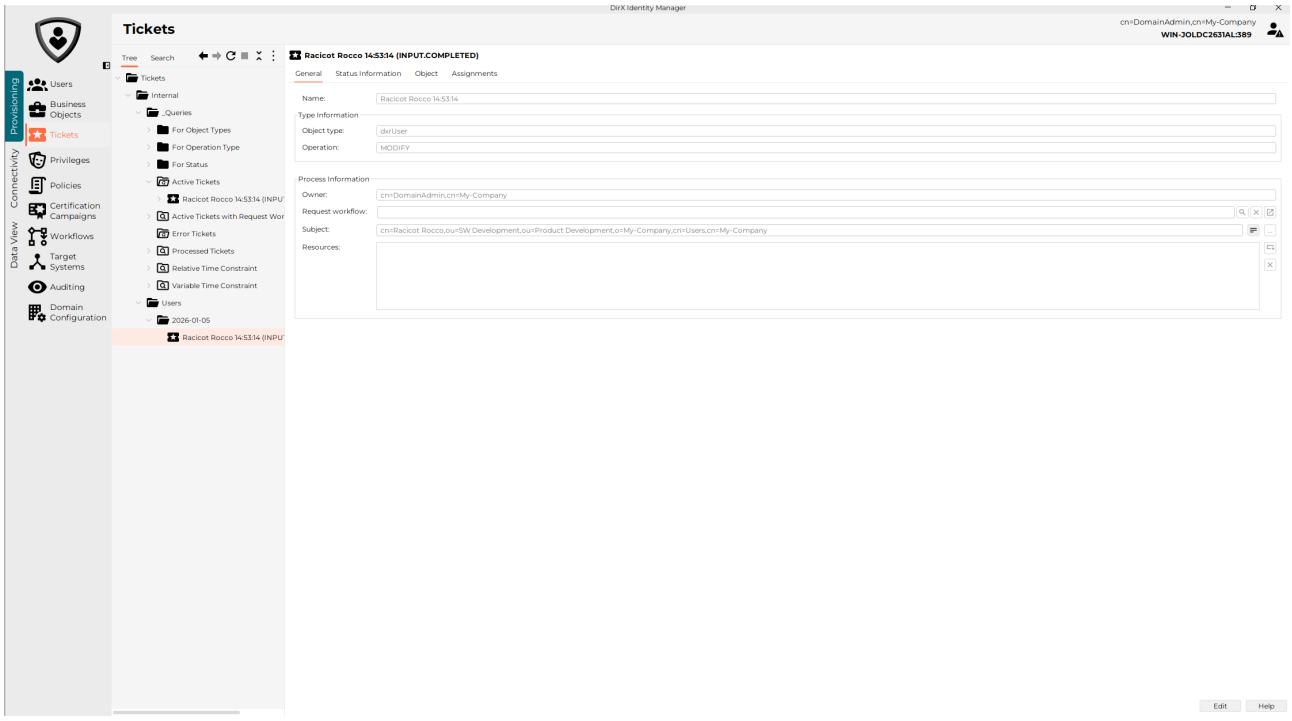


Figure 25. Viewing Internal Tickets with DirX Identity Manager

As shown in the figure:

- The **Internal** subtree displays DirX Identity's internal tickets.
- The **_Queries** subtree displays the queries you can run on internal tickets. DirX Identity provides a set of default queries; for example, to search the Internal tree for tickets with specific status, for active tickets, for error tickets and processed tickets, for tickets with time constraints and tickets that perform specific operations on specific object types. You can also create your own queries; see the topic "Creating a Query Folder" in the "Core Components" section of the DirX Identity Manager online help for details.
- The other folders in the **Internal** subtree are organized according to object types and due dates as described in "Understanding Internal Ticket Folder Structure" and named according to the scheme described in "Understanding Internal Ticket Naming".
- You can also view the ticket state in the tree; for example, **InApproval** or **Input.Complete**. See the section "Understanding Internal Ticket States" for details.



The Tickets tree view may display separate subtrees of customer-specific ticket types in addition to the **Internal** subtree, if external ticket systems have been integrated into DirX Identity.

To view the properties of a ticket, click its entry in the tree. DirX Identity also provides a search dialog that you can use to select and display a subset of these objects. For more information, see the Core Components" section of the DirX Identity Manager online help.

When you select a ticket object, DirX Identity displays the ticket state in the property dialog heading and a set of tabs that you can use to view (and edit) the ticket's properties. Click the tabs in the property dialog to move between the different property categories. See the context-sensitive help for information about internal ticket property dialogs and attributes.

To change the value or one or more of the properties displayed in a tab, click **Edit**.Manager keeps the changes you make to the object in its internal cache until you click **Save**. As long as you do not use **Save**, you can use the Manager's **Reset** function to restore the settings to their old values (the values that are stored in the Identity Store (Provisioning Configuration)).

8.2. About Customer-Specific Tickets

Customers can also create their own customized ticket solution. In the **DirX Identity Provisioning → Tickets view**, customer-specific tickets are displayed in separate customer subtrees at the same level as the DirX Identity-supplied **Internal** subtree.

The folder structure, naming schemes, states and life-cycle of these tickets - creation, modification, processing and deletion - is customer-specific. For ticket deletion, we recommend using a process that uses the Cleanup Objects workflow provided with DirX Identity. For more information about this workflow, see the section "Cleanup Objects Workflow" in the section "Understanding the Tcl-based Maintenance Workflows" in chapter "Using the Maintenance Workflows" in the *DirX Identity Application Development Guide*.

The section "Working with Source Tickets" in the *DirX Identity Tutorial* "Follow-on Tutorials" chapter demonstrates how to integrate an external "source" ticket system with DirX Identity's provisioning mechanism via a sample Web service. The *Service Management*

DirX Identity Use Case Description demonstrates this and other alternatives for integrating service management systems into DirX Identity.

9. Managing the Privilege Structure

Privilege structure management is the process of setting up the initial privilege structure; that is, creating the roles, the permissions, and the role-permission-group relationships, and then maintaining that structure. First, you follow the planning tasks outlined in the topic "Identify the Roles and Permissions" in "Planning for Provisioning". Then, you can use DirX Identity Manager to set up the privilege structure by hand.

The topics in this section describe how to use DirX Identity Manager to create and manage the Provisioning objects associated with a privilege structure.



The role structure is only visible when the optional Role Package is installed (requires an extra license).

9.1. Working with Privilege Structure Objects

When you log into DirX Identity Provisioning and select **Privileges** from the view bar, DirX Identity displays hierarchical trees of the roles, permissions, and the target system groups that you are allowed to manage in the left-hand pane.The tree displays the following privilege objects:

- Roles, ordered by role folders (only available if the optional Pro Suite is installed - requires an extra license).
- Permissions, ordered by permission folders (only available if the optional Pro Suite is installed - requires an extra license).
- Groups, ordered by target systems.

To view the properties of a privilege structure object, click its entry in the tree. DirX Identity also provides a search dialog that you can use to select and display a subset of these objects. When you select a privilege structure object, DirX Identity displays a property dialog for the object. The dialog typically consists of a set of tabs that you can use to view the object’s properties.Click the tabs in the property dialog to move between the different property categories. The following figure shows the Privileges view.

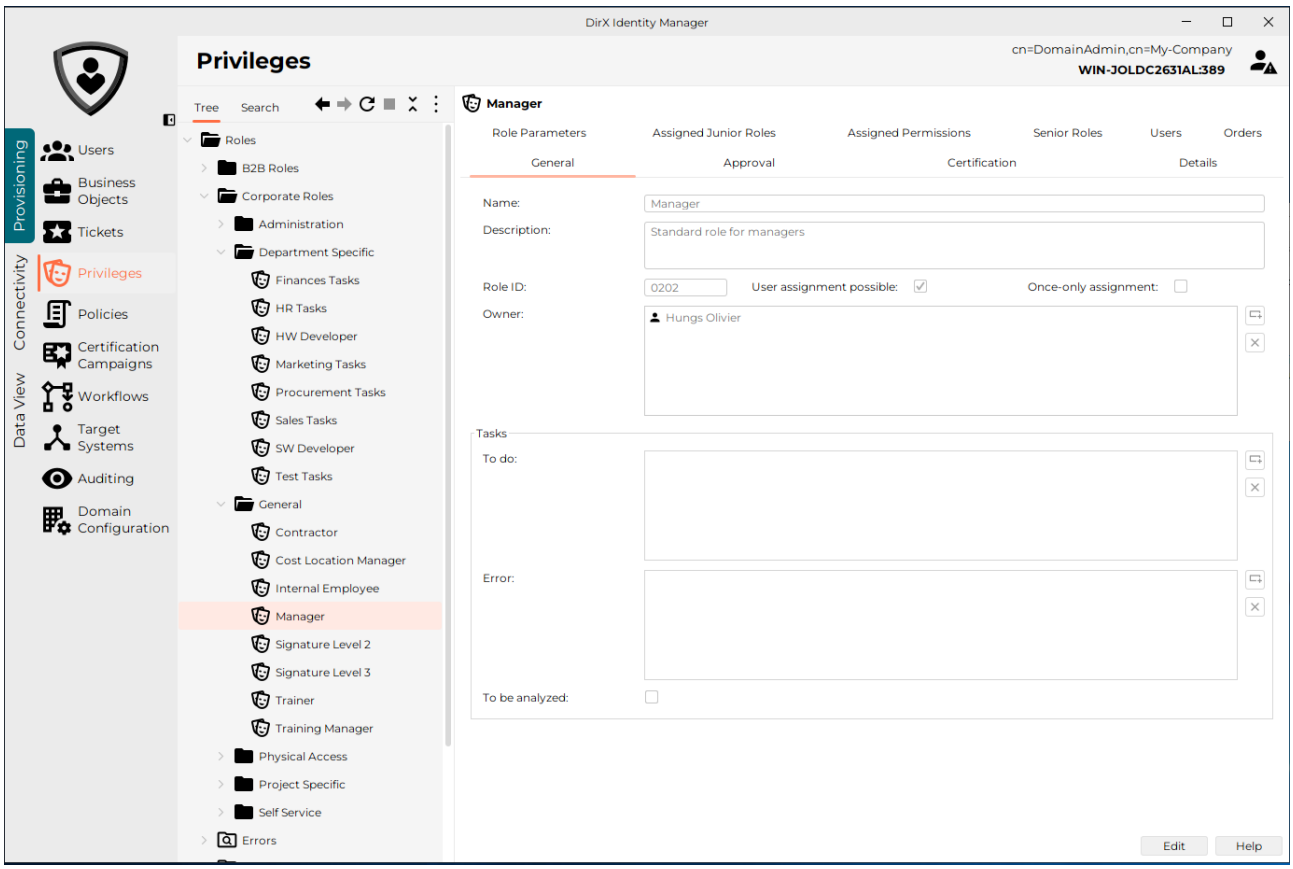


Figure 26. Viewing the Privilege Structure with DirX Identity Manager

To change the value or one or more of the properties displayed in a tab, click **Edit**. Manager keeps the changes you make to the object in its internal cache until you click **Save**. As long as you do not use **Save**, you can use the Manager's **Reset** function to restore the settings to their old values (the values that are stored in the Identity Store (Provisioning Configuration)).

When you click **Save** or change to another tab, Manager analyzes the changes you've made to the object for privilege resolution problems and other consistency problems. You can click **Offline** to interrupt the interactive privilege resolution process for later off-line processing (the topic "Managing the Privilege Resolution Process" provides further details). If you have clicked **Save**, Manager stores the changes in the Provisioning Configuration; if you have changed tabs, it stores the changes in its internal cache.Manager may return warnings and errors; the type and content depend upon the object you're changing and the type of changes you've made. If you receive warnings or errors, you can use **Cancel** to cancel your changes.You can also choose **Save**; in this case, DirX Identity Manager will mark the object as "inconsistent".Manager then saves the changes in the Provisioning Configuration. In the event that it is unable to save the data because of network failure or an LDAP server error or timeout, Manager notifies you of the error.

9.2. Managing Roles

Role management consists of the following tasks:

- Adding new roles

- Assigning a matching rule (working with role parameters)
- Creating and maintaining a role hierarchy (junior and senior roles)
- Assigning permissions to roles
- Changing the attributes of existing roles
- Deleting roles
- Organizing the role subtree



The Roles tree is only available when the optional Roles Package is installed (requires an extra license).

9.2.1. Adding a Role

To add a new role:

1. Click a role folder in the role subtree or click the top-level Role folder.
2. Select **New** → **Role** in the context menu.
3. Enter a name for the role (common name (cn)) and any other general role attributes; for example, a role type.

Once you have added a role to the role subtree, you can copy it, move it, or rename it. To copy a role, click it, then select **Copy Object** from the context menu or menu bar. To move a role, click it, then select **Move Object** from the context menu or menu bar. To rename a role, click it, then select **Rename** from the context menu or menu bar.

9.2.2. Assigning Role Parameters

Each role can be associated with one or more role parameters. Role parameters help to reduce the number of roles in a domain.

For example, suppose your organization manages a number of projects and wants to assign a project manager for each of them with project-specific access rights. Instead of defining a role (or permission) for each of the projects, you define one role, let's say *Projectleader*, and associate a role parameter *project* with it. When a user is assigned to this role, a value for the *project* parameter must be provided. The privilege resolution process grants only those groups that are needed for managing this project.

Another alternative is to use permission match rules. Since permission match rules apply to user attributes, our scenario also needs in this case a special attribute such as *projectleader*. This solution tends to require a growing number of user attributes as the number of permission match rule scenarios increases.

Each role parameter requires a matching rule that acts as a filter for calculating user-group relationships during the privilege resolution process. When a user is assigned to such a role, for each role parameter at least one value must be provided. The subsequent privilege resolution process grants the user only those groups whose attributes match with the provided role parameter values. The matching is defined by the matching rule: it compares the role parameter values with the selected group attributes.



Attributes to be used as group attributes must be defined as permissions parameters (see the Domain Configuration object).

Before you can associate a role with a role parameter, you must specify role parameter names - and optionally also the values - that are allowed for the domain. Navigate to the domain configuration folder and select the *RoleParams* sub-folder in Customer Extensions to add new role parameter definitions.

DirX Identity Provisioning applies the matching rule to all groups that this role grants, even those that are inherited by junior roles. Suppose a role *Role1* has a junior role *Junior1* with a permission *PermissionForJunior1* and an attached group *GroupJunior1*. This group is only granted if its attributes comply with the role parameter values given for the *Role1* assignment. If there are permission match rules defined for the permission *PermissionForJunior1*, they act as an additional filter; that is, they are "and-ed" with the matching rules of the role parameters. However, if the same group is granted by another role without any role parameters, it is granted independently of the *Role1* matching rules.

If more than one role parameter has been defined for a role, groups are only granted if they comply with all match rules. If more than one value has been provided for one role parameter, a group is granted if it complies with at least one of the values.

A matching rule can have the following formats, where "UTRA" stands for "User-to-role assignment":

- *UTRA.attribute operator Group.attribute* | "*string*"
- *UTRA.attribute* **startsWith** "*string*"
- *UTRA.attribute* **contains** *Group.attribute* | "*string*"
- *UTRA.attribute* **endsWith** "*string*"
- *UTRA.attribute* **isContained** *Group.attribute*

These *operators* are available: **=**, **>**, **<**, **>=**, ⇐.

Explicit String operators are: **eq**, **gt**, **lt**, **ge**, **le**.

Here are some examples:

- *UTRA*.l = Group.l
- *UTRA*.ou contains Group.ou
- *UTRA*.project isContained Group.project
- *UTRA*.l = "Mch-P"
- *UTRA*.l startsWith "Mch"



Using the asterisk (*****) as a wildcard in a group attribute used for a match rule always results in a match.

To assign or change a matching rule with DirX Identity Manager (Provisioning):

1. Click the role in the role subtree to display its properties.
2. Click the Role Parameters tab, and then click **Edit**. Now you can edit the values of the matching rule rows.

9.2.3. Creating a Role Hierarchy

DirX Identity Provisioning supports a role hierarchy consisting of *senior* and *junior* roles. A senior role contains one or more other roles, while a junior role is contained within another role.

For example, if the role **“R&D department manager”** contains the role **“R&D software engineer”**, then **“R&D department manager”** is the *senior* role and **“R&D software engineer”** is the *junior* role.

In general, a role becomes a *senior* role when another role is assigned to it, and a role becomes a *junior* role when it is assigned to another role. Assigning **“R&D software engineer”** to **“R&D department manager”** therefore establishes this *senior–junior* relationship.

The easiest way to create a role hierarchy is to build it from the bottom up: first, create an initial set of roles that have no junior roles assigned to them. Next, create another set of roles, and assign the first set you created as junior roles of the new set.

To assign a junior role to a role or delete a junior role assignment:

1. Select the role to display its properties.
2. Click the Junior Roles tab to display the roles that have been assigned as junior roles to the role.
3. Click **Edit**, and then use the role assignment function to add or remove assigned roles.



You cannot assign a senior role to a role (a senior role is at the top of the hierarchy and is assigned one or more junior roles).



The settings at the assigned role object control whether the assignment is immediately valid or valid only after approval.

9.2.4. Assigning Permissions to Roles

To manage role-to-permission assignment with DirX Identity Manager (Provisioning views group):

1. Select a role, either by browsing the role subtree or using the Manager’s search dialog.
2. Click the Permissions tab. Manager displays the role’s assigned permissions in one pane and the available roles in another pane.
3. Click **Edit**.

Now you can use the Manager’s permission assignment function to:

- Select a permission and add it to the role's assigned permissions. You should use the search dialog to reduce the number of assignable permissions that are displayed.
- Select an assigned permission and remove it from the list of assigned permissions.



The settings at the assigned permission object control whether the assignment is immediately valid or valid only after approval.

9.2.5. Saving Changes to a Role

When you click **Save** or change to another tab, the DirX Identity Provisioning system validates the changes you make to the role, and returns the following information depending on the type of changes you make:

- The role name is missing
- The role has no permission assigned to it and/or no junior role assigned to it. The role can be created and a message is written to the ERROR section of the general role property page indicating that no junior roles or permissions have been assigned.
- The role is directly or indirectly assigned to itself (a role circle). Manager removes the new role assignments.
- The user-to-group resolution process results in an error for one or more users. Manager returns the number and type of errors for these users. At this point, you can cancel the operation or proceed. If you proceed, DirX Identity Provisioning marks these users as incurring an error in privilege resolution, causing them to be identified as "inconsistent users". Here are some reasons why you might want to keep users in an inconsistent state:
- The user can continue to work as he has in the past. The previously assigned permissions are assumed to be correct and remain valid, while the new, incorrect permissions are not assigned.
- You (the role administrator) do not have the rights to change the user role assignment (you are not the user administrator and thus do not have the required access rights).

If you have clicked **Save**, Manager saves the changes in the Identity Store (Provisioning Configuration) (otherwise, it saves the changes in its internal cache until you click **Save**). In the event that it is unable to save the data because of network failure or an LDAP server error or timeout, Manager notifies you of the error.



You can click **Offline** during Manager's privilege resolution process to stop interactive privilege resolution and defer it for off-line execution. See the topic "Managing the Privilege Resolution Process" for further details.

9.2.6. Deleting a Role

To delete a role, click it, and then select **Delete** in the menu bar or context menu. DirX Identity Manager (Provisioning) checks that:

- All the role's senior roles have at least one assigned permission and junior role

- All the role's assigned permissions are assigned to at least one other role

Manager then reports all senior roles that do not have an assigned permission or an assigned junior role, and/or all permissions that do not have a role assignment, and asks you to confirm the delete. At this point, you can:

- Confirm the delete
- Cancel the operation

Once you confirm the delete operation, the DirX Identity Provisioning system:

- Resolves the user-group relationships for all users assigned to the role or to its senior role and identifies those users (if any) that will no longer have any group assignment after the role is deleted. At this point, you can choose to cancel the operation or proceed with the delete.
- Deletes all assignments of senior roles or users to the role
- Deletes the role in the DirX Identity store (Provisioning Configuration)
- Stores the changes in users, groups, and accounts in the Identity Store (Provisioning Configuration)



You can click **Offline** during Manager's role resolution process to stop interactive privilege resolution and defer it for off-line execution. See the topic "Managing the privilege resolution Process".

9.2.7. Organizing the Role Subtree

A role folder is a DirX Identity Provisioning object that the role administrator can use to organize the role subtree (the tree view of the defined roles that is displayed when "Roles" is selected in the Manager). Placing a role in a role folder does not affect the role's place in the defined role hierarchy (the role's junior and senior role assignments); the purpose of the role folder is to make it easy for the administrator to manage the viewable role subtree.

You can make the following changes to the hierarchical tree view of the DirX Identity Provisioning role structure:

- Add a new role folder
- Copy a role folder
- Move a role folder
- Delete a role folder
- Rename a role folder or change its description

To add a new folder, click the root role folder or one of the folders below it, and then select **New** → **Folder** from the context menu. Manager prompts you to enter the unique name for the folder.

To copy an existing folder, click it, then select **Copy Object** from the context menu or menu bar.

To move an existing folder, click it, then select **Move Object** from the context menu or menu bar.

To change an existing folder, click it, and then click **Edit**.

To delete an existing folder, click the folder, and then select **Delete** from the context menu or menu bar.The folder to be deleted must not contain any role objects.

To rename a role or a role folder, click it, then select **Rename** from the context menu or menu bar.



You cannot put permissions or groups into role folders.

9.3. Managing Permissions

Permission management consists of the following tasks:

- Adding permissions
- Assigning a matching rule to a permission
- Managing a permission's group assignments
- Saving changes to a permission
- Deleting permissions
- Organizing the permissions subtree

9.3.1. Adding a Permission

To add a new permission:

1. Click a permission folder in the subtree or click the top-level Permission folder.
2. Select **New** → **Permission** in the context menu. Supply a name (common name (cn)) for the permission and any other general permission attributes; for example, a description.

Manager returns an error if the name you supply for the permission is not unique.

Once you have added a permission to the privilege structure subtree, you can copy it, move it, or rename it. To copy a permission, click it, then select **Copy Object** from the context menu or menu bar. To move a permission, click it, then select **Move Object** from the context menu or menu bar. To rename a permission, click it, then select **Rename** from the context menu or menu bar.

9.3.2. Assigning a Matching Rule

Each permission can be assigned a matching rule that acts as a filter for calculating user-group relationships during the privilege resolution process. The matching rule operates as a comparison on user and group attributes that have been defined as permission parameters. The matching rule references these user and group attributes.



Attributes that shall be used as group attributes must be defined as permissions parameters (see the Domain Configuration object).

When DirX Identity Provisioning is resolving user rights for a given user-role assignment that aggregates the permission or a direct user-permission assignment, it performs the matching rule on every group assigned to the permission and assigns to the user only those groups that match the rule. If there is no matching rule defined for a permission, the privilege resolution process takes all of the groups that are assigned to the permission and assigns them to the user who is assigned to the role.

A matching rule can have the following formats:

- *User.attribute operator Group.attribute* | "*string*"
- *User.attribute* **startsWith** "*string*"
- *User.attribute* **contains** *Group.attribute* | "*string*"
- *User.attribute* **endsWith** "*string*"
- *User.attribute* **isContained** *Group.attribute*

These *operators* are available: '=', '>', '<', '>=', '⇐'.

Here are some examples:

- User.l = Group.l
- User.ou contains Group.ou
- User.project isContained Group.project
- User.l = "Mch-P"
- User.l startsWith "Mch"

You can specify a list of matching rules and concatenate them with the **AND** logical operator or the **OR** logical operator. However, you cannot use the **AND** and **OR** logical operators in the same matching rule specification. For example, the following specification is allowed:

User.l = Group.l OR User.l startsWith "Mch" OR User.l isContained Group.l

While the following specification is not allowed:

User.l = Group.l AND User.l startsWith "Mch" OR User.l isContained Group.l



Using the asterisk (*****) as a wildcard in a group attribute used for a match rule always results in a match.

To assign or change a matching rule with DirX Identity Manager (Provisioning):

1. Click the permission in the permission subtree to display its properties.
2. Click the Matching Rule tab, then click **Edit**. Now you can edit the value of the matching rule field.

9.3.3. Managing a Permission's Group Assignments

To manage permission-to-group assignment with DirX Identity Manager (Provisioning):

1. Select a permission, either by browsing the permission subtree or using the Manager's search dialog.
2. Click the Assigned Groups tab. Manager displays the permission's assigned groups in one pane and the available groups in another pane.
3. Click **Edit**.

Now you can use the Manager's group assignment function to:

- Select a group and add it to the permission's assigned groups. You can use the search dialog to reduce the number of assignable groups that are displayed.
- Select an assigned group and remove it from the list of assigned groups.



The settings at the assigned group object control whether the assignment is immediately valid or valid only after approval.

9.3.4. Saving Changes to a Permission

When you click **Save** or change to another tab, the DirX Identity Provisioning system validates the changes you make to the permission. Manager returns any errors or warnings that result from the resolution process (these messages are described in the topic "Saving Changes to a Role"). If you have clicked **Save**, DirX Identity Provisioning then stores in the Identity Store (Provisioning Configuration) the changes to the permission and also stores any changes to user-group relationships. If you have just changed tabs, DirX Identity Provisioning stores the changes in its internal cache.



You can click **Offline** during Manager's privilege resolution process to stop interactive privilege resolution and defer it for off-line execution. See the topic "Managing the Privilege Resolution Process" for further details.

9.3.5. Deleting a Permission

To delete a permission, click it, and then select **Delete** in the menu bar or context menu. DirX Identity Manager (Provisioning) checks that:

- The permission you are deleting is not the only assigned permission of a role
- The deletion of the permission does not result in users with no group assignments

Manager then requests that you confirm the delete, after which it:

- Resolves all users assigned to this permission and stores in the Identity Store (Provisioning Configuration) the user-group relations of all users that are assigned to a parental role of the permission
- Deletes all assignments of roles to the permission from the Identity Store (Provisioning Configuration)

- Deletes the permission from the DirX Identity store (Provisioning Configuration)



You can click **Offline** during Manager's role resolution process to stop interactive privilege resolution and defer it for off-line execution. See the topic "Managing the privilege resolution Process" for further details.

9.3.6. Organizing the Permission Subtree

You can use permission folders to organize the permission subtree (the tree view of the defined permissions that is displayed when the "Privileges" view is selected in the Manager). You can make the following changes to the hierarchical tree view of the permissions subtree:

- Add a new permission folder
- Copy a permission folder
- Move a permission folder
- Delete a permission folder
- Rename a permission folder or change its description

To add a new folder, click the root permission folder or one of the folders below it, and then select **New** → **Folder** from the context menu. Manager prompts you to enter the unique name for the folder.

To copy an existing folder, click it, and then select **Copy Object** from the context menu or menu bar.

To move an existing folder, click it, and then select **Move Object** from the context menu or menu bar.

To change an existing folder, click it, and then click **Edit**.

To delete an existing folder, click the folder, and then select **Delete** from the context menu or menu bar.The folder to be deleted must not contain any permission objects.

To rename a permission folder, click it, and then select **Rename** from the context menu or menu bar.

9.4. Managing Groups

Managing groups in the privilege structure consists of these tasks:

- Adding a group
- Assigning permission parameters and their values for the group
- Viewing the permissions to which the group is assigned
- Saving changes to a group
- Deleting a group

9.4.1. Adding a Group

To add a new group:

1. Click a target system folder in the group subtree.
2. Select **New → Group** in the context menu. Supply a name (common name (cn)) for the group and any other general group attributes; for example, a description.

Manager returns an error if the name you supply for the group is not unique.

For hierarchical target systems you can structure your groups with containers:

- Select **New → Container** from the context menu to create a new container.

You can use this feature, for example, to exclude a set of groups (virtual groups) from the target system synchronization or define several subtrees that are synchronized separately into special areas in your target system.

9.4.2. Assigning Permission Parameters

To assign permission parameters and their values to a group:

1. Select the group to display its properties.
2. Click the General tab to display the group's attributes. The permission parameters and values are displayed in this tab.
3. Click **Edit**. Now you can modify the permission parameter attributes.

9.4.3. Viewing a Group's Permission Assignments

To view the permissions to which a group is assigned:

1. Select the group to display its properties.
2. Click the Permissions tab. Manager displays the permissions to which the group has been assigned.

9.4.4. Saving Changes to a Group

When you click **Save** or change to another tab, the DirX Identity Provisioning system validates the changes you make to the group. If you have clicked **Save**, DirX Identity Provisioning stores in the Provisioning Configuration the changes to the group and also stores:

- Any changes to user-group relationships (caused by changes to permission parameter values)
- Any changes to permission parameter values and other attributes

If you have not clicked **Save**, but have only changed tabs, DirX Identity Provisioning stores the changes in its internal cache.



You can click **Offline** during Manager's privilege resolution process to stop interactive privilege resolution and defer it for off-line execution. See the topic "Managing the Privilege Resolution Process" for further details.

9.4.5. Deleting a Group

To delete a group, click it, and then select Delete from the menu bar or context menu. DirX Identity Manager (Provisioning) checks that you are authorized to delete the group, and then requests that you confirm the delete. The DirX Identity Provisioning system then validates that:

- The deletion of the group does not result in users with no group assignments

Manager then requests that you confirm the delete, after which it:

- Resolves and stores in the DirX Identity store (Provisioning Configuration) the user-group relations of all users that are assigned to a parental role of a permission with an assignment to the group to be deleted
- Deletes all the assignments of permissions to the group from the Identity Store (Provisioning Configuration)
- Marks all direct assignments of users to the group as "to be deleted" (DELETED)
- Sets the group's state to "to be deleted" (DELETED) and sets a timeout value for the deletion of the group in the Identity Store (Provisioning Configuration)

The target system integrator performs the actual deletion of the group in the target system when it synchronizes the information in the Identity Store (Provisioning Configuration) with the target system.At this time, the group in the Identity Store (Provisioning Configuration) is deleted.If the target system integrator is unable to delete the group in the target system, the group remains in the Identity Store (Provisioning Configuration) until its deletion timeout value is reached, at which point it is deleted from the Identity Store (Provisioning Configuration) regardless of whether or not it has been deleted from the target system.



You can click **Offline** during Manager's privilege resolution process to stop interactive privilege resolution and defer it for off-line execution.See the topic "Managing the Privilege Resolution Process" for further details.

9.5. Working with Privilege Structure Query Folders

Query folders provide a tool that you can use to search for and identify privilege structure objects with similar properties or to search for and identify privilege structure objects that need administrator action; for example, roles that do not have any permissions assigned to them.When you open a query folder, it performs a search on the DirX Identity store (Provisioning Configuration) according to the values in its Filter tab and displays the results as objects underneath the query folder.In the DirX Identity Provisioning privilege structure view, you can:

- Add a new role, permission, or group query folder

- Copy a role permission or groups query folder (and then change it to your requirements)
- Delete a role, permission, or group query folder
- Rename a role, permission, or group query folder

DirX Identity Provisioning provides the following default query folders in the DirX Identity Provisioning role view, one for each privilege structure object type:

- Error - identifies all roles, permissions, or groups that have messages in their **Error** field
- To Do - identifies all roles, permissions, or groups that have messages in their **To do** field

The topic "Using Query Folders to Detect Inconsistencies" provides more information about detecting inconsistencies with query folders.

9.6. Managing the Privilege Resolution Process

When you make an access-relevant change to a role, permission, or group or delete a role, a permission, or a group, DirX Identity Manager searches the Identity Store (Provisioning Configuration) for all of the users affected by the update and resolves their user-group relationships.This process can be very time-consuming, depending on the levels of the privilege hierarchy through which DirX Identity Manager must search and the number of users that are directly or indirectly assigned to the updated privilege structure object.

Manager performs privilege resolution in three phases:

1. It searches for the roles or permissions affected by the change.
2. It searches for the users affected by the change.
3. It resolves the user-group relationships for the affected users.

For objects to be deleted, Manager removes all references to them by other objects - for example, references from all users assigned a role or permission that is to be deleted - and marks the referencing objects (in this case, the users assigned the role or permission) as "to be analyzed". The user object will not be resolved by the Manager but the resolution will be triggered by a change event or resolve message sent by the Manager.

Manager displays a progress bar as it executes these phases. You can click **Cancel** at any one of these phases to interrupt it. Manager marks the remaining affected DirX Identity Provisioning objects as "to be analyzed", and resolution takes place at the next run of the workflow that maintains the consistency of the Identity Store (Provisioning Configuration).



Marking the objects as "to be analyzed" can take some time dependent on the number of objects.

The next sections describe details of the privilege resolution process, including:

- Role parameter handling
- Permission parameter handling

9.6.1. Handling Role Parameters

Role parameters can be used to reduce the number of roles in the DirX Identity domain. The following figure illustrates this concept:

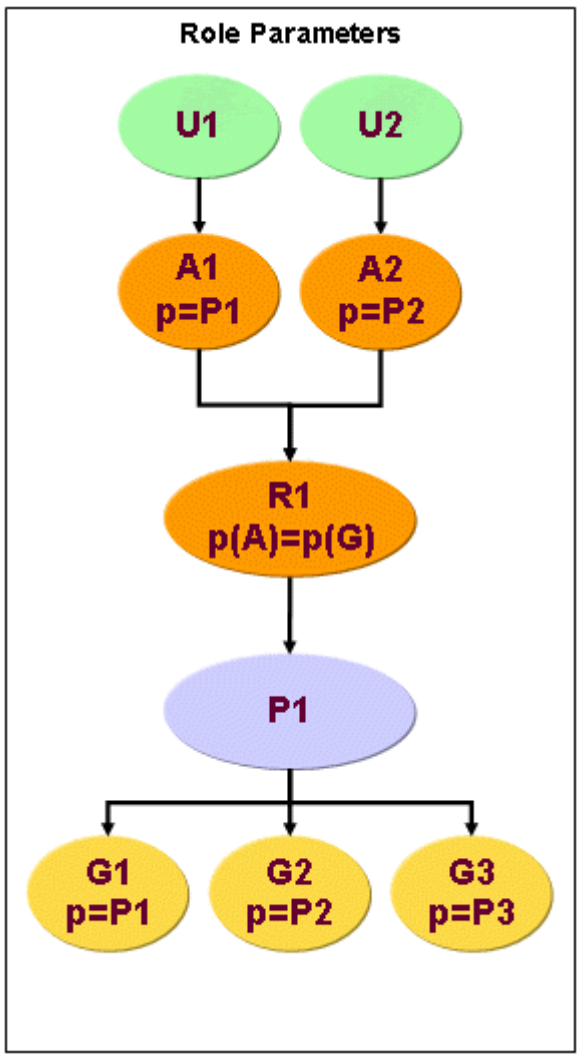


Figure 27. Example Role Parameters

This figure shows two users U1 and U2 that use the same role R1 but with different parameters. Let's assume that the parameter p stands for "Projectname". User U1 uses role R1 with parameter p set to P1 (project 1) and user U2 uses role R1 with parameter p set to P2 (project 2).

Role R1 uses one permission P1 that itself uses three groups, one for project 1 (P1), one for project 2 (P2) and one group for project 3 (P3).

The match rule of the role defines that the user gets a group assigned dependent on the project (p) attribute of the user-to-role assignment (A1 or A2). If the project attribute of the assignment matches the project (p) attribute of the group, the group is assigned to them.

In this case, the user U1 with p set to P1 gets the group G1 assigned. User U2 with p set to P2 gets the group G2 assigned.

You set up role parameters as separate objects in the **Domain** view beneath the **Customer**

Extensions → Role Parameters folder. DirX Identity supports several types of role parameter value definition. See the topic "Setting up Role Parameters" in the chapter "Managing Domains" for more information.

You can also define hierarchical role parameters that are typically modeled as an object tree (for example, as a group tree in a target system). Access to these role parameters can be restricted, which results in a special type of delegation. See the section "Policies for Hierarchical Role Parameters" in the chapter "Managing Access Policies" for more information.

9.6.2. Handling Permission Parameters

Permission parameters can be used to reduce the number of permissions in the DirX Identity domain. The following figure illustrates this concept:

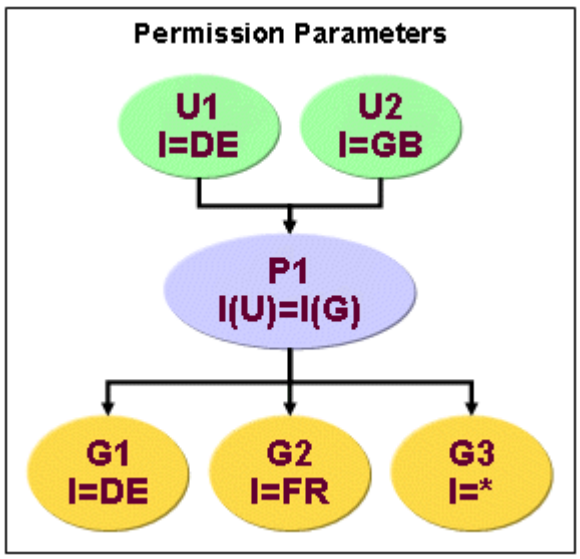


Figure 28. Example Permission Parameters

The figure shows two users U1 and U2 that use the same permission P1. This permission uses three groups, one for German users (G1), one for French (G2) and one group for all users (G3).

The match rule of the permission defines that the user gets a group assigned depending on the locality (l) attribute. If the user's locality attribute matches the locality attribute of the group, the group is assigned to him.

In this case, the user U1 with l set to DE gets the groups G1 and G3 assigned. User U2 with l set to GB gets only the group G3 assigned.

10. Managing Policies

Policy management is the process of setting up the initial policy structure. You can create different types of policies and rules and then maintain that structure. Use DirX Identity Manager (Provisioning) to set up the policy structure by hand.

The topics in this section describe how to use DirX Identity Manager (Provisioning) to create and manage the objects associated with a policy structure.

When you log into DirX Identity Provisioning and select **Policies** from the view bar, DirX Identity displays two trees of objects:

- **Policies -** contains all types of policies for access control and automated provisioning as well as segregation of duties (SoD) policies and password policies.
- **Delegations -** contains all delegations.

To view the properties of a policy or delegation object, click its entry in the tree. When you select a policy or delegation object, DirX Identity Manager (Provisioning) displays a property dialog for the object. The dialog typically consists of a set of tabs that you can use to view the object's properties. Click the tabs in the property dialog to move between the different property categories.

10.1. Delegated Administration

Delegated administration is a two-step process:

- Setting up and maintaining access policies with DirX Identity Manager (Provisioning) that define access rights for access control and retrieval of approvers for approval workflows.
- Creating and maintaining delegations with DirX Identity Web Center that are based on the access policies you defined.



If you intend to use the delegation feature, try to restrict the number of resources per operation type to less than 1000. Otherwise, you will encounter performance problems.

You can modify delegation and access right objects with DirX Identity Manager (Provisioning).

The following figure shows the access policy architecture.

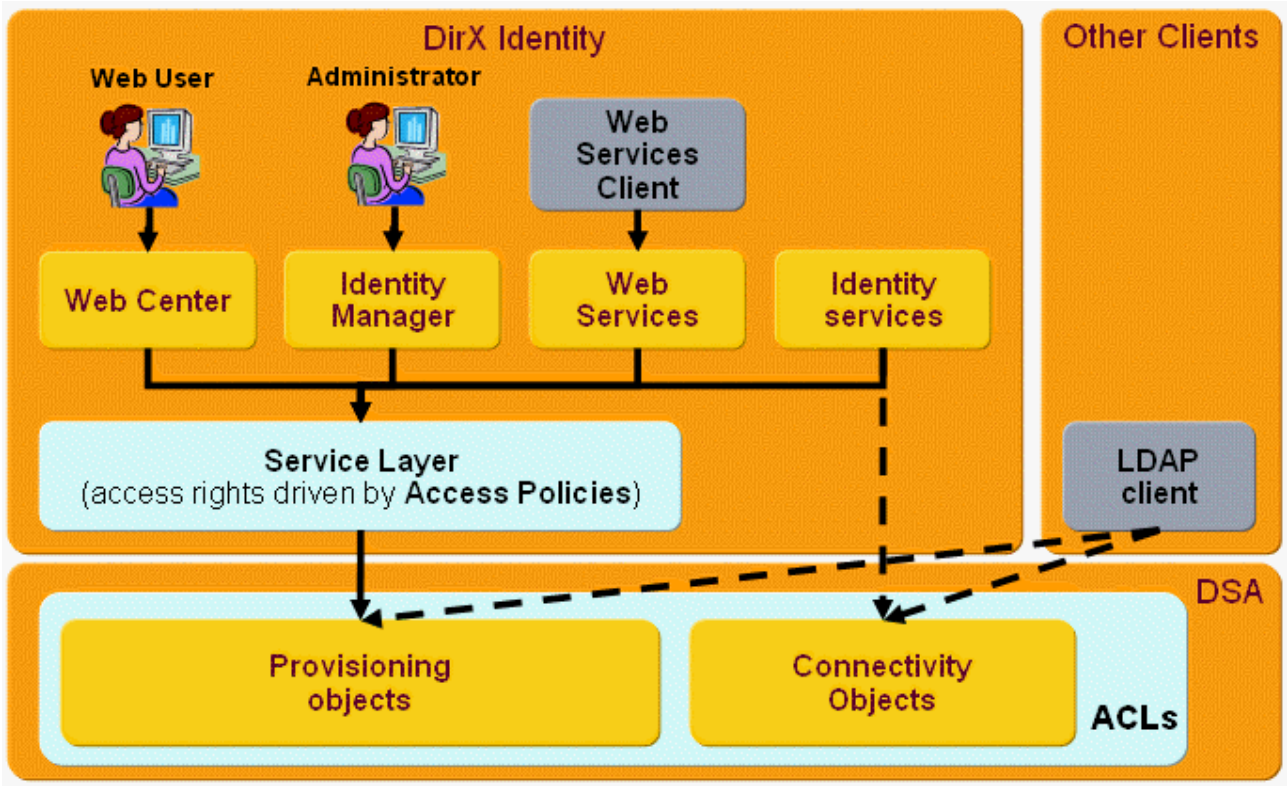


Figure 29. Access Policy Architecture

At the LDAP level (DSA), all objects are protected by directory access control lists (**ACLs**). Because these ACLs are not standardized, there is no common mechanism to set up ACLs in LDAP directories of different vendors. Furthermore, the capabilities of the ACLs vary from vendor to vendor. Most vendors do not support access control based on attribute values; for example, read access to all users that have the employeeType=Internal.

Nevertheless, DirX Identity Provisioning and Connectivity objects are protected by a basic set of ACLs at the LDAP level to control access from pure LDAP clients.

For Connectivity objects, three default ACL groups exist:

cn=Read - allows read access to all connectivity objects. The **admin** and **server_admin** accounts use this group.

cn=Write - allows to write access to all connectivity objects. The **admin** and **DomainAdmin** accounts use this group.

cn=Server - allows to write access to a few specific objects. Only the **server_admin** account uses this group.

You can find these groups under the node dxmC=Groups,dxmC=DirXmetahub. The corresponding accounts are either located under the root node dxmC=DirXmetahub or under the node dxmC=Users,dxmC=DirXmetahub.

Additionally, all DirX Identity components (Web Center, Identity Manager, Web Services, Identity Services) access all Provisioning objects through the **Service Layer**. Access policies protect the access through this layer to all configuration and data objects. This structure guarantees common operation for all components and a single point of control.

Set up audit policies to audit all operations through this layer; see the topic "Managing Auditing → Managing the Audit Trail "for more information.

10.1.1. Managing Access Policies

Access policies control the operations of a subject (normally a user) to a resource (any object) in the Identity Store (Provisioning Configuration). The following figure illustrates the access control model.

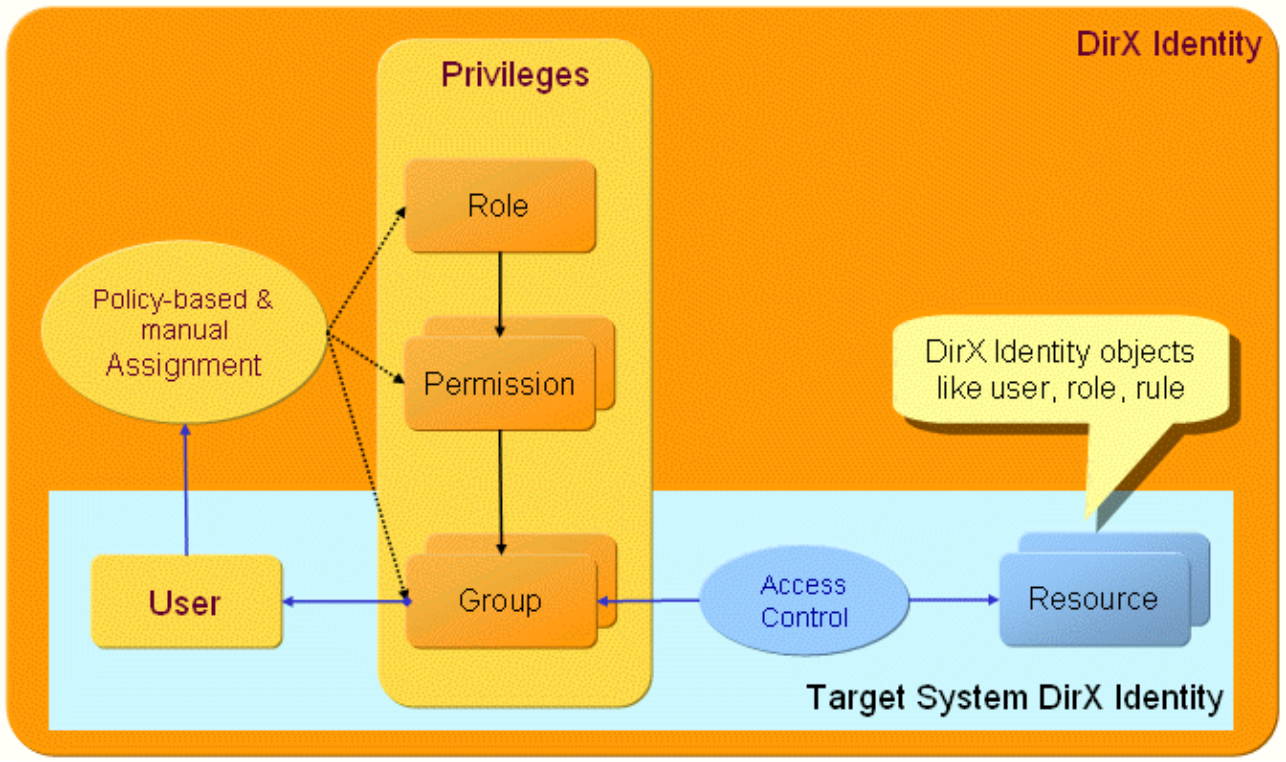


Figure 30. Access Control Model

This concept is very similar to the access control model for external target systems, but here the targets are the internal objects of DirX Identity. Access policies define a group of subjects that have specific access to a group of resources.

Access policies can be used to:

- Define access rights for access control and delegation
- Define access policies to view privilege assignments and accounts
- Define access policies for hierarchical role parameter values
- Retrieve approvers for approval workflows

The next sections describe these tasks in more detail.

DirX Identity comes with a set of default access policies that are necessary for correct operation in a configured customer domain of the initially installed product (cn=Default,cn=AccessPolicies,cn=Policies,cn=*domain*).

Add your own access policies under a custom folder; for example,

cn=MyPolicies,cn=AccessPolicies,cn=Policies,cn=*domain*. Check the inactive policies in this default area to see if you can use them, or view the **My-Company** sample domain for policies that can be reused in your custom environment.

10.1.1.1. Understanding Access Policies

You can define access policies for access control on many object types with a lot of operations. DirX Identity access policy definitions are inspired by XACML, but they are not expressed in XML. Instead, access policies are simple LDAP entries.

To specify an access policy, define the following parameters:

- The **object type** this access policy is to protect; for example, a user or a role.
- For some object types - for example role parameters - you need to specify additional parameters.
- The **operation** this access policy is to handle; for example, read, modify or delete.
- The **subjects** to which this access policy applies.
- Optionally, the resources that the subjects are allowed to work on via this access policy.
- For policies with read or modify operations, the resource attributes to which the policy applies.
- An optional **rule** that defines the relationship between subjects and resources.

These parameters allow you to define almost any kind of access control. Including groups as subjects or resources is particularly effective if you populate these groups via DirX Identity's privilege features: assigning a role to a user which results in the membership to a specific group. If this group is used by an access policy, the access policy is controlled by the role assignment which itself can be a result of a privilege rule.

DirX Identity converts these access policies to LDAP queries and attempts to work with only one combined LDAP filter for optimized performance. If this is not possible, it reads entries from the directory and then filters the resulting list in memory, for example, against another list. In some cases, this procedure may result in poor performance. Try to avoid such filter conditions.

The following figure shows the operations allowed for each object type.

	create	read	modify	delete	viewAssignment	grant	approve	execute	suspend	resume	stop	change participant	show tasks of	delegate	setPassword	readPassword
Accounts	X	X	X	X	X										X	X
Groups	X	X	X	X	X	X	X									
Menus								X								
Password policies	X	X	X	X												
Permissions	X	X	X	X	X	X	X									
Provisioning rules	X	X	X	X												
Reports								X								
Roles	X	X	X	X	X	X	X									
Role Parameters						X										
Users	X	X	X	X			X						X	X	X	
Target Systems	X	X	X	X												
Business Objects	X	X	X	X												
Workflow definitions	X	X	X	X				X								
Workflow instances		X	X	X					X	X	X					
Activity instances												X				
Certification Campaigns	X	X	X	X			X					X	X			

Figure 31. Access Policy Objects and Operations

Read the section "Understanding the Sample Domain → My-Company Sample Domain → Policies → Access Policies" in the *DirX Identity Tutorial*. It explains some of the examples that are delivered as default policies or that are part of the Sample Domain.

10.1.1.2. How to Select Subjects

This section shows different ways to select subjects. You can:

- Select a list of discrete users
- Select a dynamically filtered object list (LDAP filter)
- Select a list of groups (dynamically populated with privileges) in one of the following ways:
- By direct assignment of privileges
- Controlled by user attributes (permission parameters)
- Controlled by assignment parameters (role parameters)
- Where the assignment is controlled by provisioning rules (user parameters)

10.1.1.2.1. Selecting a List of Direct Users

You can set up a discrete list of persons. Enter the list row by row into the **Persons** field. The following figure illustrates this method:

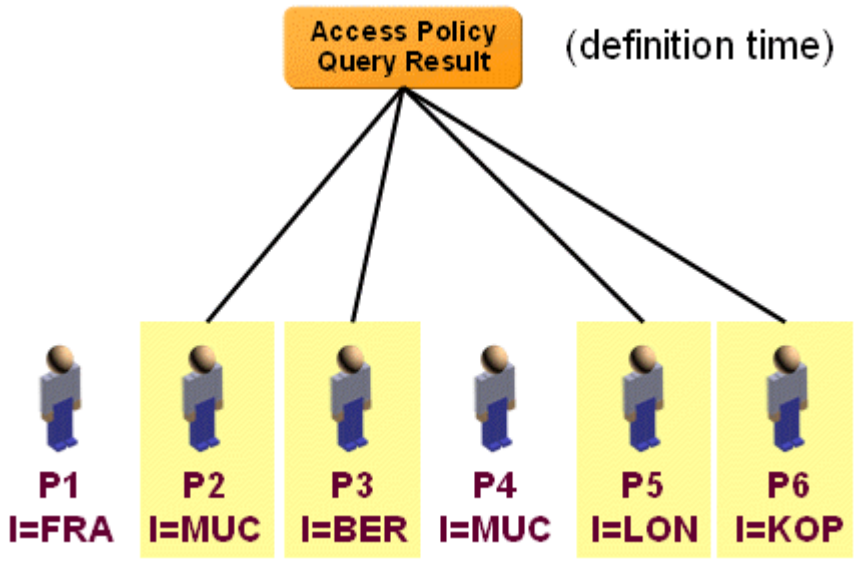


Figure 32. Selecting Subjects Directly

The figure shows that user P2, P3, P5 and P6 are valid subjects of the access policy. This selection is static and does not change.

10.1.1.2.2. Selecting a Dynamically Filtered Object List

You can set up an LDAP filter to select subjects. Use the **Persons Filter** field. The access policy is updated in real-time. The following figure illustrates this method:

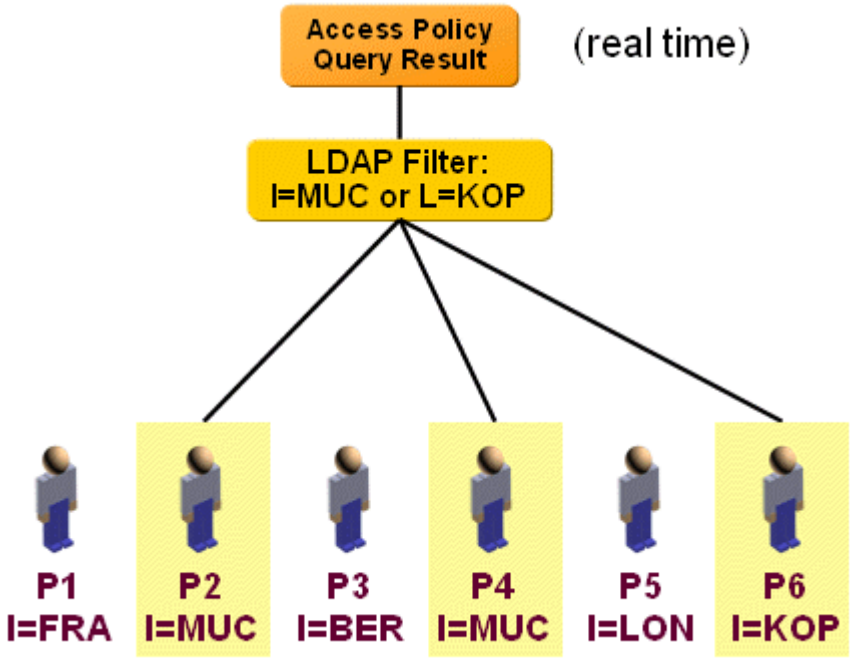


Figure 33. Selecting Subjects with an LDAP Query

In this example, all persons that are located in Munich (MUC) or Copenhagen (KOP) are selected via a dynamic LDAP query as subjects of this access policy. After changing the LDAP query, the result is changed during the next access policy request in real time.

10.1.1.2.3. Selecting a List of Groups by Direct Privilege Assignment

To select a list of groups (dynamically populated with privileges) by direct assignment of privileges, assign roles to users to fill in the members of a group accordingly. Changing the role assignments changes the membership. Add the relevant groups to the **Group of Persons** field. The access policy result is updated after privilege resolution. The following figure illustrates this method:

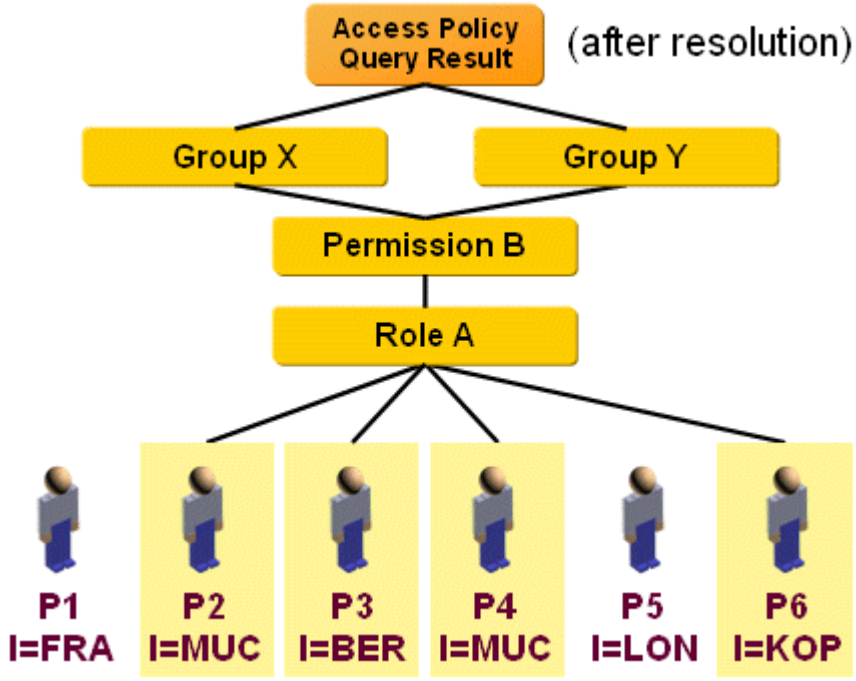


Figure 34. Selecting Subjects with Role Assignments

As shown in the figure, P2, P3, P4 and P6 are selected as subjects because role A is assigned. This action results in group memberships for group X and Y. Both are included in the access policy.

10.1.1.2.4. Selecting a List of Groups Controlled by User Attributes

To select a list of groups (dynamically populated with privileges) controlled by user attributes, use a group that is selected via permission parameters that are compared against user attribute values via permission match rules. Use the **Groups of Persons** field to select the correct group. The access policy result is updated after privilege resolution. The following figure illustrates this method:

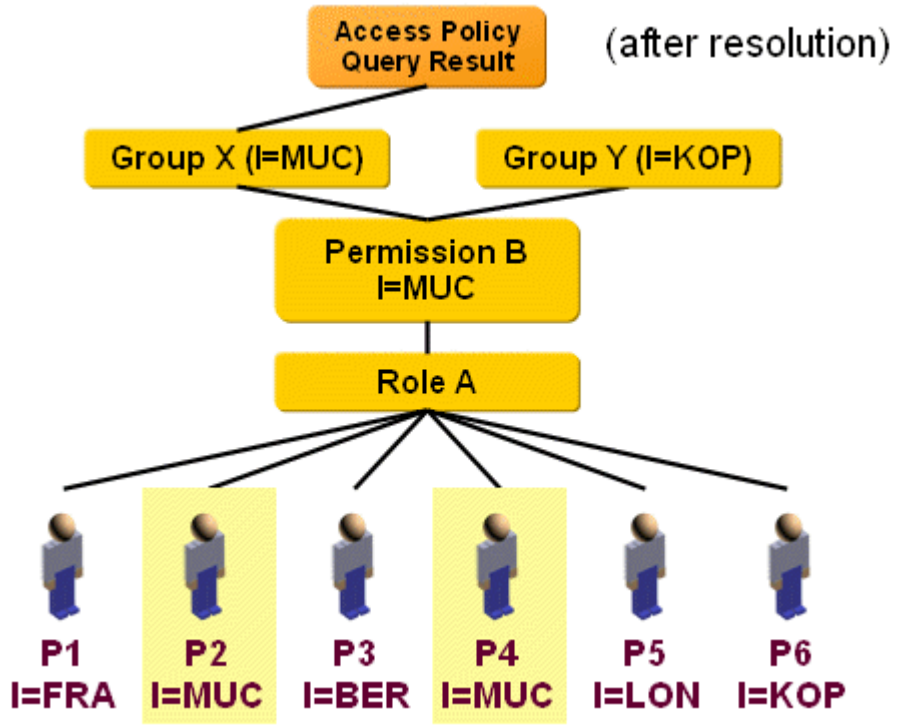


Figure 35. Selecting Subjects with Role Assignments and Permission Parameters

In this example, the users P2 and P4 are selected as subjects because they are located in Munich (MUC). This selection results in group memberships for group X. This group is included in the access policy.

10.1.1.2.5. Selecting a List of Groups Controlled by Assignment Parameters

To select a list of groups (dynamically populated with privileges) controlled by assignment parameters (role parameters), use a group that is selected via role parameters that are assigned during role assignment. These values are evaluated via role parameter match rules. Use the Groups of Persons field to select the correct group(s). The access policy result is updated after privilege resolution. The following figure illustrates this method:

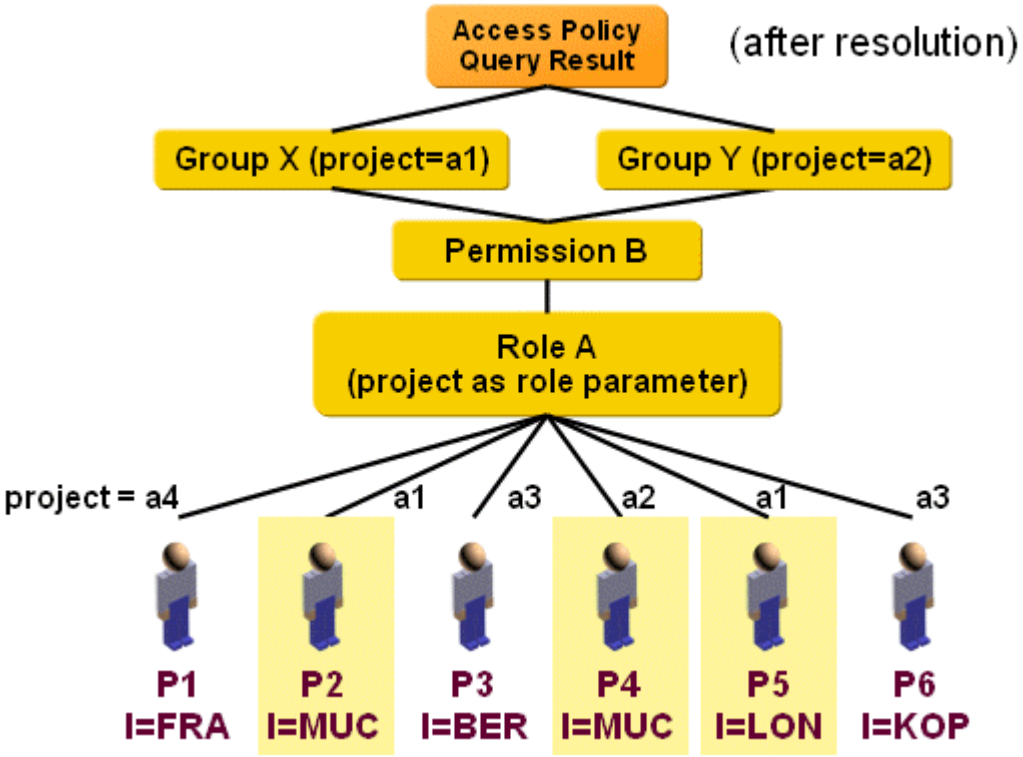


Figure 36. Selecting Subjects with Role Assignments and Role Parameters

In this example, the users P2, P4 and P5 are assigned the role parameter values **a1** or **a2** during assignment of the role. The role parameter match rule adds these persons as members to the corresponding groups. These groups are included in the access policy.

10.1.1.2.6. Selecting a List of Groups Controlled by Provisioning Rules

To select a list of groups (dynamically populated with privileges) whose assignment is controlled by provisioning rules (user parameters), use a group that depends on a role that is assigned automatically via a provisioning rule. The provisioning rule decides on user parameter values. Use the Groups of Persons field to select the correct group(s). The access policy result is updated after privilege resolution. The following figure illustrates this method:

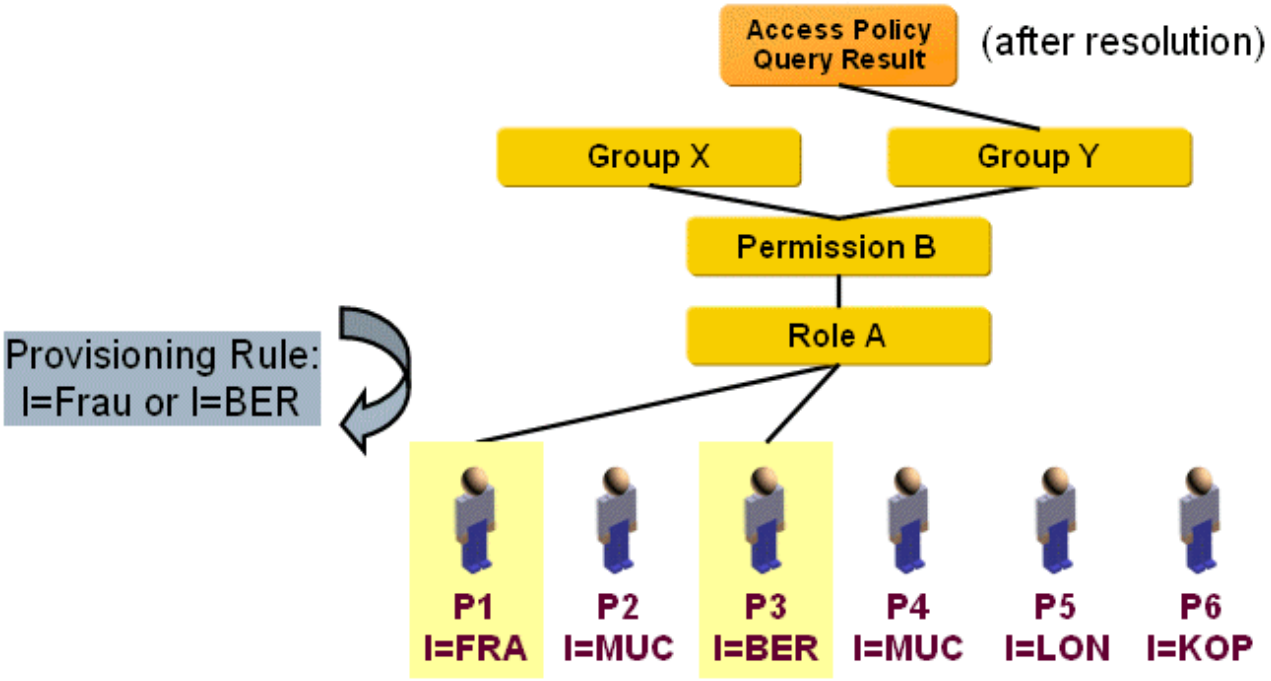


Figure 37. Selecting Subjects with Provisioning Rule and Role Assignment

In this example, the provisioning rule assigns role A to the users P1 and P5 according to the locality parameters of the users. The role uses a permission that itself is tied to group Y. Group Y is included in the access policy.

10.1.1.2.7. Excluding Specific Persons

Use the Exclude persons field to exclude specific persons from the amount of defined subjects. This is only a static list of objects.

10.1.1.3. Policies for Object Creation

You can control object creation in various ways depending on the kind of license you have:

- If you have a **business license**, the ability to create objects depends on the **Creation access policies** (operation = create) that have been established. You define which object types a user is allowed to create and the object types themselves. You can also use creation access policies to define the node under which the user can create objects. If you want to define several locations for object creation, you must define multiple access policies. A business license does not permit the use of request workflows.
- If you have a **professional license**, you can use request workflows to create objects. For more information about this feature, see the section "Creation Workflow Selection" in section "Selecting Request Workflows" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide*.

10.1.1.4. Policies for Object Access

You can control object access by read and modify operations. You can create policies that combine these two operations, or you can create separate policies for read and modify.

Without selecting an attribute, the subjects can perform the read and modify operations on all attributes of a resource. If you select one or more attributes, the operations are restricted to these attributes. For example, if you have a read policy on user resources, and you select the attributes **sn** and **givenname**, the subject users can read only the first and family name of the resource users.

Using attribute policies, you can start request workflows to approve the modification of specific attributes. See the section "Attribute Policies for Users" for more information.

10.1.1.5. Policies for Grant Operations

Grant operations control the assignment of privileges (roles, permissions, groups) to users. Because this is an operation that affects two objects (the user and the privilege to assign), it is important to understand the corresponding access policies:

- A user (the subject) can assign privileges to a set of users (the resources) he is allowed to read; this means that the access policies with operation **read** and object type **User** are relevant.
- A user (the subject) can assign all privileges (the resources) for which he has the **grant** access right to all of the above defined users.

This means that a user requires a read access right on a set of users and a grant access right on a set of privileges to assign privileges to users.

You can use two operation types for granting privileges:

grant - use this operation type for normal operation. For privileges that are marked for approval, it starts the corresponding request workflow.

assignDirect - use this operation type only for a restricted set of administrators. These administrators can assign privileges directly even if they are marked for approval. No approval workflow is started. This operation type works only in connection with a grant policy for the privileges.

10.1.1.6. Policies for Approvals

Access policies can be used to retrieve approver lists for approval workflows. Access policies are typically used twice: once for the subject (the user who gets the privilege) and once for the resource (the privilege that is assigned to the user). In both cases, a list of approver lists can be the result. The following figure shows the approver calculation.

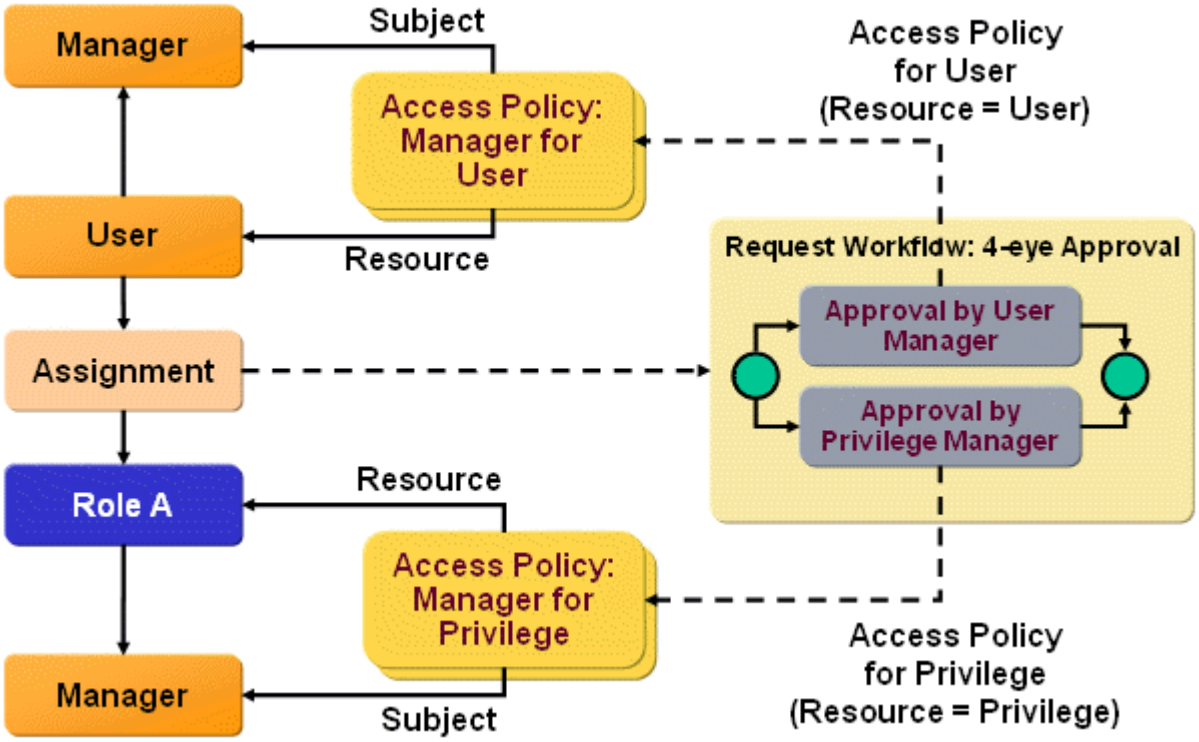


Figure 38. Using Access Policies to Calculate Approvers Dynamically

As shown in the figure, the DirX Identity Provisioning system evaluates a workflow approval request as follows:

1. The item to be processed is either of type "User" or "Privilege". If it is of type "User", it is a user object, otherwise it is a privilege object (role, permission, group). You can see the two types as 'Approval by User Manager' and 'Approval by Privilege Manager' activities in the workflow.
2. The system retrieves all access policies that contain the **Operation** 'approve'.
3. For each access policy, the system determines whether the item is a member of the **Resources** list.
4. If the item is a member, the **Rule** is evaluated against the **Subjects** list.
5. All found members of the subject list that satisfy the rule comprise the result of this access policy.
6. The process is repeated with the next access policy starting at step 3.
7. When all access policies have been evaluated, the system returns a combined list of approvers to the approval workflow request.

For example, suppose the role **Windows Administrator** is assigned to the user **Axel Bartsch**. This action (the flag "Requires Approval" is set for the role) generates a "4-eye Approval" approval workflow. The workflow consists of two parallel steps: the first step retrieves the approvers for the user (User Approval List), and the second step retrieves the approvers for the privilege (Privilege Approval List).

Axel Bartsch is member of the HW development department. His manager is **Veronique Cohu**. This example assumes that the approver of the user is the manager of this user.

Therefore, the **manager** attribute in each user entry points to the entry of the manager.

The owner of the Windows Administrator role is **Nik Taspatch**, which means that Nik is the responsible for that privilege.

The approval workflow first wants to find the approver for the user **Axel Bartsch**. The relevant access policy is "Manager for user". The system performs the following steps for approver retrieval of the User Approval List:

- The item in this case is "Axel Bartsch". It is the resource, and it is of type "User".
- When evaluating all access policies with Operation=approve, the system finds in the default setup only the access policy **Manager for user**.
- The list of resources is defined by a filter with a base object of "cn=Users,cn=My-Company" and a search filter "objectClass=dxrUser" which simply searches all users under this subtree. This means that this access policy is valid for all users in this subtree. Because Axel Bartsch is one of these users, the access policy is used for further evaluation.
- Next, the list of subjects is retrieved. It is defined by a filter with a base object of "cn=Users,cn=My-Company" and a search filter "objectClass=dxrUser" which again retrieves all users of this subtree.
- Now the rule is evaluated against the members of this list. The rule is defined as a permit rule:
  $(subject.dn)="$(resource.manager)"
  This rule finds the entry with a distinguished name equal to the value of the manager attribute of the user. This is the manager of this user.
- So the result of the access policy **Manager for user** is **Veronique Cohu**.



The calculation is not exactly as described above because this would result in huge lists read into memory if the number of users is very large. The software optimizes the calculation to simple LDAP searches that are very fast.

Now the approver for the privilege is evaluated (Privilege Approval List).

- The item in this case is "Windows Administrator". It is the resource and of type "Privilege" (in fact, it is defined more precisely as a role, permission or group).
- When evaluating all access policies with Operation=approve, the system finds after some time the access policy **Manager for resource**.
- The list of resources is defined by a search in the subtree "cn=Corporate Roles,cn=RoleCatalogue,cn=My-Company" with the search filter "objectClass=dxrRole". This means that this access policy is valid for all roles in this subtree. In our example, the privilege "Windows Administrator" is part of that subtree and thus the access policy is used for further evaluation.
- Next, the list of subjects is retrieved. It is defined by a filter with a base object of "cn=Users,cn=My-Company" and a search filter "objectClass=dxrUser" which retrieves all users of this subtree.

- Now the rule is evaluated against the members of this list. The rule is defined as a permit rule:
  $(resource.owner)="$(subject.dn)"
  This rule finds the user entry with a distinguished name equal to the value of the owner attribute of the resource.
- So the result of the access policy **Manager for resource** is **Nik Taspatch**.



The calculation is not exactly as described above because this would result in huge lists read into memory if the user number is very large. The software optimizes the calculation to simple LDAP searches that are very fast.

The approval workflow "4-eye approval" results in parallel approval requests to the manager of the user (Veronique Cohu) as well as to the manager (owner) of the resource (Nik Taspatch). If a privilege has several owners or the user has several managers, all owners and managers are selected as approvers.



For a sample workflow that uses this technology, see the section "4-Eye Approval" in the section "Understanding Assignment Workflows" in chapter "Using Request Workflows" in the *DirX Identity Application Development Guide*.

10.1.1.7. Policies for Viewing Assignments

This feature allows you to restrict the privilege assignments visible at each user entry. You can also control the accounts that are visible.

Because this feature has a significant impact on the Web Center and Identity Manager display, you must enable it explicitly:

- Set the flag **Enable View Policies** in the **Policies** tab of the **Domain** object.

The feature's effect is as follows:

- A user can always see his own privilege assignments and accounts.
- A user will only see direct privilege assignments for other users if he is allowed to grant them to other users.
- Privileges that have been implicitly given are not visible. These privileges are inherited permission assignments for roles and all calculated group assignments for role and permission assignments.
- Accounts are not visible at all.

To view additional privilege assignments and accounts, you must set up access policies explicitly with the viewAssignments operation. You can enable:

- Additional role assignments
- Additional permission assignments

- Additional group assignments
- A set of accounts



This feature can result in the situation where a user cannot see any privilege assignment and cannot see any other user accounts.

10.1.1.8. Policies for Hierarchical Role Parameters

This feature allows you to define access policies to restrict the assignment of hierarchical role parameters.

Hierarchical role parameters are typically modeled as an object tree; for example, as a group tree. The following figure illustrates an example:

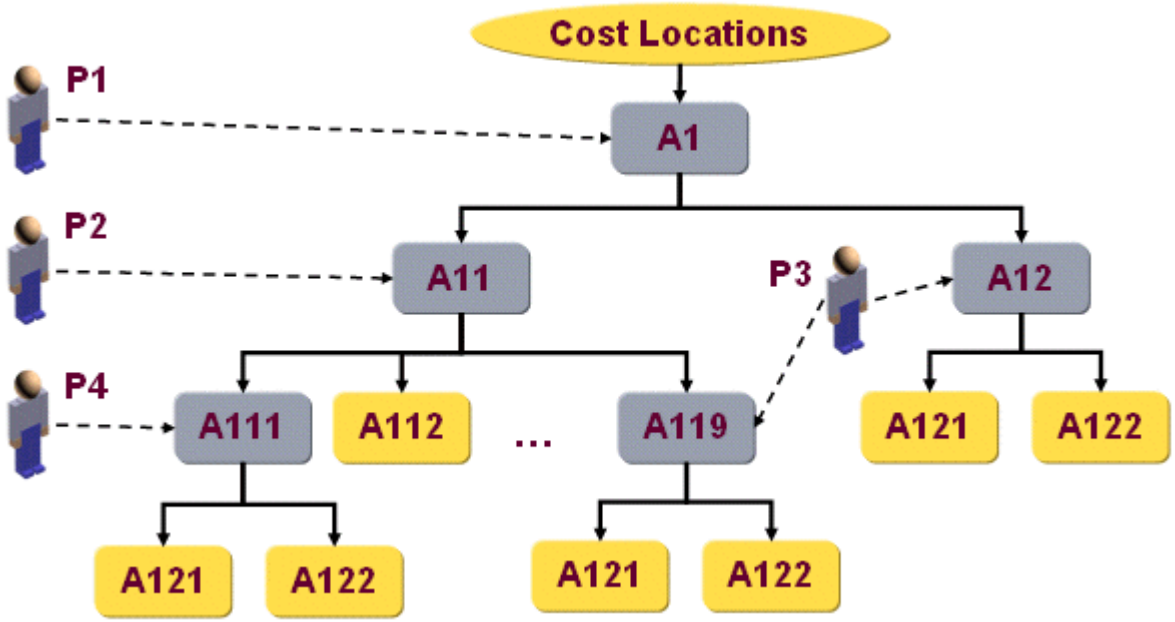


Figure 39. Hierarchical Role Parameters Example

In this example, a cost location tree is modeled as a group tree. The super administrator P1 gets the right to assign cost locations starting from the top level object A1 via an initial access policy. He assigns to another person P2 the right to assign cost locations under the tree A11 and to P3 the right to assign cost locations beneath the trees A12 and A119. P2 decides to assign the right for the A111 tree to P4. This is done via a cost location manager role.

All four persons can use the assigned objects and all objects that belong to these tree nodes as role parameters for role assignment. The following figure and sections explain the details of this mechanism.



This is only an example that explains the possibilities of this feature. It can be used in many other ways.

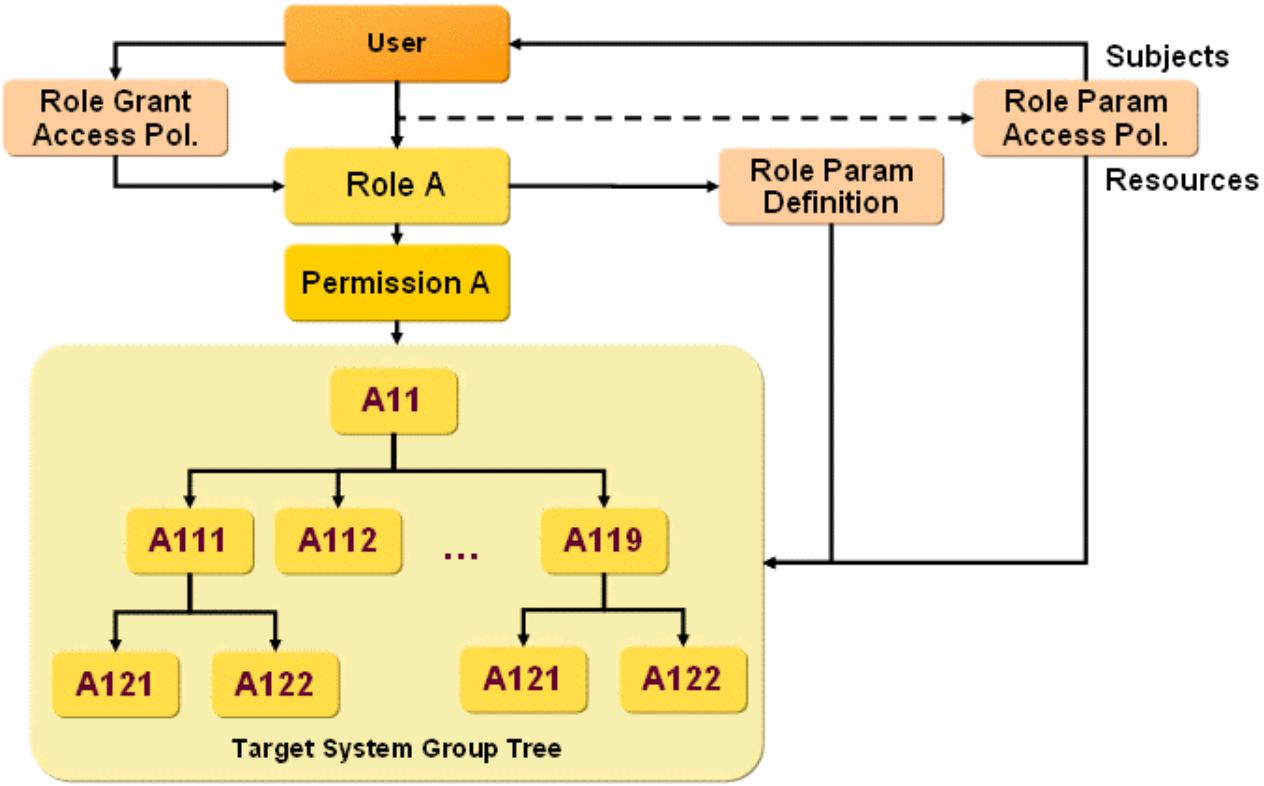


Figure 40. Hierarchical Parameter Object Relationships Example

10.1.1.8.1. Defining the Hierarchical Role Parameter Object

After you set up the cost location group tree (see the previous figure), you must define a role parameter object that allows for handling this group tree as role parameters (the role parameter definition shown in the previous figure):

- Create a new role parameter object in the **Domain** view beneath the **Customer Extensions** → **Role Parameter** folder.
- Set the **Type** to **Hierarchical DN**.
- In the **Root** field, select the cost location group tree top-level node.
- Set the object class to **dxrTargetSystemGroup**. This setting defines which objects are role parameters.

You can find an example of this kind of object in the Sample Domain: Cost Locations.

10.1.1.8.2. Defining Privileges with Hierarchical Parameters

To define privileges with hierarchical parameters (role A, permission A shown in the figure), define a permission that uses the previously set up group tree of cost locations.

- Create a new permission object in the **Privileges** view beneath the **Permissions** folder.
- Link all groups from the cost location tree to this permission.

Define a role that uses the hierarchical parameter definition object.

- Create a new role object in the **Privileges** view beneath the **Roles** folder.

- Set up a rule in the **Role Parameters** tab:
  Cost Locations = group (dn)
- Link the previously created permission to this role.

You can find an example of such an object in the Sample Domain: Cost Location Manager.

10.1.1.8.3. Defining a Grant Policy for the Role (Role A)

You must define a grant access policy that allows the user to assign Role A to users, and you must define role parameter policies.

First, use initial access policies for role parameters (Role Param Access Pol. in the previous figure) to define the starting points for hierarchical role parameter assignment:

- Create a new access policy object in the **Policies** view beneath the **Policies → Access Policies** folder.
- Set the **Operation** to "grant" and the **Object** to "RoleParam".
- Link the previously created role parameter object to the **Role Param** field.
- Define the subjects (for example, all people of the finance department).
- Define as a resource one or more of the nodes in the group tree.
- Set as the rule: **Permit**.

You can find an example of this kind of object in the Sample Domain: "Cost location admins".

All persons that meet the subject definition have now the right to assign the "Cost location manager" role to other persons where the role parameters are restricted by the initial access policy.

Next, define an access policy for the delegation of role parameters (Role Param Access Pol. in the previous figure) that allows persons that are assigned a **Cost location manager** role with a well-defined cost location parameter to assign the **Cost location manager** role with parameters that lie hierarchically beneath the assigned root node to other persons.

- Create a new access policy object in the **Policies** view beneath the **Policies** → **Access Policies** folder.
- Set the **Operation** to **grant** and the **Object** to **RoleParam**.
- Link the previously created role parameter object to the **Role Param** field.
- Define the subjects (for example, all internal employees).
- Define as a resource the whole cost location object tree.
- Set as the rule:
  $(subject.dxrprivilegesgrantedlink)="$(resource.dn)"

You can find an example of this kind of object in the Sample Domain: "Cost location managers".

All persons that meet the subject definition have now the right to assign the "Cost location

manager" role to other persons where the role parameters are restricted by this access policy.

10.1.1.9. Policies for Menus

You can define policies that define the visibility of Web Center menus. To define menu visibility:

- Define the menu keys (the primary menu items) and the menu items (the secondary menu items) as keys and items proposal lists in the folder **Domain Configuration** → **Customer Extensions** → **Proposal Lists** → **Menus**.
- Create collections of menu items that you want to control via access policies. These items can be complete primary menus (coarse granular control) or parts of it (fine granular control).
- Define the necessary access policies to control the defined menu collections.



If you use only a slightly modified version of Web Center, you can use the predefined menu items under the path **Domain Configuration** → **Proposal Lists** → **Menus**.



You cannot change or extend these items because they are overwritten with each product upgrade. However, you can add additional access policies in your environment.

10.1.1.10. Guidelines for Access Policy Setup

Setting up access rights is something you should plan carefully to avoid performance problems. Read the following hints and guidelines thoroughly.

- Avoid defining too many access policies. Too many access control definitions slow down any computer application.
- Check the sample access policies in the sample domain to see if they can be used in your environment.
- Avoid performance problems in advance: Try to create policies with simple and direct filter conditions. If necessary think about additional attributes, that means do not hesitate to extend your directory schema.
- Do not think about indirect filter conditions; for example, "Select all users that have a role assigned that requires approval". Use groups that are populated via role assignments and automatic provisioning rules. You can use any existing group in a target system. If you need a target system-neutral group, create it in the DirXmetaRole target system.
- If you intend to use the delegation feature, try to restrict the number of resources per operation type to less than 1000.
- Do not use wildcards in the filter expression, for example: **$(subject.ou)=$(resource.ou)*** This setup prevents DirX Identity from evaluating the access policy with a single combined LDAP operation. Instead, a huge number of objects may be read and

additional comparison operations may occur in memory to reduce the huge list. If size limits are set at the directory server, the result is not complete and may even be empty in some cases.

- If you use search filters to find groups, do not use the following 'not' clause:
  (&(...)(!(objectClass=dxrObligation))
  This slows down access policy evaluation due to inefficient implementation of not filters in the directory server.
- Do not create complex filter conditions that contain the dn as attribute like
  (($(resource.dxrRoleAdmin)="$(subject.dn)" or $(resource.owner)="$(subject.dn)") and $(resource.owner)=*)

  Define the policy in this way:
  Resource Filter: ((objectClass="dxrRole") and (owner=*))
  Addionally define two rule lines (click the "+" button for a new line):
  Rule 1: $(resource.dxrRoleAdmin)="$(subject.dn)"
  Rule 2: $(resource.owner)="$(subject.dn)"

  The reason why you should not use the complex filter condition with a dn is that you cannot search for a dn in the directory.
- Make sure that any multi-value attribute that is used in the evaluation of access policies has an object description with multivalue="true". Otherwise, the default "single valued string" is assumed in the storage layer, and only the first of the multiple values is evaluated. This leads to unpredictable and maybe continuously changing search results.
- We recommend restricting the number of access policies you make delegable to protect your system performance. Decide which access rights are really critical and should be delegable and keep that number small. Set the **Is Delegatable** flag at each access policy accordingly.

10.1.1.11. Verifying Access Policies

The DirX Identity Services layer evaluates access policies as rules that control access of subjects to resources. The result is a set of access rights. In some cases, it is not easy to understand why a specific person has a specific access right or why he or she does not.

DirX Identity comes with a set of default query folders that help analyzing access policies (cn=_Queries,cn=AccessPolicies,cn=Policies,cn=My-Company). Use these query folders to find all active access policies that handle, for example, read access rights. If the default queries are not sufficient, set up your own custom queries to get more specific results. Now you can check whether the access policies you have found are set up correctly.

10.1.2. Managing Delegations

The **Delegations** folder contains complete delegations (a complete delegation is a definition of which access rights are delegated from a user to another user) as well as the corresponding access rights in the folder **Access Rights** and its sub folders for all access rights (for example, create, modify, delete).

10.1.2.1. Understanding Delegations

Each of the DirX Identity users (the delegator) may delegate part of his administrative rights that are delegatable to other persons (the substitutes). You can use the DirX Identity Manager (Provisioning) Policy view to view the delegation objects and the associated access rights you must use DirX Identity Web Center to perform the actual delegation. The following figure illustrates DirX Identity Provisioning delegation.

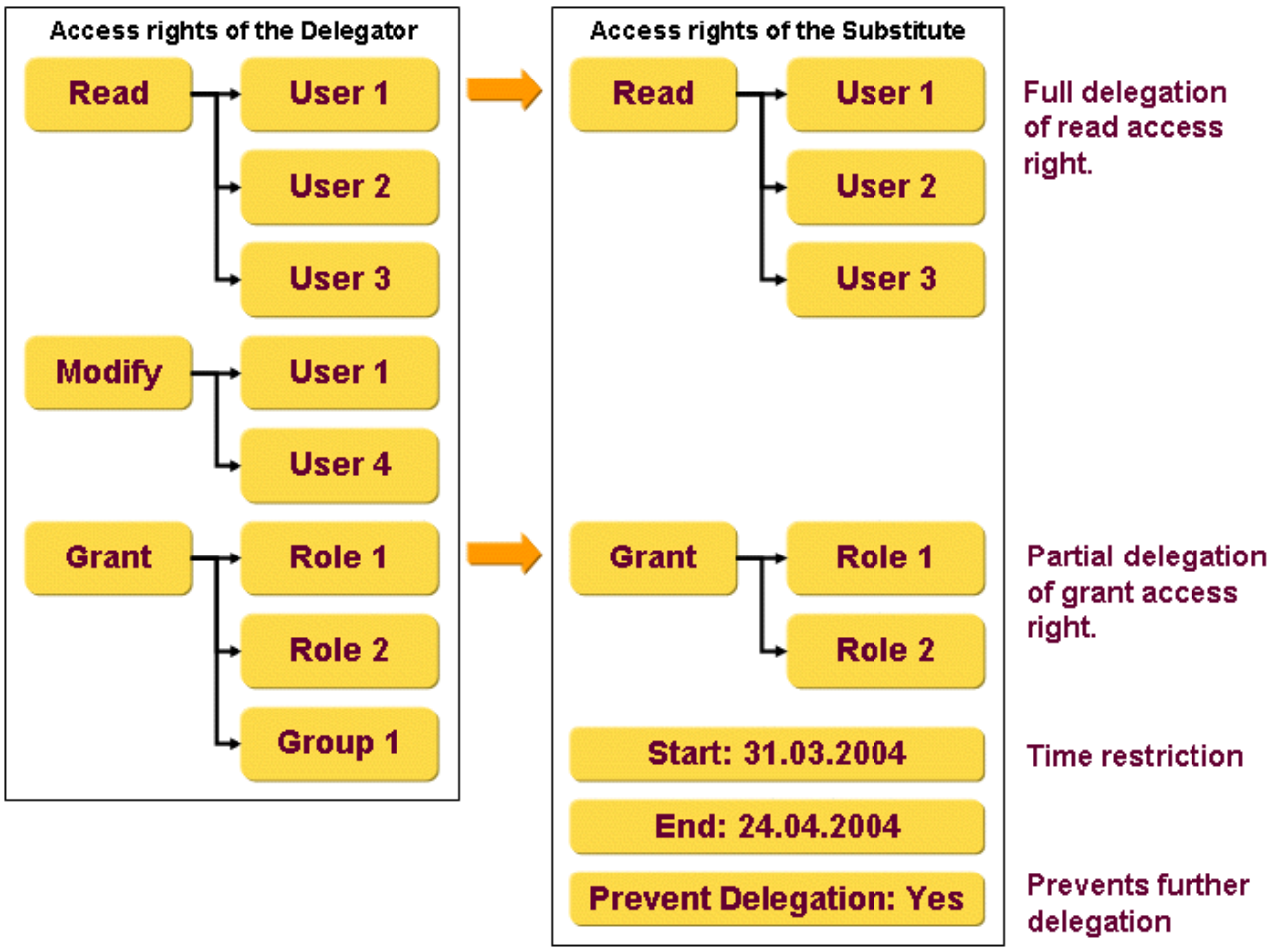


Figure 41. DirX Identity Provisioning Delegation Example

If a user (delegator) wants to delegate access rights to another user (substitute), the delegator performs these steps:

- Selects the user to be the substitute
- Selects a set of access rights from his available access rights based on specific operations (read, modify, grant, approve)
- Restricts the available resources (selects a subset) (optional)
- Defines a time period during which the delegation is valid (optional)
- Sets the flag **Prevent Delegation** to prevent any further delegation of this set of access rights (optional)

Special notes:

- You can only delegate access rights whose access policies have the **Is delegatable** flag

set. If the flag is not set, access rights from these policies have their effect but cannot be delegated.

- The **delegation of the access right approve** is treated in a special way. In this case, the substitute is retrieved as an approver instead of a delegator. Thus, the delegation is exclusive. Otherwise, several persons would be selected to approve (the delegator(s) and the substitute).
- **Delegations are currently static**: changing the access rights of a delegator does not affect existing delegations.
- **Delegation of approve access rights** results in propagation of this right from the delegator to the substitute. This means that the delegator no longer has this access right.
- Delegation performance is not sufficient when working with access policies that retrieve a huge amount of target objects. Use the flag **Is delegatable** for each access policy to specify whether it is delegatable. Set the flag only for access policies that retrieve less than 1,000 target objects.
- If person A delegates approval access rights to person B, and person B delegates this access right back to person A, a **delegation loop** exists. In this case, DirX Identity Provisioning does not retrieve an approver and the approval workflow will fail if the constraints define an approver. This process applies to loops of any depth. The criteria are that there must be a person in the delegation sequence that did not delegate the approval access right to another one. DirX Identity Provisioning takes this person as the approver.

10.1.2.2. Understanding Access Control

Delegated administration presumes a concept of access control. Access to resources can be restricted via access policies. To be able to use all the sophisticated DirX Identity Provisioning mechanisms, Provisioning treats itself as a target system, as illustrated in the following figure.

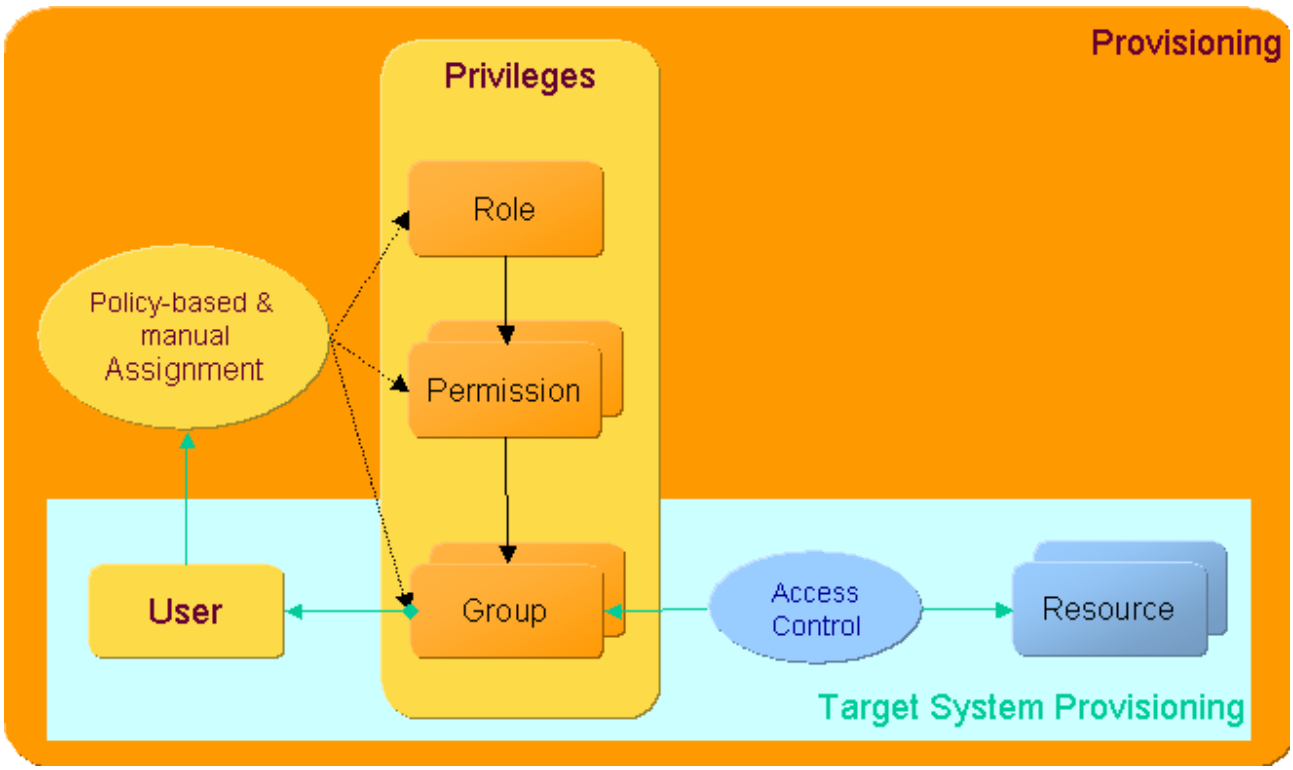


Figure 42. DirX Identity Provisioning Access Control

The access of users to resources (either other users or privileges) is defined by the assignment of privileges (roles, permissions, groups) to users. Privilege assignment can be carried out by hand or as an automated process based on rules.

Access policies control the access to the resources. The DirX Identity Provisioning security manager uses access policies to control access to all the resources in the Identity Store (Provisioning Configuration). Because the target system DirX Identity Provisioning is an integrated part of DirX Identity, no extra provisioning is necessary.



The access control layer is a virtual layer above the directory. It is not built as directory access control information. Thus, the security manager is only effective for all accesses coming from the DirX Identity Manager (Provisioning) or the DirX Identity Web Center. In both cases, a technical user (the DomainAdmin) with high-level access rights is used to access the directory. Access policies restrict this access via the security manager at the virtual level. Pure LDAP client access is not controlled via the security manager.

To simplify administration, DirX Identity Provisioning allows you to formulate access policies that grant access rights (editing users, granting roles, permissions or groups, approving privileges) comfortably for sets of users and privileges. The following figure illustrates the access policy structure and its relationship to users, groups and resources.

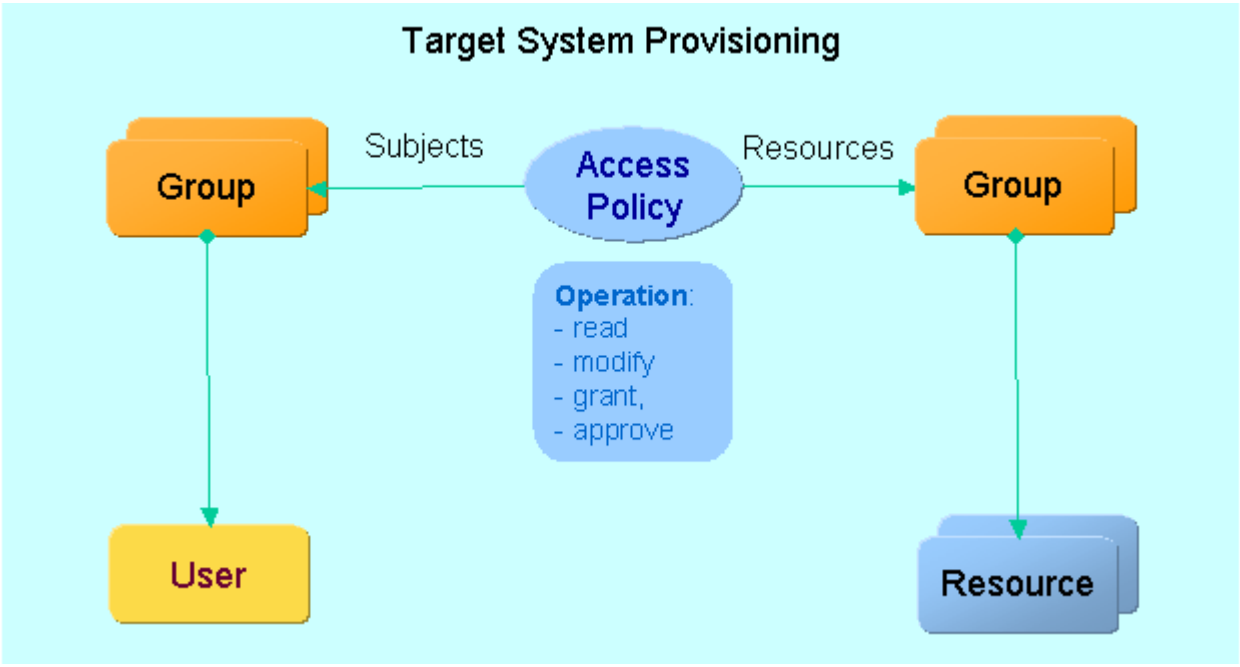


Figure 43. DirX Identity Provisioning Access Policy Structure

An access policy defines the access of subjects to resources through for example the following operations:

- **read** - controls read access to the resource.
- **modify** - controls whether the subject is allowed to modify the resource.
- **grant** - allows assignment of resources (privileges = roles, permissions, groups) to subjects (users).
- **approve** - controls approval of user-to-privilege assignments.

Subjects and resources can be a combination of

- A list of objects
- A list of groups (that each contains a list of objects).



Groups can be populated with members by hand or by automatic provisioning procedures.

- A filter definition (base object and LDAP filter) and an additional list of objects to exclude

Additional match rules can be defined in an LDAP filter style; the following figure shows an example.

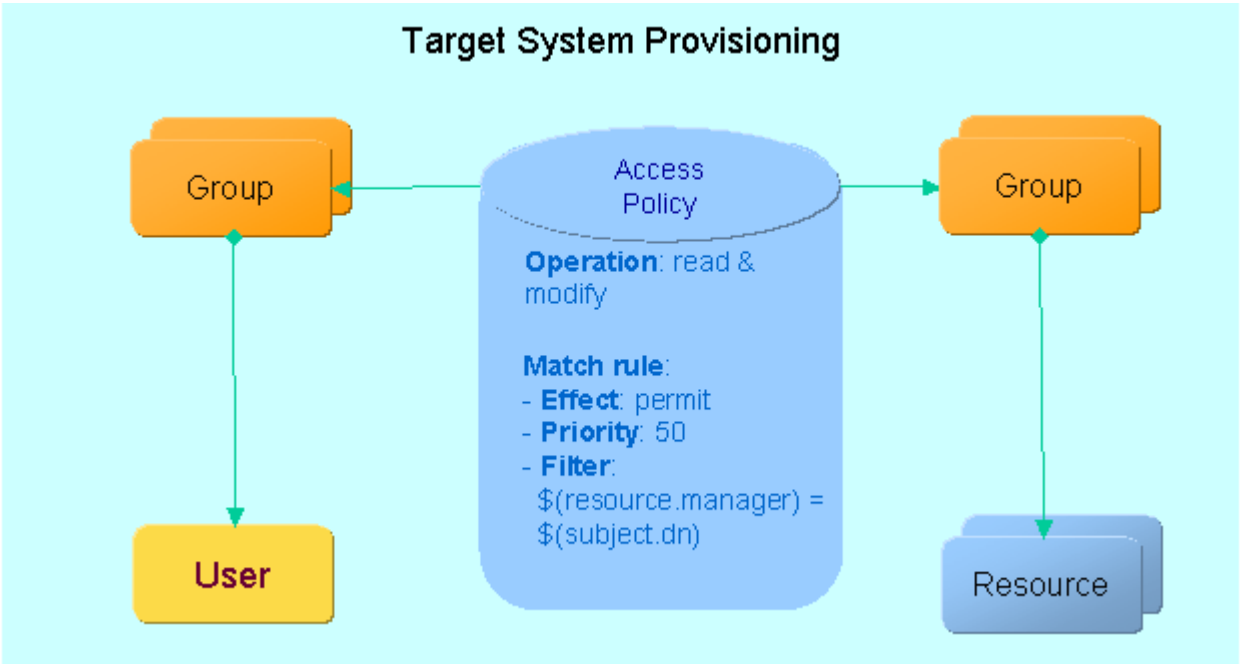


Figure 44. Access Policy with Match Rules

A match rule contains an effect (permit or deny), a priority and a filter. The example shown in the figure checks the manager attribute of the resource against the DN of the subject. This works if the manager attribute of a user points to the manager entry. The rule means that a manager can view and edit all his employees.

10.2. Managing Automatic Provisioning and Consistency Checking

You use DirX Identity Provisioning rule objects to define policies for the automatic provisioning of users and for consistency checking in the Identity Store (Provisioning Configuration). A DirX Identity Provisioning rule performs a search on a set of Provisioning objects according to a search filter, then runs an operation on the retrieved objects. Rule operations are either hard-wired or configurable, depending on the rule type.

You use the Rules and Operations subtree to manage DirX Identity Provisioning rules and rule operations. The following sections describe the types of rules and rule operations and how to use the Rules and Operations subtree to:

- Add, delete, and maintain rules
- Add, delete and maintain rule operations

The chapter also provides information about how rules operate in DirX Identity Provisioning.

10.2.1. Managing Rules

You use the Rules subtree of the Policies tree to manage the rules associated with automatic provisioning and consistency checking. The following figure shows the Rules subtree of the Policies tree.

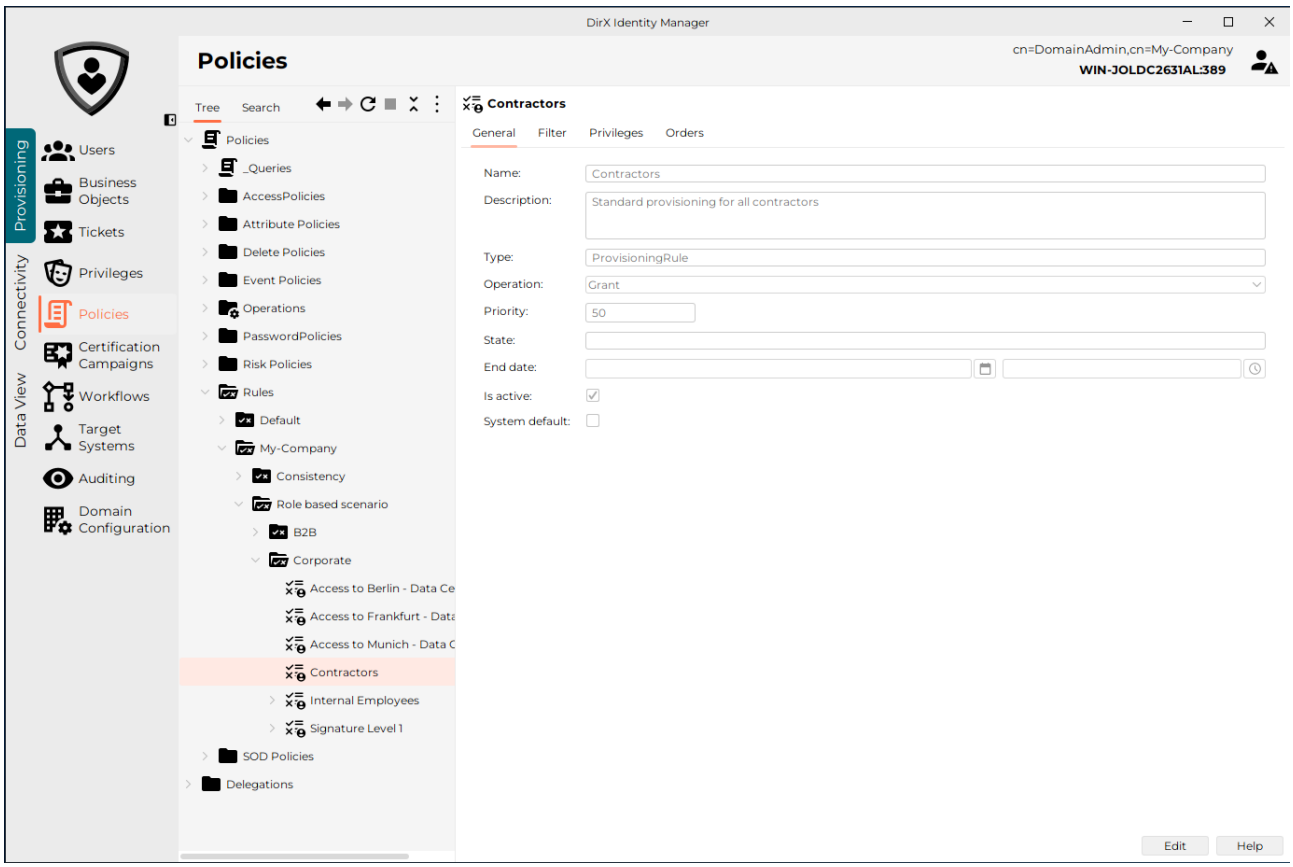


Figure 45. Viewing the Rules Subtree with DirX Identity Manager (Provisioning)

Rule management consists of the following tasks:

- Adding new rules
- Changing the attributes of existing rules
- Deleting rules
- Organizing the rule subtree

The next sections describe the types of rules available in DirX Identity Provisioning and how to manage them with DirX Identity Manager.

10.2.1.1. Understanding Rule Types

DirX Identity Provisioning provides the following rule types:

- Consistency rules
- Provisioning rules
- Validation rules

The next sections describe these rules.

10.2.1.1.1. Consistency Rules

Consistency rules provide a mechanism for checking and purifying the Identity Store (Provisioning Configuration). A consistency rule performs a search on a set of objects

specified in its search filter, then runs a specified operation on the retrieved objects. Consistency rule operations can be implemented as executables (programs or scripts), Java methods, or JavaScript programs. DirX Identity Provisioning supplies a set of default consistency rules (the following list shows only a part of the available rules; see the Provisioning configuration for more rules):

- assocAccount2User - a consistency rule that associates accounts to users via single value attributes.
- assocAccount2UserMV - a consistency rule that associates accounts to users via multi value attributes.
- cleanse-AD-mail-address - a consistency rule that deletes all mail addresses of the professional services users that are not in Windows target systems.
- cleanupDeletedCertificationCampaigns - a consistency rule that removes certification campaigns in the state DELETED without history and whose Status Expiration Date is reached.
- cleanupDeletedObjects - a consistency rule that removes objects in the state TBDEL or DELETED without history and whose End Date is empty or is reached. See the filter definition to view the objects that are handled.
- cleanupOutDatedDelegations - a consistency rule that removes delegations whose dxrDeleteDate is reached.
- cleanupOutDatedRequWflowInstances - a consistency rule that removes request Workflow instances whose dxmStatusExpirationTime is reached.
- cleanupOutDatedTickets - a consistency rule that removes tickets whose dxrDeleteDate is reached.
- CreateFunctionalUsersForUnassignedAccounts - a consistency rule template that creates functional users for all accounts that are not assigned to a user. For information on how to customize the rule, see the section "Automatic Functional User Creation" in the chapter "Customizing Program Logic" in the DirX Identity Customization Guide.
- CreatePersonasForNonPrimaryAccounts - a consistency rule template that creates personas for all non-primary accounts that are assigned to a user. How to customize the rule see section "Automatic Persona Creation" of chapter "Customizing Program Logic" in the DirX Identity Customization Guide.
- CreateUsersForUnassignedAccounts - a consistency rule template that creates users for all accounts that are not assigned to a user. For information on how to customize the rule, see the section "Automatic User Creation" in the chapter "Customizing Program Logic" in the DirX Identity Customization Guide.
- dxrNameFromAdsSAMName4Accounts - a consistency rule that fills the dxrName attribute with the contents of dxmAdsSAMAccountName.
- ListUsersWithErrors - a consistency rule that appends the DNs of users with non-empty error field to the file defined in the errorlist attribute.
- Mail-contractors - a consistency rule that creates an email address for all users that are contractors.
- Mail-internals - a consistency rule that creates an email address for all internal employees.

- removePrivilegesFromDisabledUser - a consistency rule that removes the privilege assignments that are flagged as true from a user in the state DISABLED. You can control the types of privileges (role, permission, group) to be handled.
- removePrivilegesFromDisabledUserWithWhiteList - a consistency rule that removes all privileges of a user in the state DISABLED except for the privileges stored in the white list. You can control the types of privileges (role, permission, group) to be handled. You can also define a set of roles to be excluded in a white list.
- removePrivilegesInBlackListFromDisabledUser - a consistency rule that removes the privileges listed in the black list from a user in the state DISABLED.
- removePrivilegesFromTBDELUser - a consistency rule that removes the privilege assignments from a user in the state TBDEL ten days after the end date is reached. You can control the types of privileges (role, permission, group) to be handled.
- removePrivilegesFromTBDELUserWithWhiteList - a consistency rule that removes the privilege assignments from a user in the state TBDEL ten days after the end date is reached. You can control the types of privileges (role, permission, group) to be handled. You can also define a set of roles to be excluded in a white list.
- removePrivilegesInBlackListFromTBDELUser - removes the privilege assignments from a user in the state TBDEL ten days after the end date is reached. You must define the set of roles to be removed in a black list.
- resolveInconsistentGroupMembership - a consistency rule that searches accounts/groups whose membership attributes are out of sync because many realtime workflows update the same account/group in parallel. If a member exists in both dxrGroupMemberImported and dxrGroupMemberOK, it is removed from dxrGroupMemberImported. If a member exists in both dxrGroupMemberAdd and dxrGroupMemberOK, it is removed from dxrGroupMemberAdd.
- Set-dxrToPeer - a consistency rule that finds an associated account in one of the target systems located under DestinationBase and sets the dxrToPeer attribute.
- Win2k-Mail-address-to-users - a consistency rule that copies all the email addresses of accounts in a Windows system to their users.

You can customize these default consistency rules by defining your own rule operations for them, and you can also create your own consistency rules. The section "Adding Operations" provides more details about how to create your own operations for consistency rules and integrate them into DirX Identity Provisioning.

10.2.1.1.2. Provisioning Rules

Provisioning rules are the most common type of rule. A provisioning rule allows you to define privileges for a well-defined set of user objects. A provisioning rule applies a search filter to DirX Identity user objects and then performs a grant or deny operation, either granting or denying a list of privileges specified in the rule.



Personas and functional users also possess the dxrUser object class. Thus, the existing provisioning rules act on personas and functional users as well. If you want to exclude personas / functional users from the result, you must AND the filter **(!(objectClass=dxrPersonas))** /

(!(objectClass=dxrFunctionalUser)).

A provisioning rule with the **grant** operation assigns the specified list of privileges to each user who matches the conditions of the rule's search filter. A privilege is not assigned if a deny rule with higher priority exists that refers to the same user and the same privilege.

A provisioning rule with the **deny** operation (deny rule) prevents the users from receiving a privilege due to a rule with lower or equal priority. It does not remove privileges that are assigned manually. You can specify a list of privileges and a set of privilege filters in the deny rule; both of these items are used in combination.

You can grant or deny any combination of privileges (roles, permissions or groups). Unlike the rule operations for consistency rules, the grant and deny provisioning rule operations are not configurable.



Roles that require parameters cannot be used within provisioning rules (automatic population of parameters does not make sense and is therefore not supported).



A grant rule results in two actions:

- First all users are searched with the defined filter that do not have the required privilege(s).
- Then all users are searched with an auto-revoke filter who have this right assigned but who should no longer have it.

This works for filters like **(dxrPrivilegeGrantedLink=<dn1>)**. This filter is converted to the auto-revoke filter **(!(dxrPrivilegeGrantedLink=<dn1>))**. This works well unless the attribute is only changed. It does not work for an empty attribute. In this case, you should change the filter to **(!(&(dxrPrivilegeGrantedLink=<dn1>)(dxrPrivilegeGrantedLink=*)))**.

10.2.1.1.3. Validation Rules

Validation rules are applied to target systems and are typically used with the target system validation workflows that detect deviations between the local target system's accounts, groups, and account-group memberships and the same information in the Identity Store (Provisioning Configuration). Validation rules allow you to detect imported attribute-based privileges and to accept or clean up imported privileges not granted by DirX Identity Provisioning.

As described in "Monitoring Accounts and Account-Group Assignments", newly imported accounts, groups, and account-group memberships are flagged with the IMPORTED state. Their states change to the ENABLED state when user-account and account-group assignments are made as part of DirX Identity Provisioning user-privilege assignment and privilege resolution. As described in "Handling the Results of the Validation Workflow", you need to follow up periodically on the validation workflow reports on accounts, groups, and account-group memberships that remain in the IMPORTED state. The validation workflows allow you to automate the maintenance tasks discussed in "Handling the Results of the Validation Workflow" that you otherwise must perform by hand.

Three types of validation rules exist:

- Accept rules
- Cleanup rules
- Validate rules

Validation rules of type "**accept**" search groups with imported members and change the member state to **ENABLED** if the account matches the account filter of the rule. Validation rules of type "**cleanup**" take the same imported memberships and accounts, but change the state to **DELETED**. The next synchronization workflow deletes these memberships in the target system.

Validation rules of type "**validate**" search attribute-based privileges in a target system and add the retrieved accounts into the appropriate groups in the **IMPORTED** state. "Validate" rules use "OnValidation" filters stored in group or obligation objects. When evaluating a "validate" rule, the rule processor first searches for groups or obligations that contain an "OnValidation" filter. The group filter of the rule restricts this search. Using this validation filter, the rule processor then searches for accounts. Each account in the result set is added as a group member in **IMPORTED** state if the account matches the account filter of the "validate" rule.

"Validate rules" can be used, for example, to detect mailbox-enabled Active Directory accounts. Active Directory users become mailbox-enabled by setting some mail attributes, such as the mail address, the home MTA or home MDB. The mailbox-enabled state usually represents an access right within a Windows system and thus should be controlled via a DirX Identity Provisioning group. Attaching an "OnValidation" filter to this group and applying the "validate" rule for the target system allows DirX Identity Provisioning to detect these accounts and make them members of the group.

DirX Identity Provisioning supplies a set of default validation rules (the following list might show only a part of the available rules; see the Provisioning configuration for more rules):

- **Accept-PC-Mailboxes** - accept mailboxes for all employees of the Product Development department.
- **Cleanup-non-local-mailboxes** - reject mailboxes for all non-local employees of all Windows target systems.
- **Mailbox-Enabled-Users-in-Windows2000** - discover all mailbox-enabled users in Windows target systems

10.2.1.2. Adding Rules

To add a new rule:

1. Right-click a rule folder in the rule subtree or the top-level **Rules** folder.
2. Select **New** and then select the type of rule to create: **Consistency Rule**, **Provisioning Rule** or **Validation Rule**.

To add a consistency rule:

1. Enter a **Name** for the consistency rule and describe the rule in the **Description** field.
2. Select the correct **Operation** via the object select dialog (the operation must exist).
3. Enter the **Parameter Values** that are appropriate to the operation you selected. View the **Type** and **Description** fields to understand the meaning of the parameters. Stretch the dialog for better readability if necessary.
4. Set the **Is Active** flag, because otherwise the rule cannot be executed.
5. Click the **Filter** tab and set the correct filter parameters to retrieve the subjects. Use the "+" button to add extra filter conditions.
6. Click **OK** to create the rule or **Cancel** to abort the creation process.

To add a provisioning rule:

1. Enter a **Name** for the provisioning rule and describe the rule in the **Description** field.
2. Select the operation type: use **grant** if you want to assign privileges, use **deny** if you want to forbid privileges.
3. Set a **priority**. 0 is the lowest priority, 100 is the highest priority. Setting a priority allows you to define a precedence between provisioning rules that refer to the same subjects and privileges.
4. Set the **Is Active** flag, because otherwise the rule cannot be executed.
5. Click the **Filter** tab and refine the filter condition if you don't want to use the entire user tree as subjects. Use the "+" button to add extra filter conditions.
6. Click the **Privilege** tab and select any number of privileges (roles, permissions or groups) to be granted or denied. In deny mode you can also choose a filter condition.
7. Click **OK** to create the rule or **Cancel** to abort the creation process.

To add a validation rule:

1. Enter a **Name** for the validation rule and describe the rule in the **Description** field.
2. Select the operation type: use **validate** if you want to detect attribute-based privileges, use **accept** if you want to accept imported privileges or use **cleanup** if you want to delete them.
3. Set the **Is Active** flag, because otherwise the rule cannot be executed.
4. Click the **Accounts** tab and define a filter for the accounts to be used as rule subjects. Use the "+" button to add extra filter conditions.
5. Click the **Groups** tab and define a filter for the groups to be searched for imported members (operations accept or cleanup) or onValidation filters (operation: validate). Use the "+" button to add extra filter conditions.
6. Click **OK** to create the rule or **Cancel** to abort the creation process.

Once you have added a rule to the operation subtree, you can copy it, move it, or rename it. To copy a rule, click it, then select **Copy Object** from the context menu or menu bar. To move a rule, click it, then select **Move Object** from the context menu or menu bar. To rename a rule, click it, then select **Rename** from the context menu or menu bar.

10.2.1.3. Changing Rules

You can change all the parameters of a rule. Click **Edit** to make your changes. DirX Identity Manager (Provisioning) keeps the changes you make to the rule in its internal cache until you click **Save**. As long as you do not use **Save**, you can use the Manager's **Reset** function to restore the rule settings to their old values (the values that are stored in the Identity Store (Provisioning Configuration)).

The changes to a rule take effect when it is next executed, as follows:

- For provisioning rules, privileges that are no longer valid are revoked, and new valid privileges are granted.
- For all other rule types, the changed rule is simply executed. Effects from the previous version of the rule are not removed.



Rules that are checked as System Default can be copied but cannot be changed.

10.2.1.4. Deleting Rules

You can delete rules that do not have the System Default field checked:

- For consistency rules, the rule is simply deleted. There is no mechanism that removes the effect of the consistency rule automatically. You need to define a special rule that performs this task and run it once.
- For provisioning rules, DirX Identity Provisioning removes all privileges given by this rule from the Identity Store (Provisioning Configuration) (but only if a privilege is not given by another mechanism, for example, manually) and then deletes the rule.
- For validation rules, the rule is simply deleted. There is no mechanism that removes the effect of the validation rule automatically. You need to define a special rule that performs this task and run it once.

10.2.1.5. Organizing Rules

A rule folder is a DirX Identity Provisioning object that the rule administrator can use to organize the rule subtree (the tree view of the defined rules in the "Rules" subtree that is displayed when "Policies" is selected in the Manager). Placing a rule in a rule folder does not affect the rule's priority. The purpose of the rule folder is to make it easy for the administrator to select a set of rules later on and execute them with a rule processor workflow.

You can make the following changes to the hierarchical tree view of the DirX Identity Provisioning rule structure:

- Add a new empty rule folder
- Copy a rule folder (with all its content)
- Move a rule folder (with all its content)
- Delete a rule folder (with all its content)

- Rename a rule folder or change its description

To add a new folder, click the root rule folder or one of the folders below it, and then select **New → Rule Container** from the context menu. Manager prompts you to enter the unique name for the folder.

To copy an existing folder, click it, then select **Copy Object** from the context menu or menu bar.

To move an existing folder, click it, then select **Move Object** from the context menu or menu bar.

To change an existing folder, click it, and then click **Edit**.

To delete an existing folder, click the folder, and then select **Delete** from the context menu or menu bar. The folder to be deleted must not contain any rule objects.

To rename a rule or a rule folder, click it, then select **Rename** from the context menu or menu bar.



You cannot put operations into rule folders.

10.2.2. Managing Consistency Rule Operations

You use the Operations subtree of the Policies tree to manage the operations associated with consistency rules. The following figure shows the structure of the Operations subtree.

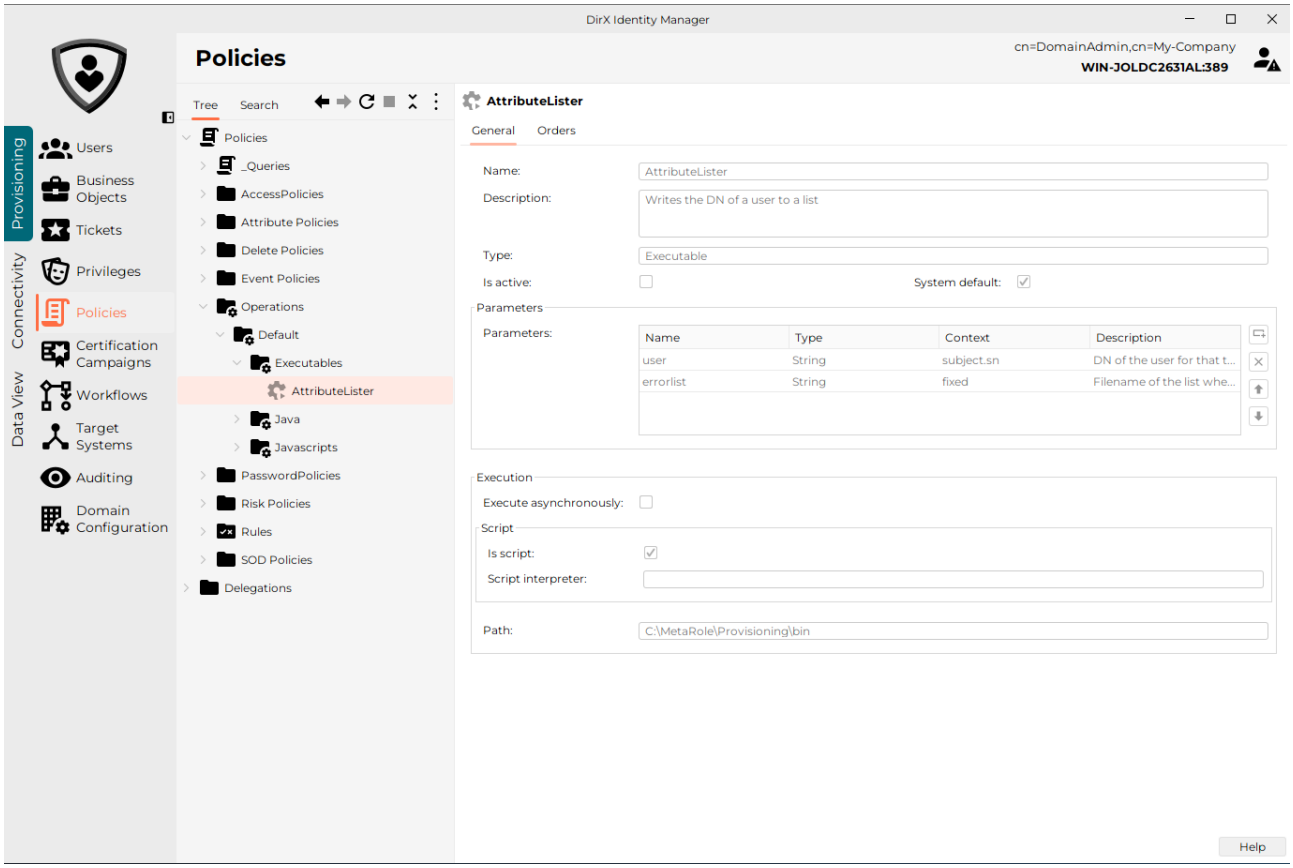


Figure 46. Viewing the Operations Structure with DirX Identity Manager (Provisioning)

Consistency rule operations management consists of the following tasks:

- Adding new consistency rule operations
- Changing the attributes of existing operations
- Deleting operations
- Organizing the operations subtree

The next section describes the types of operations available in DirX Identity Provisioning and how to manage them with DirX Identity Manager.

10.2.2.1. About Operation Types

The following types of operations can be used in consistency rules:

- Operations based on executables (programs and scripts)
- Operations based on Java methods
- Operations based on JavaScript programs

The next sections describe these operation types and how to use them in consistency rules.

10.2.2.1.1. Using Operations Based on Executables

Consistency rule operations can be based on programs or scripts that are external to the LDAP directory in use with DirX Identity Provisioning. The executable must either reside on

the system that runs the DirX Identity policy execution agent or be accessible via a shared file system. Operations based on executables are located in the Executables subtree underneath the Operations subtree in the Policies view.

Script operations must be started with a command interpreter. DirX Identity Provisioning provides the following default script command interpreters:

- **cmd.exe start/C** (Windows)
- **/bin/sh -c** (Linux)

You can also use your own script interpreter. The Windows command interpreter transfers parameters as an array, which avoids problems with spaces in parameters. The UNIX command interpreter requires a single command string. To avoid problems with spaces or special characters like * or $, the parameters are enclosed in single quotes (').

You can also control whether the rule processor starts the executable sub-process synchronously or asynchronously.

If the sub-process is executed synchronously, all output is sent to **stdout** as log messages. All output to **stderr** is output as error messages. The exit code is checked. Values not equal to zero are reported as errors.

If the sub-process is executed asynchronously, its output is muted to avoid mixed logging. The exit code is not checked because the external process can run longer than the internal one (the rule processor).

We recommend that you use synchronous mode whenever possible.



The rule processor environment is only available to UNIX script sub-processes in a UNIX environment.

10.2.2.1.2. Using Operations Based on Java Methods

You can also use Java methods to implement operations for consistency rules. Java operations are methods of a Java class. When processing a Java operation, DirX Identity Provisioning creates the class using its default constructor and then calls the method. As a result, the method is assumed to be self-sufficient: it only works with its input parameters.

To use Java methods as rule operations:

1. Implement each Java method to be used as a rule operation. The action class must implement the ActionConfig interface defined in the package **siemens.dxm.ruleprocessing.actionrunner**. This interface contains the abstract method setContext(RuleContext context) which the rule processor uses to pass the context to the action. We recommend that the context object be held as a class member to make use of its methods. The methods provided by the context object include:
   - **getSubject()** - the search result object to be processed by the operation
   - **getSimulateOnly()** - true when the policy agent runs in simulation mode. The action's implementation is responsible for handling simulation correctly.

- **getUserStorage()** - returns the user storage for the modification of users, accounts and groups.
- **getRuleStorage()** - returns the rule storage for the modification of policies.

2. Build the Java class by creating a jar file *myAction*.jar* that contains the compiled action class.
3. Update the runtime environment for the policy execution agent by adding the jar file *myAction*.jar to the policy agent's class path, in **MetaRolePolicyAgent.bat** (Windows) or **MetaRolePolicyAgent.sh** (Linux).
4. Configure the Java class and methods (as Java Method Actions) with DirX Identity Manager (Provisioning)
5. Create a consistency rule with DirX Identity Manager (Provisioning) that references one of the methods (a Java Method Action). The action is then called once for each object retrieved by the rule's search filter.

Here is an example of a rule operation implemented as a Java method:

```
/**
 * setSubjectAttribute
 *
 * @param node
 * @param attribute
 * @param namingRule
 * @return Object
 */
public Object setSubjectAttribute(StorageObject node, String
attribute, String namingRule) {
    String value = null;
// Setting attribute={0} for user={1} according to rule ''{2}''.
    logger.log(MsgID.DBG_SET_ATTRIBUTE_NR, attribute, node.getID(),
namingRule);
    try {
        value = replaceSubjectAttributes(node, namingRule);
    } catch (RunnerException e) {
// Attribute value for attribute {0} not set at subject {1}.
// Attributes from naming rule {2} are missing on the subject.
        logger.log(MsgID.WAR_ATTR_NOT_SET, attribute, node.getID(),
namingRule);
        return new Integer(1);
    }
    if (context.getSimulateOnly() == true) {
// SIMULATE: Adding property ''{0}'' with value ''{1}'' to subject
DN={2}.
```

```
        logger.log(MsgID.LOG_SIM_ADD_PROPERTY, attribute, value,
node.getID());
    } else {
// Adding property ''{0}'' with value ''{1}'' to subject DN={2}.
        logger.log(MsgID.LOG_ADD_PROPERTY, attribute, value, node
.getID());
        node.setProperty(attribute, value);
        try {
            node.save();
        } catch (StorageException e) {
            return new RunnerException(e.getMessage(), e);
        }
    }
    return value;
}
```

10.2.2.1.3. Using Operations Based on JavaScripts

You can also use JavaScript programs to define operations for consistency rules. The policy execution agent loads them at runtime and provides their input parameters in the context. Your JavaScript operation can use simple parameters or DirX Identity Provisioning Java classes, especially those for the user, the account and the group.

DirX Identity Provisioning extends the ScriptContext abstract class with the following methods:

getObject() - returns the subject.

getParameter(String *name*) - returns the parameter that is defined in the operation definition under this name.

getUserStorage() - returns the user storage for the modification of users, accounts and groups.

getRuleStorage() - returns the rule storage for the modification of policies.

getSimulateOnly() - returns the value of the simulate flag. The JavaScript program is responsible for correctly using this method.

The following log methods are also provided to permit you to output messages via the DirX Identity Provisioning logging facility:

logDebug(String message) - produces a debug-level message. If the rules are processed in the IdS-J server, you need to set the following log level to see debug-level messages:

siemens.dxm.ruleprocessing.actionrunner FINEST

You should see other log levels by default unless you have set a contradictory log level.

- logInfo(**String *message***) - produces an informational message.
- logWarning(**String *message***) - produces a warning level message.
- logError(**String *message***) - produces an error level message.

JavaScript recognizes the array type "Array". This type is not compatible with the Java String Array and thus cannot be used for the setProperty operation. A Java array can be instantiated as described in the example below:

```
...
var length = array.length;
...
// Instantiate String[] here
var JavaStringArray = Java.type("java.lang.String[]");
var attrs = new JavaStringArray(length);
```

Here is an example of a rule operation implemented as a JavaScript program:

```
// creates an Email-Address
// modifies the user
obj=scriptContext.getObject();

gn=obj.getProperty("givenName");
sn=obj.getProperty("sn");
domain = scriptContext.getParameter("domain");

if (sn == null) {
mail = null;
}
else {
if (gn == null) {
mail=sn+"@"+domain;
}
else {
mail=gn+"."+sn+"@"+domain;
}
}
if (scriptContext.getSimulateOnly() == true) {
scriptContext.logInfo("SIMULATE: Adding mail="+mail+" to user
"+obj.getID());
```

```
}
else {
obj.setProperty("mail", mail);
scriptContext.logInfo("Adding mail="+mail+" to user "+obj.getID();
scriptContext.logInfo("Saving user "+obj.getID());
try {
obj.save();
}
catch(Exception) {
scriptContext.logError("Save of user failed."+obj.getID());
}
}
```

10.2.2.2. Adding Consistency Rule Operations

You use the Operations subtree in the Policies view of DirX Identity Manager (Provisioning) to add executable, Java, and JavaScript operations for use in consistency rules. The next sections describe how to use DirX Identity Manager to add the information about these different types of operation into the Identity Store (Provisioning Configuration).

Once you have added an operation to the Operations subtree, you can copy, move, or rename it. To copy an operation, click it, then select **Copy Object** from the context menu or menu bar. To move an operation, click it, then select **Move Object** from the context menu or menu bar. To rename an operation, click it, then select **Rename** from the context menu or menu bar.

10.2.2.2.1. Adding an Executable Operation

To add an executable as a rule operation:

1. In the Policies view, right-click the **Executable Operations** folder in the operation subtree or click the top-level **Operations** folder
2. Select **New**, and then select **Executable Action**.
3. Enter a **Name** for the executable action and describe the action in the **Description** field.
4. Define the **Path** to the executable. If the path is known from the operating system path variable, the name of the executable is sufficient.
5. In **Parameters**, specify the executable's parameters. Add a line for each parameter and set the **Type**, **Context** and **Description** fields. In **Type,** select **String**. In **Context**, select **fixed** if you want to specify the parameter value at design time; that is, in the operation. Otherwise, you can select the object of the search result itself (select **subject**) or one of its attributes. The field offers you the list of attributes in the format **subject.** *attribute_type*; for example, **subject.sn**.
6. Check **Execute asynchronously** if your operation is to execute asynchronously (not recommended and only for UNIX).

7. Check **Is Script** if your operation is a (batch) script and must be started using a command interpreter.
8. For script operations, specify in **Script interpreter** one of the default interpreters (**cmd.exe start /C** for Windows and **/bin/sh -c** for UNIX) or specify your own command interpreters with its parameters.
9. Set the **Is Active** flag, because otherwise the operation (and all rules that use this operation) cannot be executed.
10. Click **OK** to create the operation (or **Cancel** to cancel the creation process).

10.2.2.2.2. Adding a Java Operation

Adding Java methods as rule operations consists of two steps:

1. Creating the Java class
2. Adding the Java methods as sub-objects of the Java class

To create a Java class with DirX Identity Manager (Provisioning):

1. In the Policies view, right-click a **Java** operation folder in the Operations subtree or click the top-level **Operations** folder.
2. Select **New,** and then select **Java Class**.
3. Enter the fully-qualified Java class name in **Name** and describe the class in **Description**.
4. Enter the class or library in **Java class path**. If the class or library is in the policy execution agent's class path settings, specifying the class name alone is sufficient. Otherwise, enter the path to the class or library.
5. Set the **Is Active** flag, because otherwise the operation (and all rules that use this operation) cannot be executed.
6. Click **OK** to create the operation (or **Cancel** to cancel the creation process).

Next, you add a Java Method Action entry to your new Java class entry for each Java method in the class that is to be used as the operation. To create a Java Method Action entry:

1. Right-click the newly created Java class and select **New Java Method Action**.
2. Enter the name for the method in **Java method** and describe it in **Description**.



DirX Identity Provisioning does not support the definition of multiple methods that use the same name but specify different parameters.

3. Create the parameter list for the method in **Parameters**. Add a line for each parameter and set the **Name**, **Type**, **Context** and **Description** fields.

   In **Type**, select one of the Java standard types (String, Integer, Vectors of standard types, and so on) or a DirX Identity Provisioning class. Click the down-arrow to see a list of types.

In **Context**, specify the parameter's context. A parameter's context can be **fixed** or **subject**. A fixed parameter is a constant whose value is defined at the moment you enter it. The declaration is made for the action, but the value is defined by editing the related rule. Thus, the same action can be called from different rules using different sets of fixed parameters. A subject parameter is a value that is derived from each individual search result object (subject) at runtime. If the context is **subject**, the subject itself is passed to the method. If the context is $(**subject**.*attribute_type*), the attribute *attribute_type* is read from the subject at runtime and its value is passed to the method. Click the down arrow to see a list of attribute types.



At runtime, the method is retrieved using the Java Reflection API.

Therefore, the **method name** and **parameter list** specified in the **Java Method Action** entry must exactly match the name and parameters of the Java method that implements the operation.

4. Set the **Is Active** flag. If this flag is not enabled, the operation—and any rules that depend on it—will not execute.
5. Click **OK** to create the operation, or **Cancel** to discard the changes.

10.2.2.2.3. Adding a JavaScript Operation

To add a JavaScript program as a rule operation:

1. In the Policies view, right-click the **JavaScript Operations** folder in the operation subtree or click the top-level **Operations** folder
2. Select **New**, and then select **JavaScript Action**.
3. Enter a **Name** for the JavaScript operation and describe it in **Description**.
4. Define all the **Parameters** that the JavaScript requires. Set the **Name**, **Type**, **Context** and **Description** fields as described earlier in this topic.
5. Set the **Is Active** flag, because otherwise the operation (and all rules that use this operation) cannot be executed.
6. Click the **Implementation** tab and enter the JavaScript code that defines the operation.
7. Click **OK** to create the operation (or **Cancel** to cancel the creation process).

10.2.2.3. Changing Operations

You can change any attribute of an operation object by opening the operation object and clicking **Edit**. DirX Identity Manager (Provisioning) keeps the changes you make to the object in its internal cache until you click **Save**. As long as you do not use **Save**, you can use the Manager's **Reset** function to restore the settings to their old values (the values that are stored in the Identity Store (Provisioning Configuration)). The changes you make to an operation take effect when it is next executed.



The effect of the previous parameters is not automatically reversed. You need to define a special operation and a rule that performs this task and run it once.



Operation objects whose System Default flag is set can be copied but cannot be changed.

10.2.2.4. Deleting Operations

You can delete an operation whose System Default field is not checked.



There is no automatic mechanism that removes the effect of this operation automatically. You need to define a special operation and define a rule that performs this task and run it once.

10.2.2.5. Organizing Operations

An operation folder is a DirX Identity Provisioning object that the policy administrator can use to organize the operations subtree (the tree view of the defined operations in the "Operations" subtree that is displayed when "Policies" is selected in the Manager). The purpose of the operation folder is to make it easy for the administrator to manage the viewable operation subtree.

You can make the following changes to the hierarchical tree view of the DirX Identity Provisioning operation structure:

- Add a new empty operation folder
- Copy an operation folder (with all its content)
- Move an operation folder (with all its content)
- Delete an operation folder (with all its content)
- Rename an operation folder or change its description

To add a new folder, click the root operation folder or one of the folders below it, and then select **New** → **Operation Container** from the context menu. Manager prompts you to enter the unique name for the folder.

To copy an existing folder, click it, then select **Copy Object** from the context menu or menu bar.

To move an existing folder, click it, then select **Move Object** from the context menu or menu bar.

To change an existing folder, click it, and then click **Edit**.

To delete an existing folder, click the folder, and then select **Delete** from the context menu or menu bar. The folder to be deleted must not contain any rule objects.

To rename an operation or an operation folder, click it, then select **Rename** from the context menu or menu bar.



You cannot put rules into operation folders.

10.2.3. About Rule Operation

This section describes:

- How to run rules
- How rule processing works
- How to optimize rule operation

10.2.3.1. Running Rules

You run the policy execution workflow to execute rules. You can define single rules or a set of rules contained in a rule folder.

To execute a set of rules:

- Create a new scenario in the DirX Identity Manager (Connectivity view group) Global View (if not yet done).
- Create your Identity Store connected directory from the Identity Store template (for example, name it **MyIdentityStore**). Create a second connected directory icon and assign **MyIdentityStore** to it.
- Connect these two connected directories with a workflow line.
- Right-click the workflow line and then select **New**.
- Select the **PolicyExecution** template from the list.
- Define a **Name** for your copy in the **General Information** tab (for example, MyPolicyExecution).
- Set either **Process Rules** or **Simulate Rules** from the **Policy Agent Parameters** tab. We recommend that you test new rules in simulate mode (this shows what would happen but does not change any real data). We also recommend that you turn off privilege resolution (perform a separate activity for this task in the same workflow or as a separate workflow in a nested workflow).
- In the **Rule Search Parameters** tab, set the **Base Object** and a **Search Filter** condition that defines the set of rules you want to execute.
- In the **Tracing** tab, set the trace parameters. We recommend that you trace only Errors.
- Click **Finish** to save your configured workflow.

Execute the workflow to run the rule processor. Check the result in the Monitor View. If necessary, retry the workflow with other tracing parameters.

10.2.3.2. How Rules Work

The rule processor searches the rules to be processed by a search filter expression. It processes all types of rules that match the search filter in one run. The rules are processed in the following steps:

The rules are separated according to their different types:

- Consistency rules are stored in a list in the sequence in which they are returned from the search
- Provisioning rules are gathered in rule clusters. For each privilege to be granted or denied, a separate cluster is created that contains all rules for that privilege.
- Validate rules are stored in a list in the sequence in which they are returned from the search.
- Accept/Cleanup rules are gathered in one rule cluster.

After it reads in all the rules, the rule processor executes them in the following sequence:

- Provisioning rules clusters - For each cluster, processes the resulting granting rule first, then processes the deny rule.
- Consistency rules - processes them in the order they were returned by the rule search.
- Validate rules - processes them in the order they were returned by the rule search.
- Accept/cleanup rule clusters - processes each cluster. The cleanup rule that has the higher priority wins.

Sorting of rules:

A list of sort attributes can be specified in a rule search. This configuration affects the sequence in which the rules are returned by the search and thus the sequence they are run within one category. The overall sequence grant/deny ⇒ consistency ⇒ validate ⇒ accept/cleanup rules is not changed by sorting.

Hierarchy issues:

The statement that all rules acting on the same privilege are combined to two searches is only true when the search bases of the user filters are equal.

For different search bases, two searches are performed for each search base, combined only of those filters that are valid in this tree.

Suppose the user store is structured according to organizational units. And for each organizational unit, a separate set of rules is provided, granting the same privilege. In this case, the rule processor will create sub-clusters for each ou that are processed separately and independently.

Next, suppose a grant rule that is valid in the whole organization is combined with a deny rule that is valid only on an organizational unit that is located in a subtree of the organization. In this case, the rule processor clusters and executes the rules bottom-up:

- Step 1: The organization-wide grant rule is combined with the more localized deny rule and executed on the scope of the deny rule in step 1.
- Step 2: Now the organization-wide grant rule is executed on the scope of the organization (without consideration of the deny rule). In this step, the users that were processed in step 1 are skipped, since the single grant rule must not be applied to them.

Simulate Mode

When the rule processor runs in **simulate** mode, all changes are simulated but are not applied.



In simulation mode, all rules are processed independent of the setting of their isActive flag; in **process** mode, only rules and operations with isActive=true are processed. As a result, simulation will show different results if some provisioning rules are flagged isActive, while others are not: In **process** mode, the inactive rules are simply ignored.

Simulate mode allows you to check new rules in a productive environment without affecting the Identity Store (Provisioning Configuration). After the run you can check the simulated results in the real environment. If everything is ok, set process mode and run the rule processor again.

10.2.3.2.1. How to Process Provisioning Rules

Each provisioning rule is defined by a search filter that is applied to DirX Identity users, by its operation type (grant or deny), its priority and the set of privileges it grants (or denies).

The rule processor combines all grant- and deny-rules for a single privilege into a “rule cluster”. The rule processor creates a combined grant filter and a combined deny filter from the filters of the rules contained in the cluster. Consequently, all grant- and deny-rules for a single privilege are processed in two searches:

- **Grant** - for users matching the grant filter, the privilege is assigned.
- **Deny** - from all users matching the deny filter, the privilege is revoked.

This method guarantees that users are only touched when they do not have that privilege (in case of grant) or if they have the privilege (in case of deny).



Users in state TBDEL are not touched by the provisioning rule algorithm. This means that a user in state TBDEL neither gets new privileges via rules nor loses any privileges that are still assigned. Use the **removePrivilegesFromTBDELUser** consistency rule to remove privileges from users in state TBDEL.

After assignment/revocation of the privilege, a user to group resolution can be performed, thus propagating the effect of assignment/revocation to the affected account and group objects in the DirX Identity Provisioning target systems. We recommend that you do not use this feature because it works correctly only in simple environments with a few rules. In complex environments with many rules, a user is resolved several times after one of the privileges is assigned, which can negatively impact performance. A clear sequence of rule processing and privilege resolution guarantees that you will not run into this problem.

The following additional issues are covered by the clustering algorithm:

- Priority

  A deny rule takes precedence over a grant rule if its priority is greater or equal to that of the grant rule

- Auto-revocation

  A privilege that has been granted to a user is automatically revoked if the user's attributes are changed so that the grant rule no longer matches or a deny rule with higher or equal priority matches.

- Exclude rules

  If a privilege is only referenced by deny rules, an exclude rule is applied, revoking the privilege from all matching users. You can use this method to remove privileges that were previously assigned by a grant rule.

10.2.3.2.2. How to Process Consistency Rules

Consistency rules perform a search on a set of objects that is defined in their search filter. An operation is applied to each subject returned by the search. The operations are assigned to the rule by a DN reference. An operation can be implemented as a Java method, as a JavaScript, or as an executable (program or script).

10.2.3.2.3. How to Process Validation Rules

These rules are processed depending on their type.

Validation Rules

- Searches for groups matching the validation rule's groups filter and which are configured for validation or have an obligation assigned.
- Creates a combined account validation filter from the groups and/or the obligation's validation filters.
- Searches for all accounts matching the rule's account filter. This action defines the set of accounts to be validated.
- For each account, checks if the account matches the account validation filter created in step 2.
- If the filter matches, determines the member state of the account:
  If the account is not already a member of the considered group, its state is set to IMPORTED.
  If its state is DELETE, the member is removed from the group.
- If the filter does not match, determines the member state of the account.
  If ADD/ENABLED, the onAssignment actions are processed, setting the account attributes.
  If IMPORTED/IGNORE, the member state is set to DELETED, and the onRevocation actions are processed.
  If DELETED, the onRevocation actions are processed.

Accept/Cleanup Rules

These rules work in the following way:

- The rule processor collects all accept/cleanup-rules are combined into an

"accept/cleanup rule cluster" and processes them together.

- First, the rule processor searches all imported accounts (paged operation).
- For each account it checks in memory whether a rule applies. If several rules apply, they are sorted by priority.
- If the priority of a cleanup rule is greater than or equal to that of an accept rule, the cleanup rule takes priority.
- The resulting rule is executed. It sets all memberships of all groups that are defined in the rule to the according state (cleanup = DELETED, accept = IGNORE).
- If an account is not affected by a rule, the memberships remain in state IMPORTED and an informational message is written to the log file.

See the section about "Managing Group Memberships" in the chapter "Managing Target Systems" for more information about handling IMPORTED memberships.

10.2.3.3. Optimizing Rule Operation

This section provides hints on how to optimize your rule processing based on the information given in "How Rules Work".

10.2.3.3.1. Separating Rules

You can set up any number of rule processing workflows that operate on a specific set of rules. Here are some guidelines to follow:

- You should carefully plan the set of workflows: define a set of rules that works on a set of subjects. Try to set up a rule environment that is easy to understand, clearly separated and only as complex as necessary.
- Define schedules for these workflows or combine them in a nested batch workflow.
- Separate policy execution and privilege resolution into two succeeding workflows; this configuration guarantees that a user is only resolved once per run and not once per rule.
- If you want to handle large amounts of subjects, distribute the rules, for example, to work on separate user trees.
- Use DirX Identity Connectivity's powerful capability to distribute these workflows on separate DirX Identity servers (perform DirX Identity Agent installations). This configuration allows you to distribute the processing load.

The following filters must be applied for the different kinds of rules if you want to separate rules:

Rule Type	Search Filter
consistency	dxrType=ConsistencyRule
grant / deny	dxrType=ProvisioningRule
validate	dxrOperationImp=validate
accept / cleanup	(|(dxrOperationImp=accept)(dxrOperationImp=cleanup))

Rule Type	Search Filter
all rules together	(|(objectClass=dxrConsistencyRule)(objectClass=dxrProvisionRule))

10.2.3.3.2. Other Hints

If you expect huge changes in your store (for example, a complex organizational change) that affects all or almost all of your users, try to perform the organizational change during night or on a weekend.

10.2.3.3.3. Warnings

- Do not separate rules that belong to the same cluster

  We recommend that you run grant/deny rules in a common run of the rule processor. For all grant/deny rules acting on the same privilege this is a must:

If a grant rule is acting on a special privilege (let's call it Group_One) and a deny rule is acting on the same privilege are processed in different workflows, the rules are not processed in a common rule cluster. Thus, the priorities of the rules are not evaluated, and you encounter the following behavior:

- Every time you run the first workflow, Group_One is granted to a set of users.
- Later, when you run the workflow that contains the deny rule, Group_One is revoked from the users.

Similarly, all accept/cleanup rules running in the same target system must be executed in one common run of the rule processor.

10.3. Managing Object Policies

DirX Identity provides several types of object-related policies:

Event Policies - use these policies to trigger event-based processing of an entry after an add or modify operation.

Attribute Policies - these policies allow starting request workflows if specific attributes have been changed.

Delete Policies - use these policies to trigger request workflows if a specific object type is to be deleted.

The next sections describe in more detail how to set up and use these policies.

10.3.1. Managing Event Policies

Event policies for objects indicate that an event must be sent if an object is created or modified. This event is used by the relevant event-based processing workflow(s). For more information about these workflows, see the chapter "Understanding the Java-based Maintenance Workflows" in the *DirX Identity Application Development Guide*.

Event policies keep configuration information for several object types. Thus, it is not necessary to create and maintain a folder structure. One event policy for a domain is sufficient.

10.3.1.1. Adding Event Policies

To add a new event policy:

- Open the **Policies** view group in the **Provisioning Configuration** and then open the **Event Policies** folder.
- Copy an existing event policy (for example, the delivered default policy). Use the **Copy Object** menu entry from the context menu.
- Alternatively, you can create a new policy. Right-click the Event Policies folder and then select **New** → **Event Policy Configuration** from the context menu.
- Set the **Name** and **Description** fields in the **General** tab. Activate the new policy.
- Click the **Configuration** tab.
- Select one or more object types from the **Available** list and move them to the **Selected** list.
- The OD name represents the name from the corresponding object description. This allows setting up multiple event policies for different object descriptions, for example you might want to distinguish between external and internal users. Both have the same object classes, but the object descriptions can differ in some aspects.
- The **Object Class** column must contain a representative object class that is used by these object descriptions. The meta controller uses the object class because it has no access to object descriptions.
- The **Object Class2** column defines a second object class for this object description type. This definition allows the meta controller to distinguish between users, personas and functional users. Since personas and functional users have the object class dxrUser (as the normal users), the meta controller matches the definition with the best match (a two-object-class match is preferred over a single-object-class match).
- The **Type** column defines another match criterion for the meta controller that evaluates the object's dxrType attribute. Matching is performed in the order
  1) two object classes match
  2) object class and type match
  3) object class matches
- In the last column, you can set this event policy to active.
- Click **OK** to store it or **Cancel** to abort the creation process.

You should restart the DirX Identity Manager to make the changes effective. Otherwise, the **aclmgr.refresh.interval** in the **dxi.cfg** file defines the refresh period (by default, 10 minutes).

Perform a logout / login sequence in Web Center to make the new policy effective.

10.3.1.2. Changing Event Policies

To modify an event policy:

- Open the **Policies** view group in the **Provisioning Configuration** and then open the **Event Policies** folder.
- Select the policy to modify.
- Adapt the attributes in the tabs. Use the Help button for more explanation.



Use the Rename method from the context menu of the policy to rename it.

- Click **OK** to store it or **Cancel** to abort the creation process.

You should restart the DirX Identity Manager to make the changes effective. Otherwise, the aclmgr.refresh.interval in the dxi.cfg file defines the refresh period (by default, 10 minutes).

Perform a logout / login sequence in Web Center to make the new policy effective.

10.3.1.3. Deleting Event Policies

To delete an event policy:

- Open the **Policies** view group in the **Provisioning Configuration** and then open the **Event Policies** folder.
- Select the policy to delete.
- Select **Delete** from the context menu.
- Confirm the message box.

You should restart the DirX Identity Manager to make the changes effective. Otherwise, the **aclmgr.refresh.interval** in the **dxi.cfg** file defines the refresh period (by default, 10 minutes).

Perform a logout / login sequence in Web Center to make the new policy effective.

10.3.2. Managing Attribute Policies

Attribute policies control the modification of critical attributes for a specific object type. If you change one or more of the object's critical attributes, the DirX Identity Services layer automatically starts an approval workflow. The next sections describe the attribute policies for objects and how to organize, change, and delete them.

10.3.2.1. Attribute Policies for Objects

You can configure attribute policies for any important (for example security relevant) attribute of an object. The next paragraphs explain typical use cases.

User attribute values can example affect his access rights in many ways.

- When used as permission parameters, the values of user attributes control the groups

assigned to users. Changing a user attribute used as a permission parameter changes the assigned groups.

- When used in provisioning policies, the values of user attributes control the privilege assignments that the provisioning workflows automatically calculate and make.

Changing the role parameter or permission parameter definition of a role or permission also affects the assignment of groups in target systems.

In both cases, changing one of these attributes is critical. Therefore, we recommend that you define these attributes as subject to approval by defining an attribute policy for the user object class. The DirX Identity Services layer starts approval workflows if one of these attributes is changed.

10.3.2.2. Adding Attribute Policies

To add a new attribute policy:

1. Right-click a rule folder in the subtree or the top-level **Attribute Policies** folder.
2. Select **New** → **Attribute Policy**.
3. Set the **Name** and **Description** fields in the **General** tab. Activate the new policy.



To enable modification approval, you generally set the Attribute modification approval flag at the domain object.

4. Click the **Configuration** tab. Select the object type and set the object class (only used by the meta controller). Select some of the attributes in the **Available** pane and move these attributes down to the **Selected** pane.
5. Click **OK** to store the policy or **Cancel** to abort the creation process.

10.3.2.3. Changing Attribute Policies

You can change all the parameters of a policy. Click **Edit** to make your changes. DirX Identity Manager (Provisioning) keeps the changes you make to the policy in its internal cache until you click **Save**. As long as you do not use **Save**, you can use the Manager's **Reset** function to restore the policy settings to their old values (the values that are stored in the Identity Store (Provisioning Configuration).

Set or reset the **Is Active** flag to enable or disable a policy.

10.3.2.4. Deleting Attribute Policies

You can delete attribute policies at any time. However, if you intend to use the policy again later on, we recommend that you only reset the **Is Active** flag.

10.3.2.5. Organizing Attribute Policies

You can structure policies within a folder hierarchy. We recommend that you create a policy folder for your project.

You can make the following changes to the hierarchical tree view of the DirX Identity Provisioning policy structure:

- Add a new empty policy folder
- Copy a policy folder (with all its content)
- Move a policy folder (with all its content)
- Delete a policy folder (with all its content)
- Rename a policy folder or change its description

To add a new folder, click the root policy folder or one of the folders below it and then select **New** → **Attribute Policy Container** from the context menu. Manager prompts you to enter the unique name for the folder.

To copy an existing folder, click it and then select **Copy Object** from the context menu or menu bar.

To move an existing folder, click it and then select **Move Object** from the context menu or menu bar.

To change an existing folder, click it and then click **Edit**.

To delete an existing folder, click the folder, and then select **Delete** from the context menu or menu bar. The folder to be deleted must not contain any policy objects.

To rename a policy or a policy folder, click it, then select **Rename** from the context menu or menu bar.

10.3.3. Managing Deletion Policies

Delete policies for objects indicate that a request workflow must be started if an object is to be deleted. Set up the corresponding request workflow with operation=delete. For more information about these workflows, see the chapter "Understanding Deletion Workflows" in the *DirX Identity Application Development Guide*.

Delete policies keep configuration information for several object types. Thus, it is not necessary to create and maintain a folder structure. One delete policy for a domain is sufficient.

10.3.3.1. Adding Delete Policies

To add a new delete policy:

- Open the **Policies** view group in the **Provisioning Configuration** and then open the **Delete Policies** folder.
- Copy an existing delete policy (for example, the delivered default policy). Use the **Copy Object** menu entry from the context menu.
- Alternatively, you can create a new policy. Right-click the **Delete Policies** folder and then select **New** → **Delete Policy Configuration** from the context menu.

- Set the **Name** and **Description** fields in the **General** tab. Activate the new policy.
- Click the **Configuration** tab.
- Select one or more object types from the **Available** list and move them to the **Selected** list.
- The OD name represents the name from the corresponding object description. This allows setting up multiple event policies for different object descriptions, for example you might want to distinguish between external and internal users. Both have the same object classes, but the object descriptions can differ in some aspects.
- In the last column, you can set this delete policy to active.
- Click **OK** to store it or **Cancel** to abort the creation process.

You should restart the DirX Identity Manager to make the changes effective. Otherwise, the aclmgr.refresh.interval in the dxi.cfg file defines the refresh period (by default, 10 minutes).

Perform a logout / login sequence in Web Center to make the new policy effective.

10.3.3.2. Changing Delete Policies

To modify a delete policy:

- Open the **Policies** view group in the **Provisioning Configuration** and then open the **Delete Policies** folder.
- Select the policy to modify.
- Adapt the attributes in the tabs. Use the Help button for more explanation.



Use the Rename method from the context menu of the policy to rename it).

- Click **OK** to store it or **Cancel** to abort the creation process.

You should restart the DirX Identity Manager to make the changes effective. Otherwise, the **aclmgr.refresh.interval** in the **dxi.cfg** file defines the refresh period (by default, 10 minutes).

Perform a logout / login sequence in Web Center to make the new policy effective.

10.3.3.3. Deleting Delete Policies

To delete a delete policy:

- Open the **Policies** view group in the **Provisioning Configuration** and then open the **Delete Policies** folder.
- Select the policy to delete.
- Select **Delete** from the context menu.
- Confirm the message box.

You should restart the DirX Identity Manager to make the changes effective. Otherwise, the **aclmgr.refresh.interval** in the **dxi.cfg** file defines the refresh period (by default, 10 minutes).

Perform a logout / login sequence in Web Center to make the new policy effective.

10.4. Managing SoD Policies

You need segregation of duties (SoD) policies to satisfy regulatory compliance. Set up SoD policies to define conflicting privileges. For example, an auditor should not be one of the administrators whose tasks are audited or a person in the procurement department should not be able to approve his own orders. You can mix any type of privilege in an SoD policy; for example, you can define a group conflict with a role. The next sections describe these tasks in more detail. See the section "Managing SoD" for a description of the SoD checking mechanisms provided with DirX Identity and how they work, and the context-sensitive help chapters for a description of the corresponding SoD policy objects and fields.

To understand the SoD concept, read the DirX Identity Use Case Document *Using SoD*.

10.4.1. Adding SoD Policies

To add a new SoD policy:

1. Right-click a rule folder in the subtree or the top-level **SoD Policies** folder.
2. Select **New** → **SoD Policy**.
3. Set the **Name** and **Description** fields in the General tab. Activate the new policy.



To enable SoD checks generally, set the **Segregation of Duties (SoD) checks** flag at the domain object.

4. Create two or more lines in the **Conflicting Privileges** pane and select the corresponding privileges.
5. Click **OK** to store it or **Cancel** to abort the creation process.

All privileges that are referenced from an active SoD policy have the **Has conflicting privilege** flag set (a read-only flag). This flag makes it easy to detect whether a privilege is part of an SoD policy.

10.4.2. Changing SoD Policies

You can change all the parameters of a policy. Click **Edit** to make your changes. DirX Identity Manager (Provisioning) keeps the changes you make to the policy in its internal cache until you click **Save**. As long as you do not use **Save**, you can use the Manager's **Reset** function to restore the policy settings to their old values (the values that are stored in the Identity Store (Provisioning Configuration).

Set or reset the **Is Active** flag to enable or disable a policy.

10.4.3. Deleting SoD Policies

You can delete policies at any time. If you intend to use the policy again later on, we recommend that you only uncheck the **Is Active** flag for the policy. Disabling or deleting an

SoD policy resets the corresponding **Has conflicting privileges** flag at the related privileges if the privilege is not referenced by any other SoD policy.

10.4.4. Organizing SoD Policies

You can structure policies within a folder hierarchy. We recommend that you create a policy folder for your project.

You can make the following changes to the hierarchical tree view of the DirX Identity Provisioning policy structure:

- Add a new empty policy folder
- Copy a policy folder (with all its content)
- Move a policy folder (with all its content)
- Delete a policy folder (with all its content)
- Rename a policy folder or change its description

To add a new folder, right-click the root policy folder or one of the folders below it, and then select **New - > Attribute Policy Container**. Manager prompts you to enter the unique name for the folder.

To copy an existing folder, click it and then select **Copy Object** from the context menu or menu bar.

To move an existing folder, click it and then select **Move Object** from the context menu or menu bar.

To change an existing folder, click it and then click **Edit**.

To delete an existing folder, click the folder and then select **Delete** from the context menu or menu bar. The folder to be deleted must not contain any policy objects.

To rename a policy or a policy folder, click it and then select **Rename** from the context menu or menu bar.

10.5. Managing Password Policies

Password policies define restrictions on password management. Changing a password or resetting a password requires fulfilling the defined password policies for a specific user.

The following capabilities are available for password policies:

- You can create a new password policy only within Web Center.
- You can change a password policy via Web Center and Identity Manager.
- You can define a specific password policy for each user.
- You can set up the default password policy to be used for all users where no specific policy is defined.

- You can define Microsoft Windows-compliant password policies.

For a detailed description of all password policy parameters, see the context-sensitive help.

You can also program your own password checks and deploy them with Web Center. For more details, see "Customizing Password Management" in the *DirX Identity Customization Guide*.

11. Managing Request Workflows

Many DirX Identity features run completely automatically and in the background if you configure them correctly. Nevertheless, today's processes in a company require that data and access rights be controlled by manual interaction and approval or certification by people in the company. To address this requirement, DirX Identity provides request workflows with a comprehensive set of features. You can use these workflows to manage the life-cycle of objects in a controlled way including the necessary approval or certification steps. Request workflow management is the process of defining, maintaining and monitoring workflow processes.

Let's look at the Request Workflows feature from DirX Identity Manager. Log in to DirX Identity Manager's Provisioning view, and then click **Request Workflows**. Manager displays the view shown in the following figure.

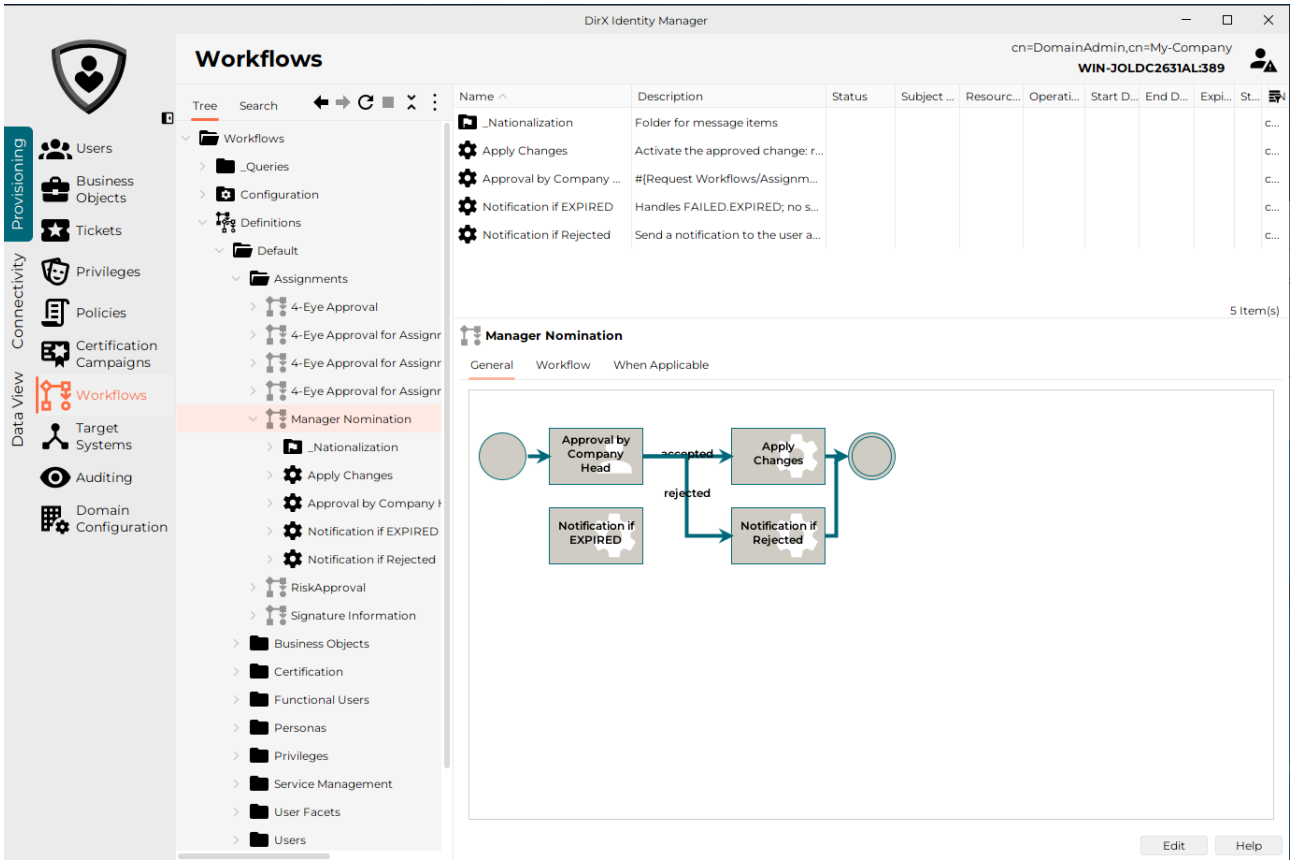


Figure 47. Request Workflows View

The Request Workflows view contains the following subtrees:

- _Queries - contains predefined stored queries
- Configuration - contains central configuration objects (for example, the configuration for the mail server)
- Definition - contains pre-configured workflows that you can use "as is" or as templates for your own request workflows, and contains your workflow definitions.
- Monitor - contains objects that reflect running and completed workflows



If you have used DirX Identity's Approval Workflows feature from previous DirX Identity releases, you will see two additional subtrees: Jobs and Workflows. The Approval Workflows feature still works, but we strongly recommend that you migrate to the Request Workflows feature. It is much more powerful and easier to handle (because it has more intuitive graphical administration).

11.1. Managing Configuration Objects

The request workflow configuration allows you to set global parameters and services that can be used by the request workflow templates, including:

- **Folders** - for structuring the configuration topics.
- **Activity types** - contains the default activity types and your own activity types.See the *DirX Identity Customization Guide* for more information.
- **Common activities** - contains all activities that can be shared.
- **Services** - contains central services; for example, the SMTP service.

Workflow configuration management consists of the following tasks:

- Adding new configuration objects
- Changing configuration objects
- Deleting configuration objects
- Organizing the configuration object subtree

11.1.1. Adding Configuration Objects

To add a new configuration object:

1. Click a folder in the configuration subtree or click the top-level configuration folder.
2. Select **New** → *type* in the context menu.
3. Enter the name of the configuration object and any other general configuration object attributes.

Once you have added a configuration object to the configuration subtree, you can copy it, move it, or rename it. To copy a configuration object, click it and then select **Copy Object** from the context menu or menu bar. To move a configuration object, click it and then select **Move Object** from the context menu or menu bar. To rename a configuration object, click it and then select **Rename** from the context menu or menu bar.

11.1.2. Changing Configuration Objects

You can change any attribute of a configuration object.

To change the value of one or more of the properties displayed in a tab, click **Edit**. Manager keeps the changes you make to the object in its internal cache until you click **Save**. As long

as you do not use **Save**, you can use the Manager's **Reset** function to restore the settings to their old values (the values that are stored in the Identity Store (Provisioning Configuration)).

These changes take effect during the next execution of a workflow.

11.1.3. Deleting Configuration Objects

You can delete configuration objects at any time.

Be sure not to delete default configuration objects.

11.1.4. Organizing Configuration Objects

A configuration folder is a DirX Identity Provisioning object that the workflow administrator can use to organize the configuration subtree (the tree view of the defined configuration objects in the **Configuration** subtree that is displayed when the **Request Workflows** view is selected in the Manager). The purpose of the configuration folder is to make it easy for the administrator to manage the viewable configuration subtree.

You can make the following changes to the hierarchical tree view of the DirX Identity Provisioning configuration structure:

- Add a new empty configuration folder
- Copy a configuration folder (with all its content)
- Move a configuration folder (with all its content)
- Delete a configuration folder (with all its content)
- Rename a configuration folder or change its description

To add a new folder, click the root configuration folder or one of the folders below it, and then select **New - > Folder** from the context menu. Manager prompts you to enter the unique name for the folder.

To copy an existing folder, click it and then select **Copy Object** from the context menu or menu bar.

To move an existing folder, click it and then select **Move Object** from the context menu or menu bar.

To change an existing folder, click it and then click **Edit**.

To delete an existing folder, click the folder and then select **Delete** from the context menu or menu bar.The folder to be deleted must not contain any configuration objects.

To rename a configuration or a configuration folder, click it and then select **Rename** from the context menu or menu bar.

11.2. Managing Request Workflow Definitions

DirX Identity delivers a set of pre-defined request workflows that you can use as templates. To work with them, log in to DirX Identity Provisioning, select **Request Workflows** from the view bar and then select the **Definitions** → **Default** folder. The main task of managing request workflow definitions is to copy the templates that meet your workflow requirements and then change them accordingly.

To view the properties of a request workflow object, click its entry in the tree. When you select such a workflow object, DirX Identity Manager (Provisioning) displays a property dialog for the object. The dialog consists of a set of tabs that you can use to view the object's properties. Click the tabs in the property dialog to move between the different property categories.



The first tab contains the graphical representation of the workflow.

Managing request workflow definitions, then, consists of these tasks:

- Creating your own workflow definition folder structure
- Copying the delivered request workflow templates and modifying them to create your own request workflows
- Adding new request workflows
- Deleting request workflows

The topics in this section describe how to use DirX Identity Manager (Provisioning) to perform them.

11.2.1. Organizing the Definition Subtree

The **Definition** folder presents a set of default workflow templates underneath the **Default** folder. You should create a parallel structure of folders to store your own request workflow definitions. This is important because the original default workflow definition might change from release to release without any notice. Copied or newly created workflows are not changed at all during an upgrade or update installation.

To create a new folder, right-click the relevant folder and select **New** → **Workflow**. We recommend that you first create a project folder and then a set of sub folders that are structured by object types similar to the structure underneath the **Default** folder.

Because your folder structures could be complex, you should use the stored queries in the **_Queries** folder to find all active or inactive workflows or to find workflows of a specific type. You can set up your own queries. Either copy an existing query and modify it or create a new query from scratch.

11.2.2. Copying Request Workflow Definitions

Creating request workflows from scratch is a complex task even if you can use the graphical editor for this task. It is much easier to copy an existing template and modify it

afterwards. Perform these steps to create a request workflow definition from a copied template:

- Select the template that best fits your requirements. You can use one of the default templates or any of your previously created request workflows as starting point.
- Right-click the template, then select **Copy Object**. Select a target location (one of your previously created workflow folders) and define the name of the new workflow. Click **OK** to copy the object.
- Select the new object and change all workflow parameters like **Description**, **Owner** and **When Applicable**. Uncheck the **Active** flag for so long as the new template is not ready to use.
- Select the **Workflows** tab and change all activities as required for the new workflow definition. You can delete, add and modify activities. Alternatively you can open the workflow in the tree and edit the activities from there.

For a description of the graphical workflow editor, see the "Workflow Editor" and "Activity Editor" sections. Use the context-sensitive help if you need more information about the fields of an object.

11.2.3. Adding Request Workflow Definitions

Creating request workflow definitions from scratch requires a good understanding of all the workflow concepts and parameters. For a detailed description of how to create and customize workflow definitions, see Customizing Request Workflows in the *DirX Identity Application Development Guide*.

11.2.4. Deleting Request Workflow Definitions

You can delete request workflow definitions that are no longer of any use to you.To delete a request workflow definition, right-click it and select **Delete**. If you are not sure whether or not you'll need a request workflow definition sometime in the future, then just deactivate it. This action allows you to activate it later on as needed.

11.3. Managing Request Workflow Instances

Running request workflows are displayed underneath the **Monitor** subtree of the **Request Workflows** view. Here you can view and maintain all instances of the workflows.

The workflow instances are shown in equivalent subtree structures of the **Definition** subtree. An additional folder is built as an endpoint for each day in which the workflow ran.

This section also discusses common problems that can occur during workflow execution.

11.3.1. Searching Workflow Instances

Workflow instances keep all important information in LDAP attributes, which allows searching via LDAP filters; for example, with query folders. The most important attributes for searches are:

- **createTimestamp** - the creation time of the workflow instance.
- **description** - the description of the workflow definition.
- **dxmDisplayName** - the human-readable display name of the workflow instance.
- **dxmSpecificAttributes** - keeps additional information about the workflow. Important attributes are:
  - absoluteDisplayName - the human-readable path of the workflow instance (for example: My-Company/Approval/4-Eye Approval/2010-05-20/Abele Marc → Test role - Parameters)
  - owner - the owner DN of the workflow definition.
  - path - the path of the wokflow definition (for example: Definitions/My-Company/Approval/4-Eye Approval).
- **dxmStatusExpirationTime** - the time at which the workflow instance is removed from the Identity Store.

  You must configure a workflow to perform this task.

- **dxrApplicationState** - the application state. For approval workflows, typical values are ACCEPTED or REJECTED.
- **dxrEndDate** - the time at which this workflow instance was completed.
- **dxrEpirationDate** - the time at which this workflow instance runs into the configured time out.
- **dxrObjectType** - the object type this workflow handles (for example, dxrUser).
- **dxrOperationImp** - the operations for which this workflow definition is applicable (for example, create or delete).
- **dxrResourceLink** - the DNs to the resources (if any) that are linked to the subject (for example, a role to a user or a permission to a role).
- **dxrStartDate** - the time at which the workflow instance was calculated by the workflow engine (this is typically later than the creationTimestamp).
- **dxrState** - the status of the workflow instance (for example SUCCEEDED).
- **dxrSubjectLink** - the subject this workflow instance handles (for example, a user or a permission).
- **dxrType** - the request workflow type.
- **dxrUserLink** - the initiator of this workflow instance.
- **modifyTimestamp** - the time of the last modification of this workflow instance.

11.3.2. Viewing Workflow Instances

Workflow instances are shown as a tree node under the respective folder of the day in which the workflow instance was started. These tabs are available:

- The **General** tab shows the graphical view of this instance. Border colors and small symbols within each activity icon indicate the state of the activity as follows:

- **Orange** (no symbol) - activity not yet started.
- **Blue** or  - a running activity.
- **Green** or  - activity successfully completed.
- **Red** or stop symbol - activity failed.

You can right-click an activity instance and then select **Open** to show the details of this activity. See the section "Viewing Activity Instances" for details. For a detailed description of all available options in the graphical workflow view, see the Workflow Editor description in the context-sensitive help.

- **Workflow** - displays global data as the name and description of the workflow instance as well as the workflow type information. It also shows the initiator, the subject and resources of this workflow.

  Some fields can be empty for specific types of workflows.

- **Status information** - shows status and timing related data.
- **Modifications** - displays the pending order information for example the attributes of a new object.
- **Resources** - shows all pending assigned resources, for example the privileges assigned to a new user object.

You can open the workflow object from the tree to see the activities in alphabetical order. Click on an activity instance to display its details.

11.3.3. Viewing Activity Instances

Activity instances are either shown as a tree node underneath the respective folder of the day on which this instance was started or as icons within the graphical view of the workflow structure. If you open one of the activities, the following tabs are available:

- **General** - displays global data as the name and description of the activity instance and the activity type information and the start condition.
- **Status information** - shows status and timing related data and the relevant participant(s) and all error handling parameters.

11.3.4. Managing Problems with Workflow Instances

Problems with request workflow instances can sometimes occur, depending on how complex the workflow definition is. This section describes common problems and how to solve them, including how to:

- Change participants
- Restart workflows

11.3.4.1. Changing Participants

There are many reasons why a participant in a running workflow may not be available and has not set up the proper delegation for approval in his absence; for example, he has suddenly become ill, or he has had an accident. To mitigate this situation, you can establish escalation procedures for an activity. If you have not done so, you must change the approver by hand. You can use the Manager or Web Center to change a participant. The following sections provide instructions.

Changing a Participant with DirX Identity Manager

If the workflow is still in the state **Running**, you can use this method to change a participant. If the workflow has timed out, you must start the entire workflow from the beginning (this action creates a new instance).

For a workflow in the **Running** state, an administrator can perform this procedure:

- Navigate to the relevant activity instance within the workflow.
- Click the **Status Information** tab.
- Click **Edit** and then select the required participant in the **Participants** section.
- Click **Save**.

This procedure changes the participant, sets the activity state to **Runnable** and informs the Java-based Server via an HTTP message that the workflow has changed and requires recalculation. If the message cannot be sent, a succeeding FullCheck will perform the recalculation.

The recalculation sends an e-mail to the new participant and the process continues with this approval step.

Changing Participants with Web Center

Whether you can change a participant in Web Center depends on the access policies in force. First, you must have the right to view the tasks of the person the workflow instance is related to (access right **showTasksOf**), and then you need the right to change the participant (access right **changeParticipant**) and finally, you need a set of users you can select from (access right **delegate**).

The context menu entry **Change Participant** in the Web Center allows you to change a participant. You can use the **Acquire** menu entry to request an approval step for yourself.

11.3.4.2. Restarting Workflows

You can restart workflows that are in states FAILED.ABORTED, FAILED.EXPIRED or FAILED.INCOMPLETE. Use the DirX Identity Manager to perform this task as follows:

- Select **Provisioning** → **Request Workflows** → **Monitor**.
- Locate the workflow to restart: either open the relevant folder or use the query "Failed Workflows".

- Select **RunJavaScript restartWF** from the context menu of the workflow instance.
- The workflow is set to state RUNNABLE and the workflow engine is informed about this change to handle it immediately.

The workflow is restarted.

11.3.4.2.1. Detailed Procedure

The RunJavaScript method calls a JavaScript procedure that performs a specific task. DirX Identity comes with a JavaScript that handles the states FAILED.ABORTED/EXPIRED/INCOMPLETE.

The script is located in the **Domain Configuration** view under the folder **JavaScripts** → **Request Workflows**.

The script performs these tasks in its prepareRestart method:

- the state is set to RUNNING
- the enddate / endtime is removed
- the expiration time is recalculated
- the request workflow engine is informed about the change

11.3.4.2.2. Custom JavaScripts

You can write your own JavaScripts procedures to handle other states. You can call these scripts via the **RunJavaScript** context menu entry.

You can also provide a corresponding action statement in the Workflow Instance object description to get your own menu entry.

12. Managing Target Systems

This chapter describes how to work with target system objects. Target system management consists of the following tasks:

- Adding target systems to the Identity Store
- Monitoring account and account-group assignments
- Handling the results of the validation workflow
- Managing target system accounts
- Managing target system groups
- Deleting target systems from the Identity Store

12.1. Managing Access Rights

DirX Identity supports several methods for access control in connected systems, as illustrated in the following figure:

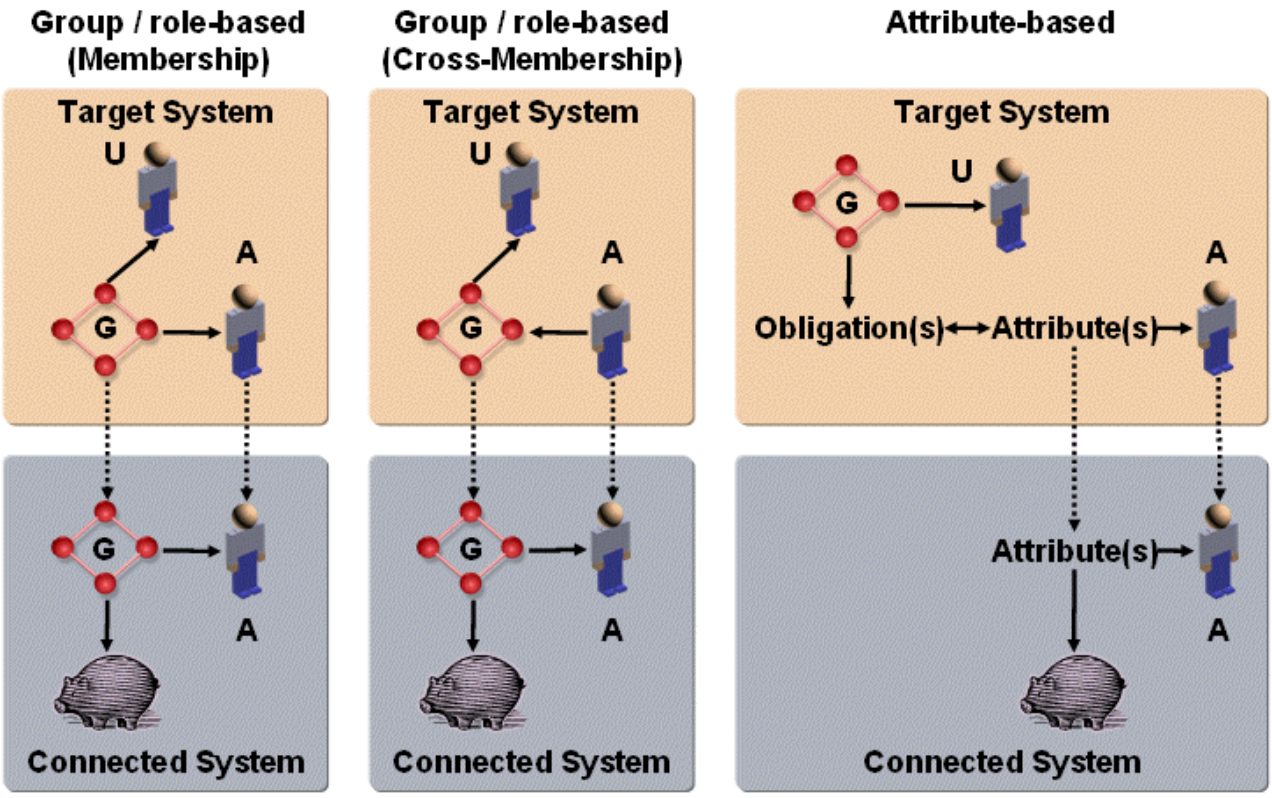


Figure 48. Supported Access Control Models

The next sections provide more information about the group-based membership and attribute-based membership access control models.For more information about the cross-membership model, see the section "Cross-Membership Scenario" in the *DirX Identity Application Development Guide*.

12.1.1. Managing Access Rights with Group-based Memberships

The preferred method is to control access rights via group memberships. Each group defines a set of access rights in the connected system regarding a set of resources. The meaning of a group is only visible from the group name. DirX Identity does not handle or store any discrete access control information.

Assigning a user a group either directly or via permissions and roles gives the user the right to handle the defined resource accordingly. Removing the user from the group revokes the access rights. DirX Identity supports group-side and account-side memberships. In group-side membership, the group holds the accounts as members. In account-side membership, the account holds the groups as members.

Fortunately, most connected systems today use the group-based access control.

You should try to work with account-side memberships as much as possible, because typically all users of a company can be members of a single (company) group. If a company has 500,000 users, this results in 500,000 group members. Handling such large groups is a problem for LDAP directories. On the other hand, the number of groups is typically much lower than the number of users in a large company, and there will be no user that is a member of all groups, which means that the user has full access to all resources in the company.

We recommend that you set up the target system in the Identity Store in account-side mode. If the connected system is not able to work in this mode, you can use cross-membership configuration for the Provisioning workflows to manage this problem. See the section "Cross-Membership Scenario" in the *DirX Identity Application Development Guide* for more information.

12.1.2. Managing Access Rights with Attribute-based Memberships

Some connected systems do not support the concept of group-based access control.In this case, access control is performed via attributes at the user’s account.To keep a unique method on the Identity Store side, we use group-based assignment of access rights as well.These groups possess obligations that set or reset the relevant account attributes.

The groups and the group memberships exist only in the Identity Store.Only the accounts with the access right-relevant attributes are synchronized between the target system and the connected system.Special validation rules can be established to create the relevant group memberships if the attributes are changed on the connected system side.

A prominent example of an attribute-based connected system is Microsoft Exchange.The attributes for the mailbox are held at the Active Directory Account.

For more information, read the sections about validation rules in the section "Managing Rules" in the chapter "Managing Policies".To understand obligations, read the section "Handling Attribute-based Access Rights" in the *DirX Identity Customization Guide*.

12.2. Working with Target System Objects

When you log into DirX Identity Manager (Provisioning) and select "Target Systems" from the view bar, DirX Identity Manager displays a hierarchical tree of the target systems whose access control information has been imported into the Identity Store (Provisioning Configuration).

Underneath a target system, the tree displays the following objects:

- An **Accounts** folder (optional), which contains the target system accounts (if the target system type requires the management of accounts).
- A **Groups** folder, which contains the target system's groups.
- Alternatively, a common subtree can exist that contains both **Accounts and Groups** (you can select this option only during creation of the target system).
- A **Configuration** folder, which can contain:
- A **JavaScripts** folder, which stores the Java scripts that are referenced from within the object descriptions.
- An **Object Descriptions** folder, which stores the object descriptions for the target system's accounts and groups.
- An **Obligations** folder that contains common On Assignment, On Revocation and On Validation rules that can be used by many group objects.
- A **Property Page Description** folder, which stores the property page descriptions for the target system's accounts and groups.
- A folder for storing **Proposal Lists**.
- A folder for storing specifications for the generation of **Reports** about the target system's accounts and groups.

The Accounts and Groups folders contain a default set of query folders, which allow you to search for Provisioning objects in an inconsistent state, such as objects that have a value in an **Error** or a **To do** field, or accounts with no user assignment.

The following figure shows an example of a target system displayed in DirX Identity Manager's Target Systems view (Provisioning view).

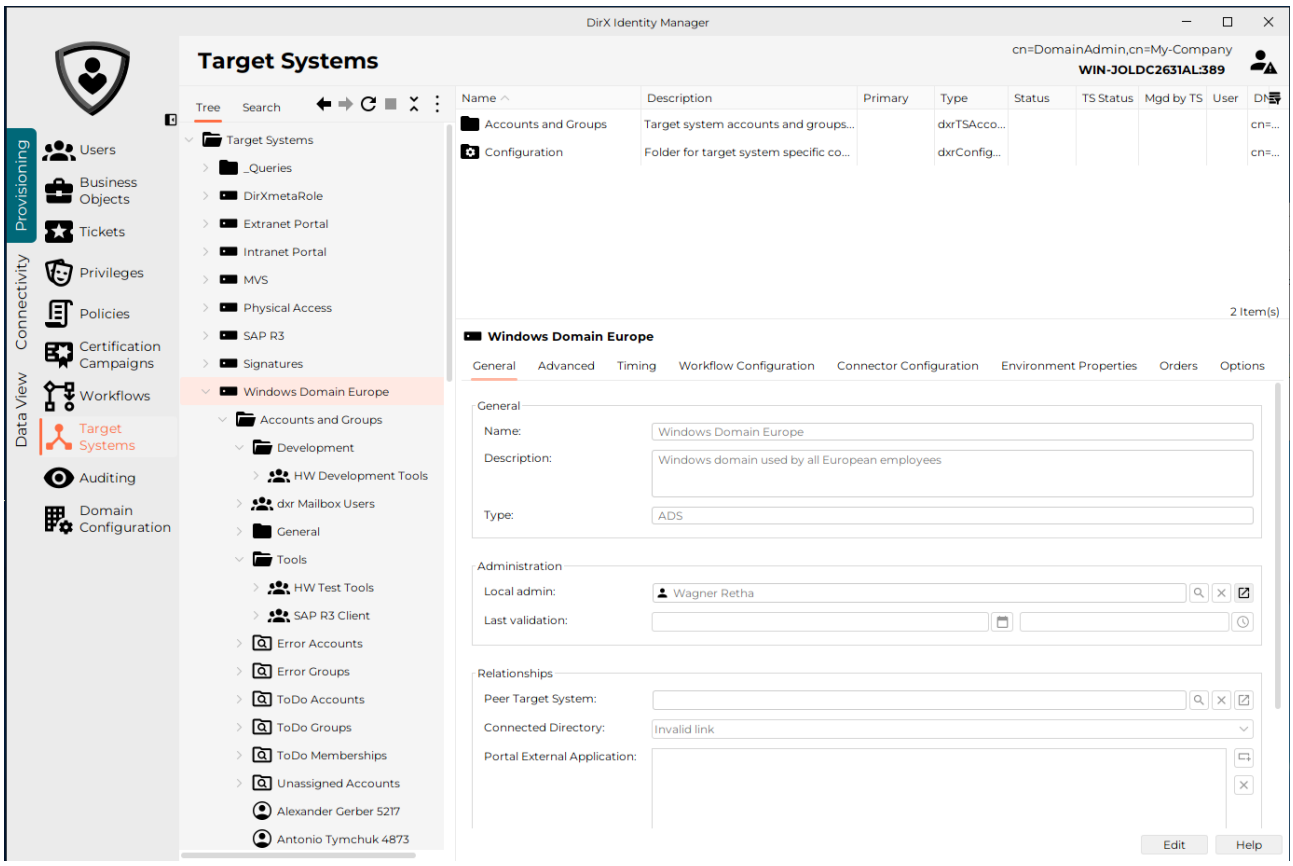


Figure 49. Managing Target Systems with DirX Identity Manager (Provisioning)

As illustrated in the figure, you view the properties of a target system object by clicking its entry in the tree.DirX Identity Provisioning also provides a search dialog that you can use to select and display a subset of these objects.When you select a target system object, DirX Identity Provisioning displays a property dialog for the object.The dialog typically consists of a set of tabs that you can use to view the object's properties.Click the tabs in the property dialog to move between the different property categories for the target system object.

To change the value or one or more of the properties displayed in a tab, click **Edit**.Manager keeps the changes you make to the object in its internal cache until you click **Save**.As long as you do not use **Save**, you can use the Manager's **Reset** function to restore the settings to their old values (the values that are stored in the Identity Store (Provisioning Configuration)).

When you click **Save** or change to another tab, Manager checks the modifications you have made for consistency.If you have clicked **Save**, Manager stores the changes in the Identity Store (Provisioning Configuration); if you have changed tabs, it stores the changes in its internal cache.Manager may return warnings and errors; the type and content depend upon the object you're changing and the type of changes you've made.If you receive warnings or errors, you can use **Cancel** to cancel your changes.You can also choose **Save**; in this case, DirX Identity Manager will store the appropriate error messages with the object.Manager then saves the changes in the Identity Store (Provisioning Configuration).In the event that it is unable to save the data because of network failure or an LDAP server error or timeout, Manager notifies you of the error.

To work with the related Connectivity objects, click the target system entry in the tree, open the context menu and select one of the options from the Connectivity submenu.

To access the Identity Store itself and the corresponding maintenance workflows, select the Connectivity submenu at the top-level node of the Target Systems view.

12.3. Adding a Target System

Adding a target system is a one-time set-up task that you must perform for each connected system that you want DirX Identity to manage.Use the target system wizard to perform this task.It helps you to create the target system object within the provisioning view as well as all necessary objects required for provisioning to the connected systems in the connectivity view.

Adding a target system consists of the following tasks:

1. Use the target system creation wizard to create a new target system including all necessary workflows in the Identity Store.
2. Customize the Provisioning account and group object descriptions for the new target system (with DirX Identity Manager (Provisioning)) - this task is only necessary if you want to use non-standard account or group properties or non-standard default values (such as name space rules or password generation rules) for generating accounts.
3. Adapt the mapping of the initial load / validation workflow.
4. Run the initial load workflow
5. Define rules to automate tasks
6. Manage the accounts that have no user assignment
7. Assign the imported groups to permissions or delegate this task to the role administrator.
8. Configure the synchronization and validation workflows and define a schedule to run these workflows regularly.

Variations:

To add a virtual target system, perform steps 1 and 2; the other steps are not necessary.

12.3.1. Creating the Target System

To create a target system with DirX Identity Manager (Provisioning):

1. Click the top-level Target System folder.
2. If you want to group your target systems into different containers, you can create a new container by selecting **New** → **Target System Container** from the context menu. Supply a name and a description for the new container and store it.
3. Select the container, where you want to create your new target system. Select **New** → **Target System** from the context menu. The target system wizard starts. It allows you creating a target system including all necessary workflows.
4. For cluster target systems, you need to build a fixed structure:
    - Under the top-level node (Target Systems), you can create a cluster workflow

container (for example, **My Unix systems**).

- Under this container, you can create any number of cluster target systems.



If the target system requires a target system-specific schema extension, be sure to extend the schema before you create the new target system.

If you have created a virtual target system, you may need to customize the target system's properties. These properties are available on the target system's Advanced and Options tabs. The *DirX Identity Customization Guide* provides further details on how to set up these properties to correspond to your virtual target system.

12.3.2. Extending Account and Group Object Descriptions

During the privilege resolution process, DirX Identity Provisioning automatically creates target system accounts without any administrator interaction. As a result, DirX Identity Provisioning requires that you provide it with the information it needs about how to create the target system account names and passwords and other required attributes so that it can perform this automatic function. You can:

- Define naming rules for account name and password creation: simpler ones that generate attribute values by combining other attributes, or more complex ones, by using Java scripts.
- Supply default values for attributes
- Add new attributes for accounts or groups that you want to synchronize with the connected system
- Use proposal lists to restrict the value range for attributes

To accommodate an individual connected system's requirements for account and group descriptions, DirX Identity Provisioning supplies template descriptions of account and group objects and allows you to customize them to a particular target system's account and group descriptions. These templates are located within each target system's Configuration folder, as follows:

- The Object Descriptions folder contains the attribute descriptions with the default values and naming rules
- The Property Page Descriptions folder contains the layout description of the tabs, where you can view and edit the properties
- The JavaScripts folder contains the Java scripts

To edit these descriptions with DirX Identity Manager (Provisioning):

1. Select the description in the subtree to display its properties.
2. Click the Object Description tab, and then click **Edit**.

Now you can use the object description editor to specify:

- The naming rules that DirX Identity Provisioning is to use when creating accounts and

groups in the target system

- The target system-specific account and/or group attributes and default values, if any

The *DirX Identity Customization Guide* describes the structure of these files and how to modify them.

Alternatively, you can use DirX Identity Manager (Provisioning) **Export text** function to write the description to a file, and then use your favorite XML editor to edit it within the file system. You then use DirX Identity Manager (Provisioning) **Import text** function to import the file back into the Identity Store (Provisioning Configuration).



During an upgrade installation of DirX Identity your versions of the account and group object and property page descriptions are preserved. See "Upgrading Customized Descriptions" for further details.



If you are adding target-system specific account and/or group attributes, you will also need to add these attribute types to the default LDAP directory schema supplied with DirX Identity Provisioning. You use the schema modification tools of the particular LDAP directory to accomplish these tasks. For example, if you are using the DirX LDAP directory, you use the DirX administration tools for modifying the schema. The *DirX Identity Customization Guide* provides more information about these tasks.

12.3.3. Configuring the Initial Load Workflow

The initial load workflow is a batch workflow that you use to load a snapshot of the connected system's accounts and groups into the Identity Store (Provisioning Configuration). DirX Identity Provisioning supplies an initial load workflow template (the validation workflow is used here) that is copied by the target system creation wizard. This design ensures that the workflow is also correctly configured.

If you have extended the account or group object descriptions, you may need to adapt the mapping of the workflow. Use the DirX Identity Manager (Connectivity) **Configure Workflow** wizard to adapt the mapping (and maybe other parameters) as follows:

- Adapt the attributes to be exported for accounts and groups (source selected attributes)
- Adapt the attributes to be imported for accounts and groups (target selected attributes)
- Create a corresponding mapping for accounts and groups

The *DirX Identity Connectivity Administration Guide* and the *DirX Identity Application Development Guide* provide more information about workflow configuration parameters and how to use DirX Identity Manager (Connectivity) to configure them.

12.3.4. Running the Initial Load Workflow

To run the initial load workflow, click the target system in the Target System view and then

select **Connectivity** → **Validation** → **Run Workflow** from the context menu. Alternatively, you can use the DirX Identity Manager's Connectivity view to run the initial load workflow by hand.

The workflow exports all accounts and groups and the relationships between them from the connected system and imports them into the target system subtree in the Identity Store (Provisioning Configuration).

The first execution of this workflow runs in mode "initial load". All subsequent executions run in mode "validation". The mode sequence is controlled by the **Last validation** field on the **General** tab. If it is empty, the workflow runs in "initial load" mode, otherwise it runs in "validation" mode.

To re-run the initial load workflow if it fails to complete due to a network or database failure, just clear the **Last validation** field before the run.

The *DirX Identity Connectivity Administration Guide* provides more information about how to use DirX Identity Manager (Connectivity) to run and monitor a workflow.



The difference between "initial load" and "validation" mode is only that the validation mode sets the ToDo fields at accounts and groups.

12.3.5. Defining Rules to Automate Tasks

To automate tasks in target system management, you may need to configure some consistency rules and validation rules. For information on how to check and automatically handle account-group assignments with validation rules, see the section "Checking Account-Group Assignments".

DirX Identity Provisioning provides the following consistency rule templates that you can copy and customize:

assocAccount2User or assocAccount2UserMV

These rules search for unassigned accounts; that is, accounts without a user assigned. For each account, it searches for a matching DirX Identity user. To match accounts to users, it performs a best-guess match using the value configured for the input parameter *join filter*. If the search returns exactly one entry, it makes the user-account assignment. The MV rule works with multi-value attributes.



You can use event-based processing workflows for accounts. These workflows have a configurable feature to associate accounts to users. This is especially useful when new accounts are added during a validation workflow. Then the accounts are immediately bound to users.

CreatePersonasForNonPrimaryAccounts

This rule creates personas for all non-primary accounts that are assigned to a user. First, it checks whether a user already has a primary account assigned in the target system. Next, it creates personas for all non-primary accounts of the user and links the personas to the user. Read the section "Full Provisioning" in the section "Managing Personal

Accounts" to understand the use case. See the section "Customizing the Consistency Rule for Automatic Persona Creation" in the chapter "Customizing Program Logic" in the *DirX Identity Customization Guide* for details about customizing the rule.

CreateFunctionalUsersForUnassignedAccounts

This rule creates functional users for all accounts that are not assigned to a user and then links the accounts to the newly created functional users. Read the section "Full Provisioning" in the section "Managing Personal Accounts" to understand the use case. A description and hints for customization are given in the section "Customizing the Consistency Rule for Automatic Functional User Creation" in the chapter "Customizing Program Logic" in the *DirX Identity Customization Guide*.

CreateUsersForUnassignedAccounts

This rule creates users for all accounts that are not assigned to a user and then links the accounts to the newly created users. Read the section "Full Provisioning" in the section "Managing Personal Accounts" to understand the use case. A description and hints for customization are given in the section "Customizing the Consistency Rule for Automatic User Creation" in the chapter "Customizing Program Logic" in the *DirX Identity Customization Guide*.

Win2k-Mail-address-to-users

This rule copies all mail addresses of the Windows accounts matching the rule filter to their assigned users.

Ex55-Mail-address-to-users

This rule copies all mail addresses of the Exchange 5.5 mailboxes matching the rule filter to their assigned users.

cleanse-AD-mail-address

This rule deletes all mail addresses of user subjects that are not in their Windows accounts.

cleanse-NT-mail-address

This rule deletes all mail addresses of user subjects that are not in the NT 4.0 or Exchange 5.5 target systems.

12.3.6. Configuring the Synchronization and Validation Workflows

Read the *DirX Identity Application Development Guide* for information about synchronization and validation workflows and how to set them up and configure them.

12.4. Monitoring Account and Account-Group Assignments

When it imports a connected system’s accounts, groups, and account-group relationships into the Identity Store (Provisioning Configuration), the target system validation workflow in "initial load" mode assigns newly imported accounts and newly imported account-group assignments to the IMPORTED state.This state indicates to DirX Identity Provisioning that it

should not automatically delete the account-group assignment and the corresponding account if they are not granted through a user-privilege assignment.The state exists to give the role administrator time to assign the newly imported groups for example to permissions in the privilege structure.

After the initial target system load, the user-privilege assignment and resolution process causes DirX Identity Provisioning to change these states as follows:

- When user-privilege assignment and privilege resolution results in the assignment of the account to a user, DirX Identity Provisioning changes the account state to ENABLED.When the last user-privilege assignment is later revoked, DirX Identity Provisioning disables the account.
- When user-privilege assignment and privilege resolution results in the account-group assignment, DirX Identity Provisioning changes the account-group assignment state to ENABLED and thus to a normal assignment.When the user-privilege assignment is later revoked, DirX Identity Provisioning removes the account-group assignment.

If accounts and/or account-group assignments remain in the IMPORTED state for some time after a target system has been added to DirX Identity Provisioning, it means that their assignment to one or more Provisioning users was made locally at the connected system and outside the DirX Identity Provisioning user-privilege assignment process.You should periodically monitor the state of recently imported target system accounts and account-group assignments to make sure that, over time, they move to being controlled by the DirX Identity Provisioning access management functions.

The role administrator or the connected system administrator can choose to accept an account-group assignment that has been made outside the user-privilege assignment process by using Manager or a rule to change the assignment state to IGNORE (see the topic "Checking Account-Group Assignments" for details).If the DirX Identity Provisioning user-privilege assignment process subsequently grants the assignment, DirX Identity Provisioning changes its state to ENABLED.

You can create a query folder underneath each target system (and also in the Users view) that you can use to search for and identify assignments in the IMPORTED state.When you open the folder, it performs a search on the Identity Store (Provisioning Configuration) according to the values in its Filter tab and displays the results as objects underneath the query folder.

12.5. Handling the Results of the Validation Workflow

When the target system validation workflow runs, it writes messages about the deviations it finds between the local connected system and the Identity Store (Provisioning Configuration) into the **To do** fields of the affected target system objects in the DirX Identity store (Provisioning Configuration).The **To do** fields of the target system objects will contain messages when the following actions have been carried out in the target system without DirX Identity Provisioning triggering them:

- An account has been created or deleted

- A group has been created or deleted
- A group-account membership has been created or deleted

DirX Identity Provisioning provides a To do query folder underneath each target system that you can use to search for and identify objects with values in their **To do** fields.When you open the folder, it performs a search on the Identity Store (Provisioning Configuration) according to the values in its Filter tab and displays the results as objects underneath the To do folder.You should periodically use the target system's To do query folder to search for objects with values in their To do fields, and follow up with the appropriate actions.

The validation workflow identifies accounts and groups created in the connected system by the **To do** field value **Created in target system** and the IMPORTED state.You must decide whether to delete these accounts and groups or integrate them into the Identity Store (Provisioning Configuration).

The validation workflow identifies accounts and groups deleted in the connected system by the **To do** field value **Deleted in target system** and the DELETED state. If you determine that an account that has been deleted in the connected system is no longer needed, use DirX Identity Manager (Provisioning) to delete it from the Identity Store (Provisioning Configuration). If an associated user needs the account, DirX Identity Provisioning automatically creates a new one with the next privilege resolution.

If new groups have been created autonomously in the local connected system, you must perform the same actions for these groups as you did for the groups that were imported during the initial load procedure:

- Mark the groups that do not need permission assignments as "target system local" by selecting them for editing and then checking their **Manage only in target system** fields
- Delete any groups that do not match the privilege structure; see the topic "Deleting a Group" for further details
- Assign permission parameters and values to the groups; see the topic "Assigning Role Parameters" for further details
- Identify the new groups to the privilege structure administrator for permission assignment

DirX Identity Provisioning automatically re-creates deleted group-account memberships with the next privilege resolution.



You can use rules to automate the tasks described here.

12.6. Managing Target System Accounts

DirX Identity distinguishes between two types of accounts:

- Personal accounts, which belong to exactly one person.
- **Privileged accounts**, which exist for administrative tasks and can be used by more than one user.Privileged accounts are typically created for tasks that require very high access rights; for example, the root account in UNIX operating systems and the Administrator

account in Windows systems.

Both types of accounts can be used in a single domain.

The next sections describe how to manage personal and privileged accounts and how to use **functional accounts** to manage multiple personal accounts

12.6.1. Managing Personal Accounts

The connected system administrator generally does not need to create, delete or modify personal connected system accounts because DirX Identity Provisioning automatically handles these tasks. DirX Identity Provisioning creates an account when a user is granted his first access right in the target system and disables the account when the last right has been revoked.

However, the synchronization and validation of the remote connected system can reveal differences in the target system object states in the Identity Store (Provisioning Configuration). The validation workflow detects accounts, groups, and account-group memberships that have been created or deleted in the connected system independently of a DirX Identity Provisioning request. The workflow notifies you about these objects by writing messages into their **To do** fields. As a result, you should periodically use the **To do** query folder to search for accounts (and other objects) in an inconsistent state, and follow up with the appropriate actions, as described in the topic "Handling The Results of the Validation Workflow".

The topic "Handling Imported Accounts" describes the account management functions that you need to perform when you add a target system to the Identity Store (Provisioning Configuration).

Managing target system accounts comprises these concepts:

- Adding a new personal account
- Changing the attributes of a personal account
- Handling imported accounts
- Handling multiple accounts
- Automatic account creation
- Smooth account creation
- Managing accounts only in the connected system
- Deleting a personal account

In general, you will only need to perform these tasks as the result of some unusual condition; these tasks are not typical day-to-day maintenance tasks.

12.6.1.1. Adding a Personal Account

In general, you do not need to create new target system accounts, because they are automatically created as part of the privilege resolution process. However, you may need to add an account by hand if the privilege resolution function was unable to create it due to

naming rule violations or other problems.

To add a new target system account:

- In the Target Systems view, right-click the Accounts folder in the target system's subtree.
- Select **New** → **Account**. Enter a name (common name (cn)) for the account.

Manager returns an error if the name you supply for the account is not unique.

For hierarchical target systems, you can structure your accounts with containers:

- Select **New** → **Container** to create a new container.
- Use the **Move Object** function to move accounts to containers.

12.6.1.2. Changing the Attributes of an Account

In some cases, you may need to make changes to a target system account. For example, you may need to:

- Mark the account as "Managed only in target system" because it does not need a user assignment.
- Make the account a primary account, if the user has more than one account in the connected system. Only one account can be the primary account and DirX Identity Provisioning does not check for this constraint. For a user-to-privilege assignment, only the primary account is used for the group membership.
- Clear the **To do** field.

All of these properties are available as attributes of the target system account object. To change an account's attributes:

1. Select the account in the target system subtree to display its properties.
2. Click the tab that contains the attribute you want to change, and then click **Edit**.
3. Check the **Primary account** field as necessary.
4. Check the **Manage only in target system** field as necessary.
5. Select whether the account is to be managed as **personal** or **privileged** account.
6. Click **Save** to save your changes.

12.6.1.3. Handling Imported Accounts

After initial load or after a validation, you will encounter accounts in the state IMPORTED. These accounts are not under DirX Identity's control. The following figure illustrates alternative methods for handling these accounts.

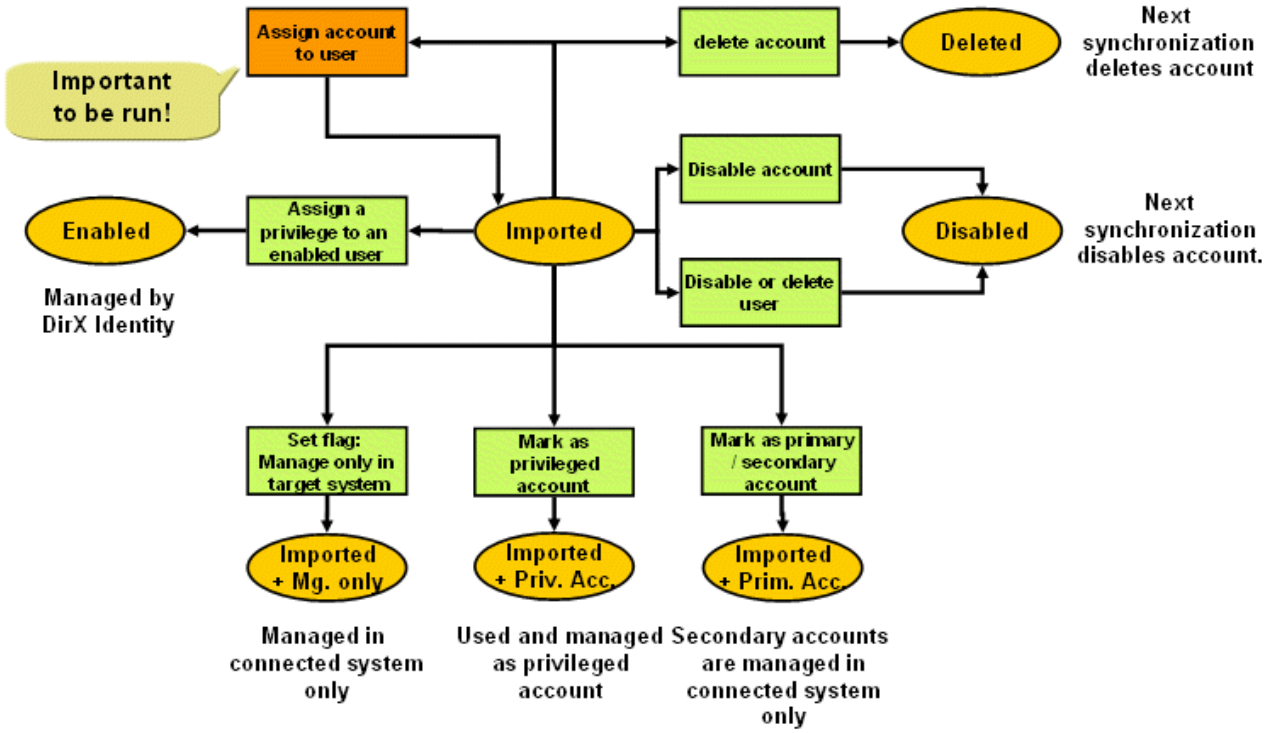


Figure 50. Alternatives for Handling Imported Accounts

12.6.1.3.1. Association of Accounts to Users

First, you should try to associate the imported account with an existing user. You can use either the consistency rule **assocAccount2User** or the **EventBasedAccountProcessing** workflow to perform this task automatically. Depending on your join criteria, most imported accounts will be assigned to a user.

However, you must handle the accounts that the rule could not assign because it could not find a matching user or because it found multiple matches. Use the query folder **unassigned** below the **Accounts** or **Accounts and Groups** folder to display all remaining unassigned accounts.

For these accounts, try to perform this procedure:

1. Select the account from the subtree to display its properties, then click **Edit**.
2. Right-click the ellipsis button to the right of the **User** field. Manager displays a search dialog.
3. Use the search dialog to search for a matching Provisioning user. If you find a matching user, select the user to assign it to the account.
4. If you cannot find an existing user, you can decide to create a new one and then assign this account to the user.

12.6.1.3.2. Handling Assigned Accounts

For accounts that you can successfully assign to a user and that are still in the state IMPORTED, you now have these options:

- **Assign privileges**

You can assign a privilege to the user. The state of the account will change to ENABLED. If a membership of a group already exists, this membership also changes its state from IMPORTED to ENABLED.

- **Delete or disable the user**
  You can delete or disable the user. The account's state will be set to DISABLED.

  By default, DirX Identity Provisioning sets an end date for the account's lifetime when it changes to the DISABLED state. When this date is reached, DirX Identity Provisioning automatically changes the account's state to DELETED. When the target system synchronization workflow runs, it deletes the account from the connected system; if this operation is successful, it deletes the account from the Identity Store (Provisioning Configuration).
- **Set the Primary Account flag**
  If there are several accounts from the same target system assigned to the user, you must flag one of them as the primary account, or the privilege resolution will fail. For more information about this concept, read the section about handling multiple accounts and the description of partial provisioning.

For all accounts that you still cannot assign to a user, you must select one of the options described in the next section.

12.6.1.3.3. Handling Unassigned Accounts

You need to decide whether to:

- **Delete the account**
  If you know that an unassigned imported account is of no use to the DirX Identity Provisioning system (or perhaps should not be available within it) you can delete it from the Identity Store (Provisioning Configuration). See the topic "Deleting an Account" for information on how to delete it with DirX Identity Manager (Provisioning). These accounts will be deleted from the connected system during the next synchronization (it was a lost account).
- **Disable the account**
  If you are unsure about the purpose of an imported account, you can disable it. To disable an account, right-click it in the subtree, then click **Disable**.



This option is only available for unassigned accounts! DirX Identity Provisioning changes the account's state to DISABLED. The target system synchronization workflow will attempt to disable the account in the connected system the next time it runs. If this operation is successful, the workflow changes the account's state in the target system to DISABLED in the Identity Store (Provisioning Configuration). You can use DirX Identity Manager (Provisioning) to enable the account later on, if necessary.



By default, DirX Identity Provisioning sets an end date for the account's lifetime when it changes to the DISABLED state. When this date is reached, DirX Identity Provisioning automatically changes the account's

state to DELETED. When the target system synchronization workflow runs, it deletes the account from the connected system; if this operation is successful, it deletes the account from the Identity Store (Provisioning Configuration).

- **Handle the account as a multiple account** for a user; see the section "Handling Multiple Accounts" for instructions.
- **Handle the account as a privileged account**; see the chapter "Managing Privileged Accounts" for instructions.
- **Manage the account only in the connected system.**
  If you want to keep an imported account in DirX Identity Provisioning, but you do not want to assign it to a user and you do not handle it as a privileged or multiple account, you can check its **Manage only in target system** field. Checking this field indicates that the account will be managed locally in the connected system, and not by DirX Identity Provisioning. See the topic "Changing the Attributes of an Account" for information on how to make this modification with DirX Identity Manager (Provisioning). See the section Manage Only in Connected System for more information about the concept.

12.6.1.4. Handling Multiple Accounts

DirX Identity provides several mechanisms to handle multiple accounts per target system. To understand this feature, you first need to understand why users can have multiple accounts in target systems. Typical reasons are:

- Users want a clear separation of access rights when they are logged in. A user might have a personal account where he handles all personal tasks, for example writing and reviewing of documents, browsing the Intranet and Internet and so on. This user might also be an administrator in one or more target systems with very high access privileges that allow him to handle important resources. The user prefers to use the privileged account only for his administration tasks to minimize the danger of making unintentional errors.
- Compliance requirements force companies more and more to monitor sensitive actions in IT systems. If users work with different accounts, the company can focus on auditing the most important accounts in more depth (for example, privileged and functional accounts).
- If personal accounts (which really represent the user) and functional accounts (which represent a specific task carried out for some time by a specific user) are clearly separated, it is easy to switch a functional account from one user to another if tasks change.

These requirements show the benefits of having different accounts for the same user. Initial load of target systems shows that these methods are widely used. The requirements also show that different accounts exist because they represent different access rights in the connected systems. In this case, a user has an identity that can exist as several personas, as shown in the following figure.

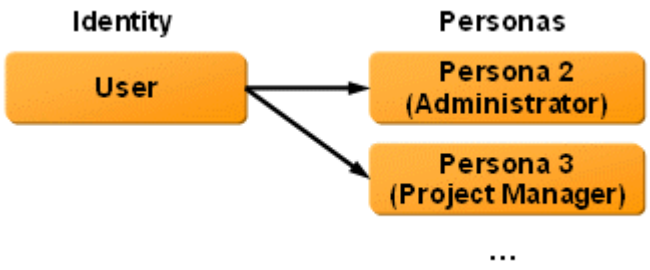


Figure 51. One Identity User, Multiple Personas

As shown in the figure, the user needs a personal representation (the primary representation that is the identity itself). He can also be represented in other facets; in this example, as an administrator or a project manager. Personas are typically represented as additional accounts in target systems.

DirX Identity allows for modeling personas with the full set of features like privileges and access policies, referred to as "full provisioning". We recommend using this method if multiple accounts are frequently used. If only a few multiple accounts exist, you can use a simpler approach, called "partial provisioning". The next sections describe these methods in more detail.

12.6.1.4.1. Full Provisioning

The "full provisioning" method models each persona as an additional persona entry, as shown in the following figure:

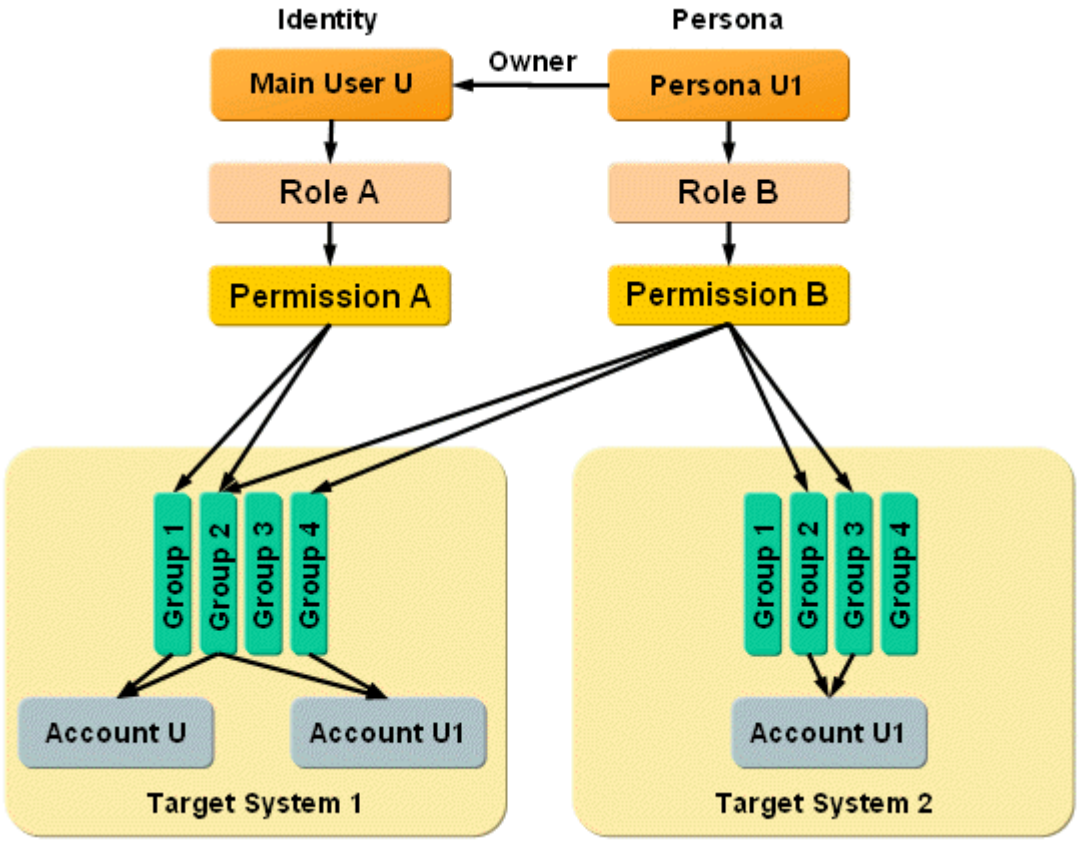


Figure 52. Full Provisioning of Multiple Personas

As illustrated in the figure:

- User U represents the primary identity. Perhaps this entry is fed by an HR system. Assignment of role A to this entry results in two group memberships and the corresponding account U.
- A persona entry (Persona U1) is created. It represents for example administrative tasks in one or more target systems. The assigned role B results in two group memberships in target system 1 and the corresponding account U1 as well as two group memberships in target system 2 and the corresponding account U1.

This example shows that you can handle the privileges of the identity and the personas completely independently. To indicate the affiliation, you link the persona U1 to user U via the owner attribute.

You can run a consistency rule on a regular basis to check that the owner links are still valid (that is, they point to enabled users). If an owner link is no longer valid, the persona is marked to be erroneous. You can use the standard query folders to assign valid identities again.

You can run other consistency rules regularly to copy important attributes from the identity to all linked personas, if this is a required feature.

To handle all of these operations correctly, it makes sense to work with personas and functional users. Consistency rules can then distinguish easily between identities and personas.

We have just described how the system operates. Another issue is how to set it up when connected systems already exist. Perform these steps for target system setup:

- Use the target system wizard to create the new target system in DirX identity.
- Run the initial load workflow to import all relevant accounts and groups.
- Run the consistency rule **assocAccount2User** to link as many accounts as possible to user entries.
- Build a consistency rule that checks for multiple assignments. Run it and check all multiple assignments. Decide which account is the identity account and mark it as the primary account.
- For all other assignments, create persona entries and link the accounts to these new entries. Set the owner link to the corresponding identity. For this task, you can use the consistency rule **CreatePersonasForNonPrimaryAccounts**, which is delivered as a template and must be configured to your needs. See the section "Automatic Persona Creation" in the chapter "Customizing Program Logic" in the *DirX Identity Customization Guide* for details.
- Check for unassigned accounts. Decide whether these represent identities, personas or functional users. Create the necessary objects and link the accounts to them. For this task, you can use the consistency rules **CreateFunctionalUsersForUnassignedAccounts** or **CreateUsersForUnassignedAccounts**, which are delivered as templates and must be configured to your needs. See the section "Customizing the Consistency Rule for Automatic User Creation" or "Customizing the Consistency Rule for Automatic

Functional User Creation" in the chapter "Customizing Program Logic" in the *DirX Identity Customization Guide*.

Now you should have a target system that is completely controlled by DirX Identity.

12.6.1.4.2. Partial Provisioning

If only a few multiple accounts exist (for example, some administrator accounts), you can either use the **Manage only in target system** flag, the **Primary Account** flag or the **privileged account feature** to model multiple accounts. The following figure illustrates this method:

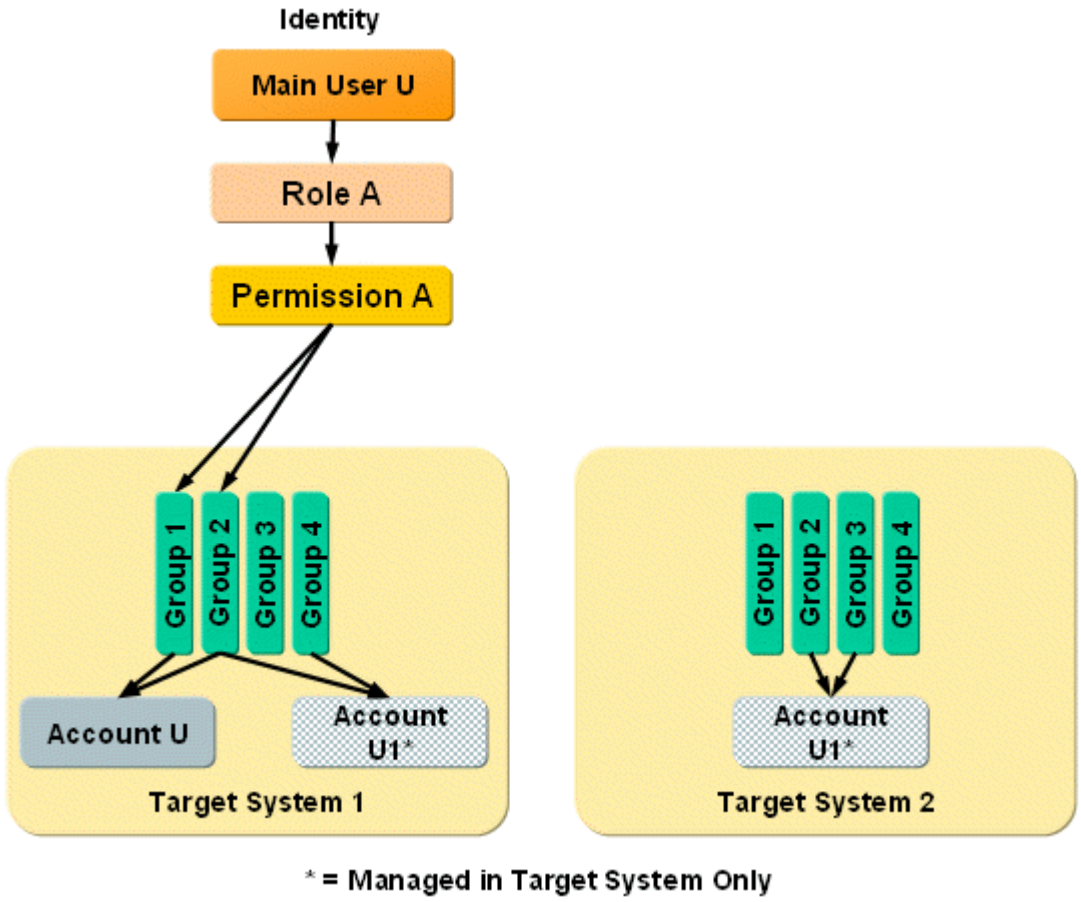


Figure 53. Full Provisioning of Multiple Personas

The figure shows an example with three target system accounts. If you use the **Manage only in target system** feature, only the identity (user U) is controlled by DirX Identity (Account U). The other two accounts and their group memberships are controlled only by the connected system administrators.

A disadvantage of using the **Manage only in target system** feature is that you cannot see these accounts at the user entry. If you use the Primary Account flag instead, you can see all accounts at the user. In our example, you flag Account U as the **Primary Account** and Account U1 as the secondary account (which means that the Primary Account flag is clear). In this case, all accounts are linked to the user and thus visible but only Account U is used by the privilege resolution process for membership control to groups. Account U1 is not touched at all.

In Target System 2, you can only use this method if you disable smooth account creation. Then you can reset the Primary Account flag for Account U1 so that this account is not used by the privilege resolution process. Instead, a new account is created if the naming rules allow it. If the privilege resolution process tries to create an account with the same name, an error message occurs. Thus it might be better to use the **Manage only in target system** flag for this target system.

12.6.1.5. Automatic Account Creation

This section describes how the privilege resolution process handles the creation or the re-use of single or multiple accounts. This is especially important if a user has multiple accounts in a target system because the privilege resolution process needs to know which account to use for subsequent automated group assignments. These algorithms work per target system.

Algorithm for one account:

- If an account is assigned to a user and it is the only one, this account is taken and the **Primary Account** flag is set if not yet set.
- If the privilege resolution process needs an account. **smooth account creation** at the domain is enabled and the account fits for the privilege resolution (it has the calculated name), the account is linked to the user and the **Primary Account** flag is set.
- If **Smooth Account Creation** is not enabled, a new account is created. If an account with the same name already exists, an error is indicated and privilege resolution stops.

Algorithm for multiple accounts:

- If none of the existing accounts that are linked to this user is flagged as the **Primary Account**, a warning is issued and privilege resolution stops.
- If more than one account is flagged as the **Primary Account**, an error is issued and privilege resolution stops.



The **Primary Account** flag is reset if a user link is removed either manually or by an automatic process that uses the DirX Identity Services layer.

12.6.1.6. Smooth Account Creation

Assigning a privilege to a user will result in account creation if an account in the respective target system is not yet created. Depending on the algorithm defined for the account, name errors can occur. If an account with the same name already exists, and the algorithm is not able to calculate an alternative name, this is reported and account creation is aborted.

You can switch on smooth account creation to avoid this situation. Check the corresponding flag **Smooth account creation** at the domain object.

If the flag is set and an account is to be created, the software performs this procedure:

- It checks whether an account with the same name already exists where the **Manage**

only in target system flag is not set.

- If no, it creates the account again.
- If yes, it checks whether this account is linked to this user or whether the user link is empty.
  - If yes: it takes this existing account and does not create a new one. If the account is already in dxrState DELETED it enables the account. The attribute dxrState is set to DISABLED and the attribute dxrTSState is set to NONE.
  - If no: if the calculation of alternative names is configured it tries with these new names. If only a fixed name is available an error message is generated.

This feature allows maximum use of existing accounts, but be aware that there are situations where an incorrect account can be selected. If you perform an initial load from a connected system, all available accounts are imported. If you do not run the procedure to associate these accounts to users or if the procedure cannot assign all accounts to users, and you assign privileges to users, an account may be used that is not the account of this user. This situation occurs if the account name algorithm calculates an account name that is equal to an existing account name. In this case, the account is assigned to a user to which it does not belong.

12.6.1.7. Managing Accounts Only in the Connected System

You can use the flag **Manage only in target system** to indicate that an account should be managed completely in the connected system and not by DirX Identity. When you set this option, the account is only visible in the target system in the Identity Store and requires regular updating by the validation workflow.

The flag is evaluated by all automatic associations of accounts to users. It indicates to these processes that this account should not be linked to a user entry. Processes that operate this way include:

- The **EventBasedAccountProcessing** workflow
- The **AssocAccount2User** rules
- Smooth account creation

The flag is reset only if you:

- Set the user link directly either manually or through any automatic process that uses the DirX Identity Services layer.
- Reset the flag either manually or through any automatic process.

12.6.1.8. Deleting an Account

You may need to delete a target system account by hand if, for example, a user has several accounts on the target system, but really only needs one of them. You can only delete secondary (non-primary) accounts. DirX Identity Provisioning automatically handles the deletion of group memberships.

To delete an account, right-click it in the subtree, then click **Delete**. DirX Identity

Provisioning changes the account's state to DELETED. The target synchronization workflow will attempt to delete the account from the connected system the next time it runs. If the operation is successful, it will also delete it from the Identity Store (Provisioning Configuration).

12.6.2. Managing Privileged Accounts

Privileged accounts typically come with the initial installation of a target system. Accounts like "root" for a UNIX system or "Administrator" for a Windows system already exist and are not bound to a specific user entry.

Any user can use a privileged account as long as he can authenticate with this account: he must know the password of this account or his certificate must be assigned to this account (this method is only supported by some target systems).

DirX Identity's role-based management allows for managing password access to these privileged accounts. The following figure illustrates this concept:

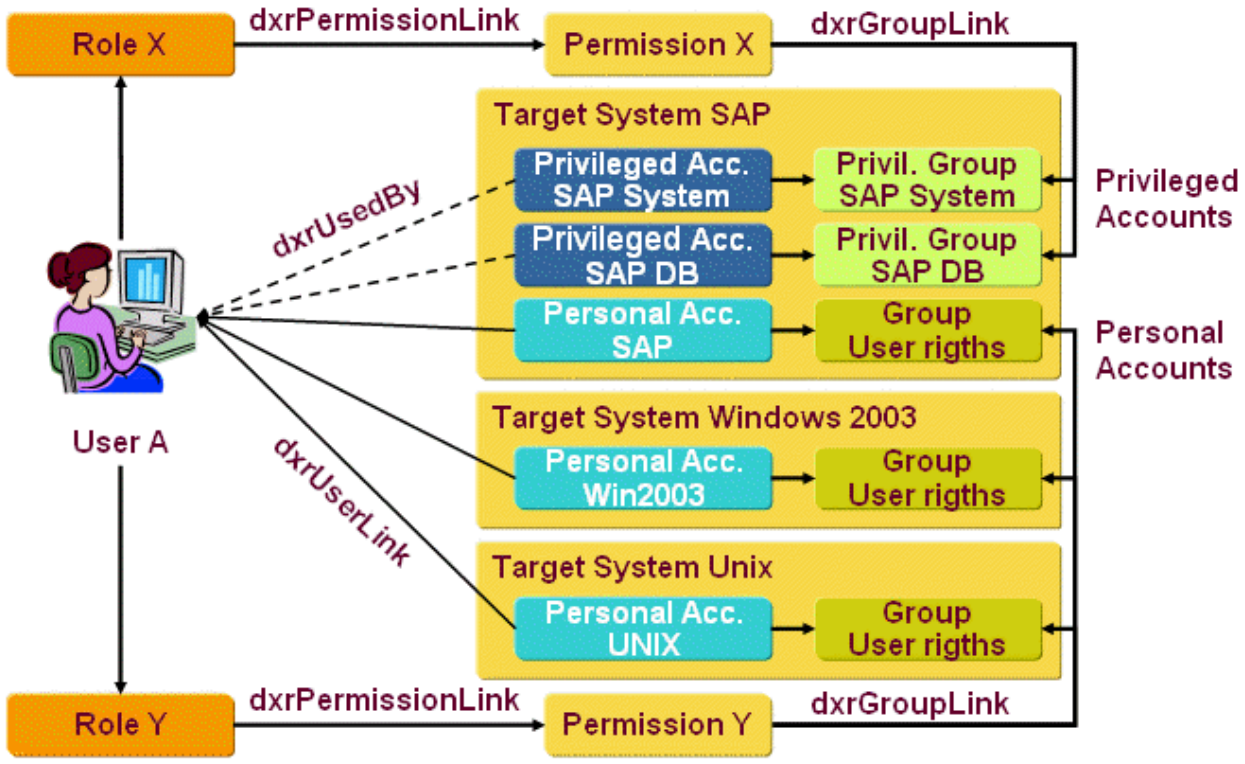


Figure 54. Personal and Privileged Accounts

The figure shows that assignment of a role (Role Y) assigns via permissions (Permission Y) one or more groups in different target systems to a user (Group user rights). The necessary accounts (Personal Account ...) are created automatically and assigned to these groups. The accounts are linked via the dxrUserLink attribute to the user entries.

If Role X is assigned to the user, then via Permission X a set of privileged groups may be assigned. It does not imply the creation of personal accounts; only a link is set from the corresponding privileged accounts to the user entries (attribute dxrUsedby).

As a result, the user can now read the passwords of the assigned privileged accounts in

clear text. The system can also copy the certificates of the user to this account. Provisioning of the password and the certificates to the connected system allows accessing the target system with these credentials.

If the role assignment is removed, the dxrUsedBy attribute is deleted. Because the user could still remember the old password, the system automatically changes the password. This action guarantees that only users who are still assigned to the privileged account can read the password. The system also removes all certificates of the user from the account, which disables access to the connected system.

To handle other critical situations, DirX Identity comes with a set of account password management services. See the *DirX Identity Application Development Guide* for more information.

The next sections describe the management of privileged accounts in more detail.

12.6.2.1. Enabling Privileged Account Management

For performance reasons, privileged account management is disabled by default. To enable it, set the switch **Enable Privileged Accounts** at the domain object.

12.6.2.2. Marking a Privileged Account

After initial load of your target system content, or later on if a privileged account was added to the target system and a validation retrieved it from the target system, follow this procedure for each privileged account:

- Set the **Account Type** flag at each of these accounts to **Privileged**. This action guarantees that the privilege resolution service handles this account as a privileged account (it uses the dxrUsedBy attribute).
- Set up additional password policies for this target system with a correct **Maximum Age** parameter. Be sure that this time is shorter as the required connected system setting. Set the **Character Check** parameters, too.
- Assign one of these password policies to each privileged account (**Authentication** tab).
- Check the assigned groups of this account (view the **Member of** tab of the account).
- If groups are already assigned (they came from the target system via initial load or validation), set the **Group for Priv. Account** flag for each of these groups to indicate to the privilege resolution service that this group handles only privileged accounts.
- If there are no groups assigned, create a new group for each privileged account. Set the **Group for Priv. Account** flag and assign the privileged account in the Privileged Members tab. If this group is subsequently assigned to a user, the dxrUsedBy links are set for all assigned privileged members (that are the privileged accounts).



These groups are not necessary for the target system handling, so it doesn't matter whether they are synchronized to the target system.

- For each account that you do not want to manage from DirX Identity, do not create a group for the account. Instead, mark the account as Privileged to indicate that it cannot

be used as a personal account.

- Assign the existing or newly created groups to permissions and maybe to roles if you want to handle the privileged accounts via permission or role management.

If there are any criteria that let you recognize all privileged accounts, you can write a consistency rule that automatically performs the procedure described here.

Now you can assign privileged accounts to users via role → permission → group assignment.

12.6.2.3. Managing Security Tokens

DirX Identity provides two authentication mechanisms for privileged accounts:

- **Passwords** - use passwords to authenticate in the target system with a privileged account.
- **Certificate**s - use certificates for your privileged accounts.

For each target system, set up one of these mechanisms. You can also set up a mixed solution.

12.6.2.3.1. Password Management for Privileged Accounts

If you manage access to your connected system via passwords and you have set up the system according to the instructions in "Marking a Privileged Account", you can use the following features:

- The default access policy **Users handle passwords of their accounts** defines that users can read and set privileged account passwords if the corresponding group assignment has set the **dxrUsedBy** attribute (check the policy definition).
- Setting the passwords at these privileged accounts requires setup of the Java-based **AccountPasswordManager** workflow. This workflow receives the JMS event (for example, from Web Center) and sets the password at the account.
- The **AccountPasswordManager** workflow also creates a Java-based Server internal event to trigger the corresponding target system workflow. Set up the required **SetPassword** target system workflows to guarantee correct propagation to the connected systems.
- Set up a schedule for the **AccountPasswordManager** that checks at regular intervals if passwords are about to expire. If this is the case, the account password manager creates a new password according to the defined password policy and sets it at the account within the target system. The **SetPassword** workflow propagates the changed password to the connected system.
- If a connected system must be restored for some reason - for example, because the passwords for the privileged accounts might be out of sync after a severe error - use the **ResetAccountPasswords** workflow to generate new passwords for all privileged passwords of the connected system. The **SetPassword** workflow propagates the changed password to the connected system.

12.6.2.3.2. Certificate Management for Privileged Accounts

If the connected system provides authentication via certificates, set up these features:

- Set up the master mechanism to transfer the certificate(s) of a user to a privileged account.For more information, see the section "Using the Master Attribute in a Property Element" in the *DirX Identity Customization Guide*.
- Set up the Java-based provisioning workflows for synchronization of the certificate attribute (adapt the mapping accordingly).

12.6.3. Managing Functional Accounts

Functional accounts can also come with the initial target system.Examples of functional accounts include a trainee account, a meeting room account, or a group mailbox account.

You can choose to handle a functional account as a privileged account or as a personal account of a functional user.If you have a scenario similar to the one described in "Managing Privileged Accounts", where multiple administrators use the same account to log on to a system, it is convenient to model this situation with a privileged account.If you have several different functional accounts, it makes sense to create a functional user for each functional account.You then use the sponsor link to associate the functional user with its responsible user.The account is completely managed by the DirX Identity privilege resolution process and is assigned to the functional user as his personal account.For details on creating a functional user, see the chapter "Managing Functional Users".

12.7. Managing Target System Groups

You can manage groups in the Identity Store.Synchronization and validation with the local connected system can reveal differences between the target system content and the Identity Store.The validation workflow detects accounts, groups, and account-group memberships that have been created or deleted in the local connected system independent from a DirX Identity Provisioning request.The workflow notifies you about these objects by writing messages into their **To do** fields.As a result, you should periodically use the To do query folder to search for groups (and other objects) in an inconsistent state, and follow up with the appropriate actions, as described in the topic "Handling The Results of the Validation Workflow".

Managing target system groups consists of these tasks:

- Adding a new target system group
- Changing the attributes of a group
- Managing group hierarchies
- Managing imported groups
- Managing group memberships
- Deleting a group

12.7.1. Adding a Target System Group

You can import groups and group from target structures via the validation (optionally in initial load mode) workflows or create groups by hand within the DirX Identity Manager.

To add a new target system group:

- In the Target Systems view, right-click the **Groups**, **Accounts and Groups** or any of its sub folders in the target system's subtree.
- Select **New** → **Group**. Supply a name (common name (cn)) for the group.

Manager returns an error if the name you supply for the account is not unique.

For hierarchical target systems, you can structure your groups with containers:

- Select **New** → **Container** from the context menu to create a new container.
- Use the **Move Object** function to move accounts to containers.

12.7.2. Changing the Attributes of a Group

In some cases, you may need to make changes to a target system group. For example, you may need to:

- Change group attributes
- Change one or more of the permission parameters to influence assignment of this group to users
- Mark the group as **Manage only in target system** because DirX Identity is not to actively handle it
- Clear the **To do** field

All of these properties are available as attributes of the target system group object. To change a group's attributes:

1. Select the group in the target system subtree to display its properties.
2. Click the tab that contains the attribute you want to change, and then click **Edit**.
3. Change the relevant attributes.
4. Click **Save** to save your changes.

12.7.3. Managing Group Hierarchies

DirX Identity allows using group folders and hierarchies. With this feature:

- DirX Identity allows Group folders to structure groups only if the corresponding connected system supports group folders (for example, LDAP directories or Active Directory).
- The structure must be identical in DirX Identity and in the target system.
- You can only create folder structures in the connected system. These structure can be

synchronized with the target system workflows to DirX Identity.

Creating group folders in DirX Identity requires that you customize the target system workflows.

12.7.4. Managing Imported Groups

This section explains the options for handling imported groups. Imported groups are immediately set to state ENABLED. The following figure illustrates the options for handling imported groups.

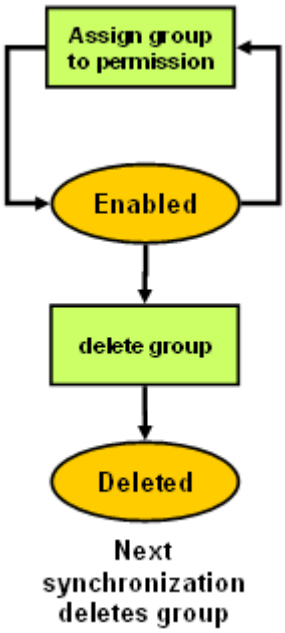


Figure 55. Alternatives for Handling Imported Groups

Processing tasks on the imported groups after the initial load workflow has executed include:

- Identifying the imported groups to the privilege administrator so that he can assign them to permissions in the privilege structure (or assign the groups to permissions yourself, if you are the privilege administrator). See the topic "Managing a Permission's Group Assignments" for details.
- Assigning permission parameters and their values to the imported groups; see the topic "Assigning Permission Parameters" for details
- Adding additional values for attributes that were not imported from the connected system.
- Deleting the imported groups that do not match the privilege structure. See the topic "Deleting a Group" for details (analogous to deleting accounts). The next synchronization deletes the group in the connected system. The group in the target system will be deleted by the privilege resolution or consistency checker after the end date is reached.



You can use rules to automate some of these tasks.

12.7.5. Managing Group Memberships

The target system workflows - especially the validation workflows - import the accounts and groups, and also import their memberships: the account-group assignments. They assign the IMPORTED state to newly imported assignments. The following figure illustrates the alternatives for managing imported group memberships.

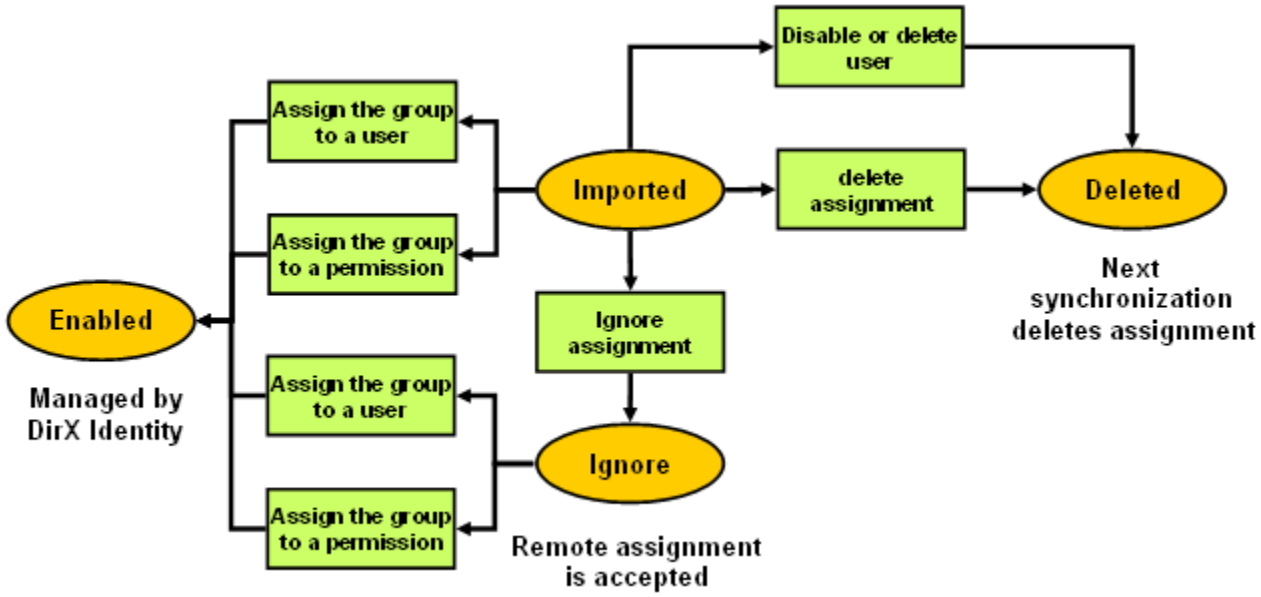


Figure 56. Alternatives for Handling Imported Group Memberships

To display the memberships of the group, select it in the subtree, then click the **Member** tab. Manager displays the accounts that are members of this group and the account-group assignment state. You can perform these tasks:

- **Change the assignment state manually**
  Edit the group, select the Member tab and then select the assignment in the **State of Assignment** column. You can

  - Set the assignment to DELETED. The next synchronization removes the account from this group in the connected system.
  - Set the assignment to IGNORE. This setting indicates that this assignment is accepted and controlled from the connected system.

  You can automate these tasks by configuring validation rules of type **Accept** or **Cleanup**. Accept rules set assignment state of the matching members to IGNORE. Cleanup rules set it to DELETED. For a description how these validation rules work, see the section "How Rules Work" in the "Managing Policies" chapter.

- **Assign the group to a user or a business object**
  You can do this by hand, with a rule or with business object inheritance. The assignment state of this assignment changes to ENABLED, as does the corresponding account state. From now on, this assignment is controlled by DirX Identity.
- **Assign the group to a permission**
  If the permission or any higher level role is already assigned to users, the assignments of

these users are created automatically as well as the necessary accounts (all go to state ENABLED). Existing accounts and assignments (coming from an import) that are assigned via this permission or role change their states to ENABLED. From now on, this group is controlled by DirX Identity.

After performing these tasks, you should not find any more group memberships in state IMPORTED.Use the **todoMemberhips** query folder to check on it.

If the member is a group, Manager displays the state NESTED.Account members of the nested group are also shown with state NESTED.

12.7.6. Deleting a Group

You can delete a target system group by hand.To delete a group, right-click it in the subtree and then click **Delete**.DirX Identity Provisioning changes the group's state to DELETED, that is, the group is not immediately and physically deleted.

The target synchronization workflow attempts to delete the group from the connected system the next time it runs.If the operation is successful, it sets the group's dxrTSState to DELETED.Running the consistency rule cleanupDeletedObjects will remove the group if its audit trail is empty.

12.8. Deleting a Target System

Deleting a target system means deleting the target system data from the Identity Store (Provisioning Configuration).The connected system itself is unaffected by this operation.

To delete a target system, click it, and then select **Delete** from the context menu.DirX Identity Manager (Provisioning) checks that you are authorized to delete the system, and then requests that you confirm the delete.The DirX Identity Provisioning system then deletes from the Identity Store (Provisioning Configuration):

- All assignments from Provisioning users and permissions to the target system's accounts and groups
- All the target system accounts
- All the target system groups
- The target system and its object descriptions

DirX Identity Provisioning returns a warning message if it cannot remove a user-account or permission-group assignment; in this case, you must remove the assignment by hand with DirX Identity Manager (Provisioning).

You must also use DirX Identity Manager (Connectivity) to disable the schedules for the target system's synchronization and validation workflows. Alternatively, you should delete the workflows and connected directories that were used with the deleted target system. Perform these activities in this sequence:

- Global View - remove the connected directory related to the deleted target system from all scenarios (there could be several instances).

These visual representations are only links to the real object, thus you need to delete the real object in the next step:

- Expert View - remove the connected directory.
- Expert View - remove all related workflows.
- Expert View - remove all related jobs.
- Expert View - remove all related channels from the Identity Store.

13. Managing Compliance

Compliance management consists of the following tasks:

- Setting up segregation of duties (SoD) to ensure that conflicting privileges are not assigned to users.
- Setting up risk management to ensure that high-risk privileges are only assigned to users when absolutely necessary.
- Setting up certification campaigns to control user access rights to resources in connected systems.
- Setting up the DirX Identity audit trail to get historical information about objects.
- Setting up and using DirX Identity status reports to get up-to-date information about objects.

The next section describes these tasks in more detail.

13.1. Managing Segregation of Duties (SoD)

Segregation of duties (SoD) is the set of policies that specify user-privilege assignments that constitute conflicts of interest and the process of checking whether privilege assignments to a specific user are in conflict according to the policies. These policies work with any type of privilege and even with different types of privileges.

DirX Identity's features for SoD allow you to:

- Create an unlimited number of SoD policies.
- Combine any type of privilege with any other type to define a conflict (roles, permissions, groups). For example, you can define that a role is in conflict with a group.



You cannot define SoD policies for role parameters. Instead, you should define policies at the group level.

- Detect hierarchical SoD problems. The SoD checking algorithm can detect hierarchical SoD violations.For example, if you assign two roles that are not in conflict, DirX Identity might detect a conflict at the permission or group level during privilege resolution.
- View all accepted SoD conflicts for a user from the SoD Exceptions tab.
- View all SoD exceptions for a specific SoD policy from the SoD Exceptions objects that are located under it.
- Run a full SoD check at regular intervals. Use the FullSoDCheck rule via the FullSoDCheck workflow that runs the policy execution service.

To understand the SoD concepts, read the DirX Identity Use Case Document *Using SoD*.

13.1.1. How SoD Works in DirX Identity

This section explains how segregation of duty works within DirX Identity.

- To avoid reduced performance, you must activate this feature for each domain at the domain object explicitly (by using the Segregation of Duties (SoD) checks flag).
- To optimize performance, activating an SoD policy sets the **Has conflicting privileges** flag in the corresponding privileges. If the privilege is no longer part of an SoD policy, the flag is reset automatically.



If you intend to import SoD policies directly via LDAP, be sure that the **Is active** flag is not set. Set the flag manually after import for each SoD policy.

- All users are also checked to determine whether SoD violations already exist. If SoD violations are detected, the corresponding approval workflows are started (for the selection mechanism see below).
- During privilege resolution, the DirX Identity services layer checks for SoD conflicts only if a privilege is marked with the **Has conflicting privileges** flag. "Add privilege" orders handled in approval workflows that are already running are also checked. This means that already running approvals and the new approvals started by current actions are both counted. These SOD conflicts are stored in the orders of the approval workflows described in this section. If the DirX Identity Services layer detects a conflict, it starts an approval workflow automatically (for the selection mechanism see below).



This action is completely independent of whether the privilege is marked for approval. The approvers can see the SoD conflicts and must decide whether they approve the conflict.

- If the approvers reject the assignment, the privilege is not assigned to the user.
- If the approvers accept the assignment despite the SoD conflict (which makes sense in some cases), the privilege is assigned to the user. In this case:
  - The privilege is automatically marked for regular re-approval with an end date according to the default parameters of the domain.
  - The SoD conflicts are shown in the SoD Exceptions tab of each user entry.
  - For each SoD policy, you can view all existing SoD Exceptions as sub-objects.
- You can run FullSoDcheck workflows regularly to be sure that no undetected SoD violations exist in your domain. This workflow checks that all detected SoD violations already have a corresponding SoD exception.



It is important to understand that DirX Identity does not use special SoD workflows. Workflow selection operates as follows:

- If a privilege is flagged for approval (the "Requires approval" flag is set), the defined approval workflow is used also for SoD approval.
- If a privilege is not flagged for approval (the "Requires approval" flag is **not** set), a default

workflow is used for SoD approval.

For both cases, the workflow selection mechanism is described in the section "Workflow Selection" in the chapter "Understanding Request Workflows" in the *DirX Identity Application Development Guide*, where the DomainAdmin is the logged-in user (the DomainAdmin executes the SoD services). Be sure that the access policies allow the DomainAdmin to execute the required approval workflow(s).

To understand the SoD concept, read the DirX Identity Use Case Document *Using SoD*.

13.2. Managing Risk

DirX Identity provides the ability to perform risk assessment for identities based on an extensible set of risk factors.

When risk assessment is enabled at the domain and a risk policy is defined, DirX Identity's Risk Calculation workflow regularly calculates the risk factors for every user in the domain and then aggregates them into a compound risk, thus classifying users into risk categories from low to high. Web Center displays a user's risk category, and DirX Identity Manager displays a user's individual risk factors. Examples for risk factors include SoD violations, imported accounts and group memberships and total number of group memberships or privileged accounts. For any requested change in a user's privilege assignments, a DirX Identity assignment request workflow can compare the compound risk before and after privilege assignment. If the risk category increases as the result of the requested privilege assignment, additional approval steps can be required before the privilege can be assigned.Compliance officers, line managers and administrators can use DirX Identity's risk assessment feature to monitor the risk values in the domain and plan actions to reduce the number of high risk users; for example, by running appropriate certification campaigns or by enforcing additional approval steps.

This section explains how DirX Identity risk management works and how to configure it.

13.2.1. How Risk Assessment Works

DirX Identity provides the following mechanisms for performing risk management:

- A domain-wide control for activating and deactivating the risk assessment function.
- A risk policy object for defining the risk factors to be measured and their relative importance in the domain. Exactly one domain-wide risk policy must be defined.
- A Risk Calculation workflow that calculates the risk for every user in the domain and assigns every user a computed overall risk level of low, medium or high.
- A Risk Approval workflow for privilege assignments and deassignments that computes the potential risk level for the user if the change is approved. If the potential assignment indicates high risk, the workflow invokes an additional approval step.

13.2.1.1. Supported Risk Factors

DirX Identity supports the following risk factors:

- Group memberships
- Imported memberships
- Imported accounts
- Privileged accounts
- SoD violations

The Risk Calculation workflow computes each factor separately for every user. In the risk policy for the domain, you specify a weight for each factor to identify which factors are more important than others for your domain. Risk calculation takes place over the entire domain and the workflow stores each risk factor's mean and standard deviation in the risk policy object.

13.2.1.2. Computed Values

Values computed by the Risk Calculation workflow include:

- The raw value for a risk factor, which is the computed score of the factor at a specific user.
- The standard score for a risk factor, which is a normalized score for the factor at a specific user. It is computed in the following way:

  Standard Score = (raw score – arithmetic mean) / standard deviation
- The compound score for a user, which is all the standard scores for the user computed with the formula:

  $$\sum(\text{riskFactorScore}_i * \text{riskFactorWeight}_i)$$

  The result is a double-precision floating point number.
- The overall risk for a user, which is calculated using the high-risk and low-risk limits defined in the risk policy object. If the user's compound score exceeds the upper limit, the risk classification is high. If the user compound score falls below the lower limit, the risk classification is low. All values between are treated as medium.

13.2.1.3. How Risk Factors are Computed

This section describes how the Risk Calculation workflow calculates each supported risk factor.

13.2.1.3.1. Group Memberships

Some groups contribute more to a user's risk than others. In the same way, some applications (which are equivalent to target systems) are considered risky for a company. As a result, a risk weight can be attached to each group and to each target system. The default weight is 1 for both groups and target systems. So the risk score for the memberships of a target system would be:

$$\text{riskWeight}_{TS} * \sum(1_{Group} * \text{riskWeight}_{Groupi})$$

The risk score/raw value for all group memberships is then the sum of the target system scores.

13.2.1.3.2. Imported Memberships

The risk score is calculated in the same way as normal memberships.

13.2.1.3.3. Imported Accounts

The risk score is the sum of imported accounts, each account multiplied with the risk weight of its target system. An imported account is one with **dxrState** = **Imported**. Imported accounts might not be associated with a user. Therefore, they are not counted for any user.

13.2.1.3.4. Privileged Accounts

The risk score is the sum of privileged accounts, each account multiplied with the risk weight of its target system. A privileged account is one with **dxrType** = **Privileged**.

13.2.1.3.5. Sod Violations

Each SoD Policy that a user violates increases the risk. An SoD violation is stored as an LDAP entry of type **dxrSoDException** beneath the violated policy entry. Its attribute **dxrUserlink** refers the violating user.

13.2.1.3.6. Special Handling for Personas, User Facets and Functional Users

The Risk Calculation workflow computes and displays the risk values for all of these objects. For personas, the risk factors are implicitly added to the corresponding user. So at the user, the additional factors obtained via the persona are included. This means that risk factors are counted here - for example, special Group memberships - that you cannot see looking at the user entry (as they come just from an associated persona). For user facets, this action is performed implicitly during user facet inheritance. There is no special handling for functional users.

13.2.2. Configuring Risk Management

To configure your domain for risk assessment (risk-based governance):

- Define a risk policy for the domain
- Configure the Risk Calculation workflow
- Set up the Risk Approval workflow
- Activate risk management at the domain

The next sections describe these tasks in more detail.

13.2.2.1. Defining a Risk Policy

An inactive risk policy object is available in the My-Company domain at the location **Policies** → **Risk** → **Risk policies** → **Risk Policy** in DirX Identity Manager's Provisioning view.

In **Customer Domain**, create your own risk policy and check the **Is active** flag. Be aware that there should be only one active risk policy for a domain.

In the risk policy object:

- Select the risk factors that you want to use for risk calculation and assign a weight to each one according to your needs. Risk factors that you leave empty are not used for risk calculation.
- Define the upper and lower risk limits for overall risk calculation.

Last Risk Calculation, **Compound Score Deviation**, **Standard Deviation**, **mean** and **count fields** are populated by the Risk Calculation workflow.

13.2.2.2. Configuring the Risk Calculation Workflow

In DirX Identity Manager's Connectivity view, you'll find the template Risk Calculation workflow in:

Workflows → **Default** → **Identity Store** → **RiskGvnController**

Use this template to create this workflow type in your domain. You can use the Global View if you have already a scenario for your domain. If not, a scenario that includes this workflow type will be created during the first run of the Target System Wizard

Activate the workflow and create a schedule to run it periodically; for example, every night.

This workflow uses the RiskGovernance resource family. Be sure to configure threads for this resource family in your Java-based Server.

The workflow computes the risk scores for all users, personas, user facets and functional users in your domain. It stores the values at the corresponding objects. The workflow also populates the **Last Risk Calculation**, **Compound Score Deviation**, **Standard Deviation**, **mean** and **count** fields in the risk policy object.

13.2.2.3. Configuring the Risk Approval Workflow

n DirX Identity Manager's Provisioning view, you'll find the sample risk approval request workflow at:

Workflows → **Definitions** → **Default** → **Assignments** → **RiskApproval**

The first activity in this workflow calculates the new risk level that would become effective if the assignment was approved.If the risk level is computed to high (is equal to 3), an additional approval step **Approval by Company Head** must be executed if **Approve by Privilege Manager** has been accepted.For more information about the risk approval workflow, see the section "Understanding Assignment Workflows" in the chapter "Using Request Workflows" in the *DirX Identity Application Development Guide*.As with other assignment workflows, you can customize the risk approval workflow to your needs.See the *DirX Identity Customization* Guide for details.

13.2.2.4. Activating Risk Management

To enable risk management, check **Risk Check active checks** in the Compliance tab of the domain configuration object.This action enables the Risk Calculation workflow to run on schedule.You can check the risk policy object in DirX Identity Manager to view the time at which the workflow last ran.

To view the risk assessment for a user, user facet, persona or functional user, click the object's Risk Parameters tab.This tab displays the risk values computed during the last Risk Calculation workflow run.

13.3. Managing Certification Campaigns

Managing certification campaigns consists of the following tasks:

- Preparing a campaign
- Starting a campaign
- Monitoring a campaign
- Generating reports

For details on how to perform these tasks for user or privilege certification campaigns, see the DirX Identity Use Case Document *Certification Campaigns*.



The access certification technique is now deprecated in DirX Identity and is replaced with the technique used for user certification campaigns.

The next section provides information about setting up DirX Identity for certification campaigns. These steps are pre-requisites to running user certification campaigns and are described in more detail in the DirX Identity Use Case Document *Certification Campaigns*.

13.3.1. Enabling the Mail Workflow

Certification campaigns send mail at various intervals, so you need to ensure that the mail workflow is enabled:

- In the Provisioning View, browse to **Workflows** → **Configuration** → **Services** → **SMTP**
- Enable the mail workflow by providing the SMTP host address and port and removing the value **dummy** for **Map Mail Address**.

13.3.2. Setting up the Certification Campaign Controller

Before you can start, monitor or complete a certification campaign, you need to configure the Certification Campaign Controller workflow, add a resource family for it and create a schedule for it.

To add a resource family for the Certification Campaign Controller workflow:

- In the Connectivity view, browse to **Expert View** → **Configuration** → **DirX Identity Servers**

→ **Java Servers** → *your_domain_name* → *your_domain_server*.

- In the Resource Families tab, move the **CertificationCampaign** resource from the **Available** list to **Selected**. Set the value for **Thread number** to **1**, and then click **Save**.

To configure and enable the Certification Campaign Controller workflow:

- In the Connectivity Global View, select the appropriate scenario for your domain. Select the workflow line between the two Identity Store connected directories and then select **New** from the context menu.
- Select the Certification Campaign Controller workflow. The workflow wizard starts and guides you through the configuration options. In the first step, make sure to check **Is Active**. After the last step, click **Finish** to save the new workflow in your scenario.

To add a schedule for the Certification Campaign Controller workflow:

- Browse to Connectivity → **Expert View** → **Schedules**.
- Add a schedule task for **CertificationCampaignController**. See the context-sensitive help for details.

13.3.3. Creating a Certification Campaign Folder

The Certification Campaigns view contains certification campaigns. It also contains a folder named **Default**, which contains notification templates in the **Notifications** subfolder. You can use these templates when you set up a campaign.

When recurring certification campaigns are used, a folder **Archive** will be automatically created and will store all successfully completed certification campaigns. This folder is a backup repository and is not handled by the Certification Campaign controller.

To create a certification campaign, you create a new campaign folder:

- In the Certification Campaigns view, select a campaign folder and then select **New** from the context menu.

For more information on creating certification campaigns, see:

- The *DirX Identity Tutorial*.This document provides a tutorial on how to create a user or privilege certification campaign.
- The DirX Identity Use Case Document *Certification Campaigns*.This document provides use cases that illustrate how to create and manage user or privilege certification campaigns.



The access certification technique is now deprecated in DirX Identity and is replaced with the technique used for user certification campaigns.

13.4. Managing the Audit Trail

You can configure an audit trail for each DirX Identity domain.They track all changes in the Identity store and generate audit trail information from all workflows.

The audit records are in XML format.Most of them are stored in the Identity Store along with the entry that was changed, or, when produced by a workflow, in log files of the corresponding Java-based Server or pushed to a Message Broker, if DirX Audit is installed.Each audit record may be signed.

You can use the DirX Identity Provisioning audit trails in conjunction with the tracing information from the Tcl-based workflows to audit the DirX Identity Provisioning system.

This chapter explains

- How audit trail works
- The audit trail format
- How to set up audit trail
- How to handle the audit trail service
- How to verify signatures

13.4.1. How Audit Trail Works

DirX Identity's audit trail mechanism ensures that all relevant actions in the Identity Store and the Identity Servers are correctly tracked:

- Audit policies allow you to configure the DirX Services layer audit trail. This audit trail tracks all requests to the Identity Store coming from the user interfaces (Business User interface, Web Center, Identity Manager), Web Services (REST service, SPML service), consistency and resolution workflows, as well as the meta controller in Tcl-based provisioning workflows. These components store the audit records in the attribute dxrHistory of the affected LDAP entry.
- The auditing of Java-based realtime workflows can be activated per workflow, and for request workflows, globally for the whole domain. They store their audit trails in extra log files of their hosting Java-based server.
- For Web Center, auditing for authentication can be enabled. The corresponding audit records are written to the dxrHistory attribute in LDAP.

The following figure illustrates DirX Identity's audit service components.

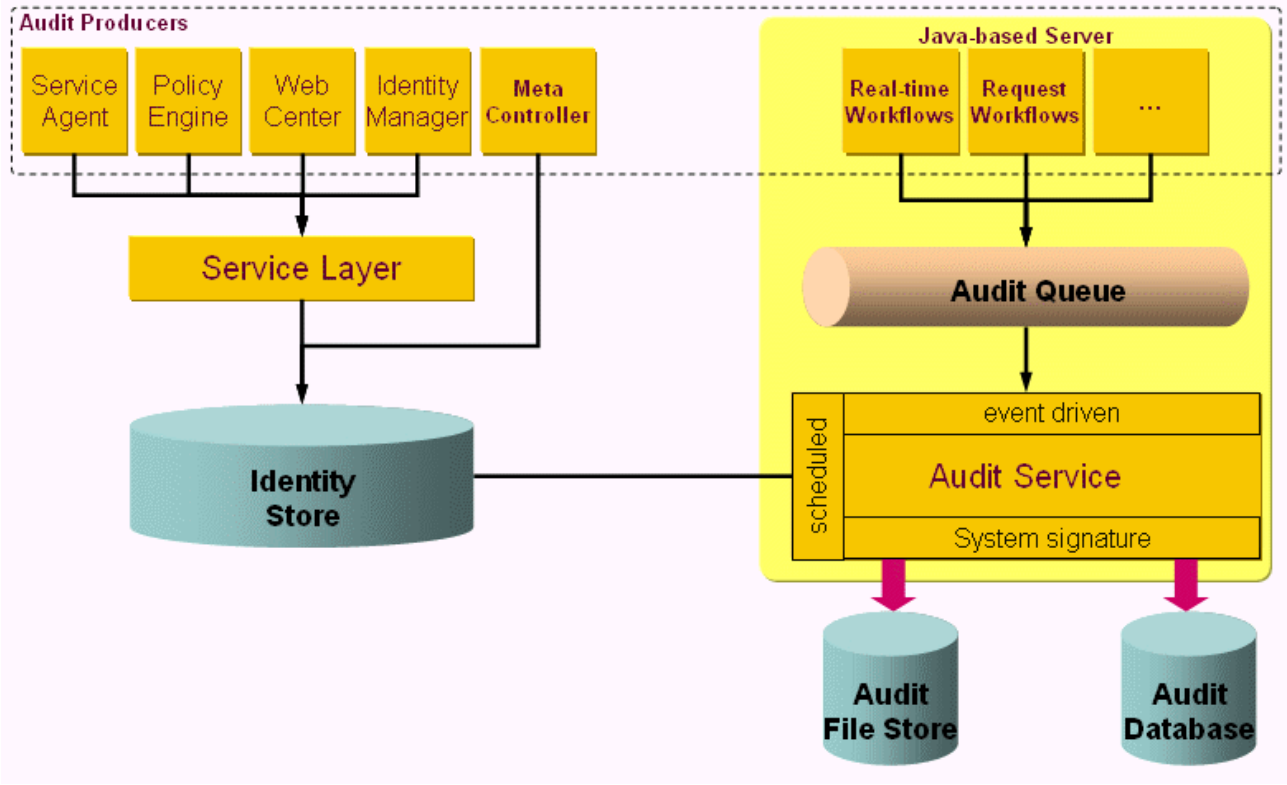


Figure 57. DirX Identity Audit Service

If DirX Audit is installed:

- The Java-based server can send the audit trail directly to DirX Audit's Message Broker using the JMS audit handler. A DirX Audit JMS collector picks them up immediately and stores them in the Audit Database.
- DirX Audit server regularly reads the audit records from the dxrHistory attribute of the LDAP entries, stores them in the Audit Database and removes them from the LDAP entry.

Without DirX Audit, you can schedule the **ExtractAuditTrail** workflow to read the audit trail from dxrHistory attributes and store them in files. Afterwards, it also removes the extracted values from LDAP.

DirX Identity comes with a pre-configured setup of objects and attributes that are tracked by the audit trail mechanism. You can change this default audit trail configuration as necessary.



Either create your own audit policies or copy the default policies to prevent any changes you make to them from being overwritten during an upgrade installation.



You can only establish auditing for objects that have a dxrHistory attribute in their schema definition. The corresponding object description must also contain a definition for the dxrHistory attribute. Otherwise, the audit policy does not allow selecting this object type.

13.4.2. Identifying Multiple Causes of Account and Group Changes

To audit the complex operations in DirX Identity, it is essential to identify the causal event of a group membership change or an account creation or account state change. The following causes for these changes exist:

- Changes to date fields at the user (can result in state changes)
- Privilege assignments at the user
- Change to permission parameters at the user
- Changes in privilege structure
- Privilege assignments to business objects

Each audit record contains the ID of the causing audit event, if it was triggered by another action. Audit causes are transported from the causal event (like a new privilege assignment) down to secondary (account and group modification) and ternary events (provisioning actions at the connected system) automatically. The system makes sure to pass the causing event ID to downstream actions, either by putting the ID into messages or by storing it at the changed LDAP object (like the changed role). If the object was changed multiple times, there can be more than one causing ID. You can see them in audit records in the **why → referenceID** section; the multiple cause IDs are separated by blanks.

The following figure shows the use cases in more detail.

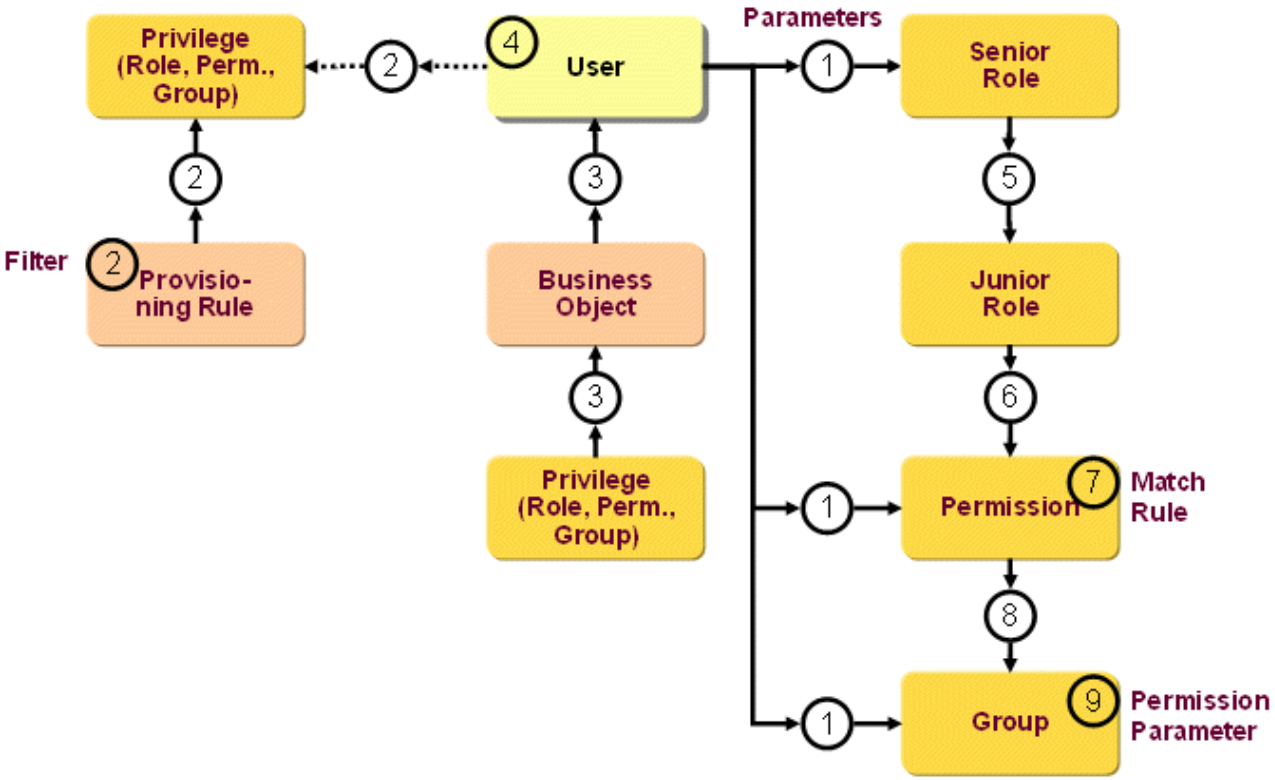


Figure 58. Possible Causes for Account or Group Changes

As shown in the figure, the possible causes for account or group changes are:

1. Adding, changing or removing an assignment to the user manually in any client (for

example, Identity Manager, Web Center or a consistency rule in the Policy Execution). Change covers changes of start- and end date and of role parameters. If a privilege is flagged for approval, the assignment is forwarded to the corresponding request workflow.

2. Adding or removing an assignment to the user by executing a provisioning rule.
3. Adding or removing an assignment to the user by business object inheritance.
4. Changing a date field or a permission parameter at the user. Changing a date field can result in a state change. Permission parameter changes may affect the evaluation of a permission match rule
5. Adding or removing a junior role assignment to the role.
6. Adding or removing a permission assignment to the role.
7. Adding, changing or removing a match rule at the permission.
8. Adding or removing a group assignment to the permission.
9. Changing a permission parameter at the group.

13.4.2.1. Identifying and Referencing Audit Causes

Each audit trail is identified by a unique audit trail ID. A sample ID is uid-c0a8c84-2701ff04-130994315e4—7fff.

If multiple causes occur at one object in one edit step (for example two or more role assignments are added to the user before the user is saved), the audit trail IDs of the second (and further) causes are deduced from the first cause's ID (= basic ID) by incrementing an integer number, for example

```
uid-c0a8c84-2701ff04-130994315e4--7fff: first cause, basic ID
uid-c0a8c84-2701ff04-130994315e4--7fff-1: second cause
uid-c0a8c84-2701ff04-130994315e4--7fff-2: third cause
```

In this special case, in the account or group audit trails, the first cause is always referenced in the **why → referenceID** element. This is done since the current reference algorithm cannot distinguish between the different causes.



The numbering is arbitrary, which means that each of the causes may be the real causal event of the account/group change.

13.4.3. About Audit Trail Format

Audit trails are in XML format. For example, setting an end date at a user which results in a deactivation of this user entry is recorded as follows:

```
<audit:auditRecord recordID="uid-8b19a156--48df5847-11c27441c27--
7f96" seqNo="25" when="20080903085602.000Z"
```

```
xmlns:audit="urn:com:siemens:dxm:audit:1:0"
xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
xmlns:history="urn:com:siemens:dxr:history:1:0"
xmlns:spml="urn:oasis:names:tc:SPML:1:0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<audit:activeParticipant>
<spml:identifier type="urn:oasis:names:tc:SPML:1:0#DN">
   <spml:id>cn=Taspatch Nik,ou=Global IT,o=My-Company,cn=Users,cn=My-
Company</spml:id>
   <spml:identifierAttributes>
       <spml:attr name="Name">
            <dsml:value type="string">Taspatch Nik</dsml:value>
       </spml:attr>
        <spml:attr name="AltName">
            <dsml:value type="string">5326</dsml:value>
        </spml:attr>
        <spml:attr name="First Name">
            <dsml:value type="string">Nik</dsml:value>
        </spml:attr>
        <spml:attr name="Last Name">
            <dsml:value type="string">Taspatch</dsml:value>
        </spml:attr>
    </spml:identifierAttributes>
    <audit:uid xsi:type="audit:uid">uid-7f001-2c9c34a9-11c24031414--
7ff9</audit:uid>
    <audit:objectType
xsi:type="audit:objectType">User</audit:objectType>
</spml:identifier>
</audit:activeParticipant>
<audit:whereFrom application="ApplyChange" server="139.25.161.86"/>
<audit:whereTo port="389" server="kaiserswerth06"/>
<audit:why>
<audit:type>onRequest</audit:type>
<audit:referenceID>uid-8b19a156--48df5847-11c27441c27--
7fad</audit:referenceID>
</audit:why>
<audit:operation operation="modify">
<audit:operationCategory>Object</audit:operationCategory>
<spml:identifier type="urn:oasis:names:tc:SPML:1:0#DN">
    <spml:id>cn=Franca Baretti 8744,cn=Accounts and Groups,cn=Windows
Domain Europe,cn=TargetSystems,cn=My-Company</spml:id>
```

```
    <spml:identifierAttributes>
        <spml:attr name="Name">
            <dsml:value type="string">Franca Baretti
8744</dsml:value>
        </spml:attr>
        <spml:attr name="AltName">
            <dsml:value type="string">uid-7f001-2c9c34a9-
11c24031414--7ff1</dsml:value>
        </spml:attr>
        <spml:attr name="First Name">
            <dsml:value type="string">Franca</dsml:value>
        </spml:attr>
        <spml:attr name="Last Name">
            <dsml:value type="string">Baretti</dsml:value>
        </spml:attr>
    </spml:identifierAttributes>
    <audit:uid>uid-7f001-2c9c34a9-11c24031414--76c5</audit:uid>
    <audit:objectType
xsi:type="audit:objectType">Account</audit:objectType>
</spml:identifier>
<spml:modifications>
    <spml:modification name="State" operation="add">
        <dsml:value type="string">DISABLED</dsml:value>
    </spml:modification>
    <spml:modification name="State" operation="delete">
        <dsml:value type="string">ENABLED</dsml:value>
    </spml:modification>
    <spml:modification name="EndDate" operation="add">
        <dsml:value type="string">20081002220000.000Z</dsml:value>
    </spml:modification>
</spml:modifications>
</audit:operation>
<audit:response result="urn:oasis:names:tc:SPML:1:0#success"/>
</audit:auditRecord>
```

This audit trail example shows most of the features:

Header

each trail starts with general information:

seqNo

if the trails are stored in a file, a sequence number protects deletions of items (this

requires a system signature).

when
each trail comes with the date and time stamp at which this event occurred

activeParticipant
indicates the person that initiated this event. It contains a set of identifying attributes:

spml:id
the DN of the person

audit:uid
the audit trail UID. DirX Identity automatically provides a UID value that allows the object to be tracked for as long as it exists. Moves or renames of the object do not change this UID value.

audit:objectType
the object type to which this event belongs. You can use audit policies to define this value per object type.

spml:identifierAttributes
the attributes for each object type that are to be delivered with each audit trail. You can use audit policies to define these values per object type.

audit:whereFrom
information about the application that caused the event and the server on which this application ran.

audit:whereTo
information about the server and the location to which the event was written.

audit:why
the reason the event was created:

audit:type
the type of event (event based, manually triggered, ...)

audit:referenceID
only available if this event (a secondary event) was caused by another event (primary event). This information allows for tracking dependencies between events.

operation
the audited operation in detail:

audit:operation
the type of operation (add, modify, delete)

audit:operationCategory
a coarse-grained category for the operation (object, assignment, ...)

spml:identifier
: the object. It contains a set of identifying attributes. For details, see the activeParticipant description in this section.

spml:modifications
: the modifications in detail.

audit:response
: the operation result (success, failure)

Features that are not represented in the example trail shown here include:

System signature
: each trail can optionally be protected by a system signature which guarantees that the audit trails cannot be changed.

Client signature
: most manual actions can optionally be signed with a client certificate to guarantee the identity of the activeParticipant.

Audit trails remain in the Identity Store (Provisioning Configuration) until the ExtractAuditTrail workflow writes them to a file and then deletes them from the Identity Store.

DirX Identity Provisioning does not provide any clients for audit trail viewing. Use standard XML tools instead.

To prevent audit trails from being deleted with their related object, the objects remain in the Identity Store (Provisioning Configuration) until the audit trail is saved by a ExtractAuditTrail workflow. A special consistency rule **cleanupDeletedObjects** can be used to periodically delete all objects without audit trails.

13.4.4. Setting up Audit Trail

Configure the audit trail mechanism for DirX Identity and for Web Center authentications at the service, request workflow, and real-time workflow layers. The next sections provide instructions on how to perform these tasks, as well as providing information about how to set up audit policies for custom object types and set up client signatures.

13.4.4.1. Auditing for the Service Layer

To set up auditing for the DirX Identity Services layer:

- In the Compliance tab at the Domain object, check the **Enable Auditing for** → Service Layer flag.
- In the **Audit Trail Database** connected directory in the Connectivity view, set the path and filename where the audit information is to be written in the **Filename** field of the **Data File** definition. For more details, see the section "Handling Audit Trail Workflows".
- In the **Provisioning** → **Auditing** view, set up audit trail policies to configure the details of

audit trail recording. For instructions on how to set up audit trail policies for your own object definitions, see the section below.

13.4.4.2. Auditing for Request Workflows

To set up request workflow auditing:

- In the Compliance tab at the Domain object, check the **Enable Auditing for** → Request Workflows flag.
- In the Compliance tab at the Domain object, check the **Enable Client Signature for** → **Request Workflows** flag if you want to work with client signature for people activities. In this case the relevant people activities must be flagged accordingly.
- In the Compliance tab at the Domain object, check the **Enable Client Signature for** → **Privilege Assignment Requests** flag if you want to work additionally with client signature for privilege assignment requests.
- In the Compliance tab at the Domain object, check the **Enable Client Signature for** → **Attribute Modification** flag if you want to work additionally with client signature for attribute modification requests.
- At each Java-based Server object (see the **Audit Trail Folder** field), define the path where the audit trail information for request workflows is to be written.

13.4.4.3. Auditing for Real-time Workflows

To set up auditing for real-time workflows:

- You can enable auditing for real-time workflows at each individual real-time workflow (see the activities' **Write Audit Log** flag). There is no central flag to enable / disable auditing for real-time workflows.
- At each Java-based Server object (see the **Audit Trail Folder** field), define the path where the audit trail information for real-time workflows is to be written.

13.4.4.4. Auditing for Web Center Authentications

To set up auditing for successful and failed attempts to log in to Web Center:

- In the Compliance tab at the Domain object, check the **Enable Auditing for** → **Authentications** flag.
- In the **Audit Trail Database** connected directory in the Connectivity view, set the path and filename where the audit information is to be written in the **Filename** field of the **Data File** definition. For more details, see the section "Handling Audit Trail Workflows".
- If auditing for the DirX Identity Services layer is enabled and an audit policy for user objects is activated, the audited identifying attributes for users authenticating against Web Center are taken from that policy. Otherwise, some default identifying attributes apply.

13.4.4.5. Setting up Audit Policies for Custom Object Types

You can create audit policies for your own custom object types.

If you use an object description with a symbolic name (for example, '<object name="myObject">' that is represented as the LDAP object class myObjectClass, you should map these values in the file Customer Extensions → Object Descriptions → ObjectTypeMap.properties in the form:

```
myObject=myObjectClass
```

Now you can select myObjectClass from the drop-down list in the access policy definition.

13.4.4.6. Setting up Client Signature

Client signature works with both, Internet Explorer and Firefox.

We recommend setting up a certificate owner check. For more information, see the section "Customizing the Certificate Owner Check" in the chapter "Customizing Auditing" in the *DirX Identity Customization Guide*.

13.4.4.6.1. Internet Explorer

Take into account the following prerequisite to set up client signature:

- Microsoft CAPICOM SDK (version 2.1.0.2 or higher) is a prerequisite for operation of DirX Identity with client signature.



Microsoft does not support CAPICOM on Windows 7.

Perform the following steps to install CAPICOM:

- Download the correct versions from the vendor's site and install them.
- Navigate to the folder containing the shared object **capicom.dll**.
- Register the shared object using the suitable operating system commands.

The following sample command sequence shows how to locate and register capicom.dll using a DOS command prompt:

```
cd C:\Program Files\Microsoft CAPICOM 2.1.0.2 SDK\Lib\X86
regsvr32 capicom.dll
```

13.4.4.6.2. Firefox

Firefox supports client signature out of the box. Depending on the PKI infrastructure used, you must import relevant certificates into the browser's certificate store and you may have to load a security module for card reader access.

13.4.5. Handling Audit Trail Workflows

The **ExtractAuditTrail** workflow creates audit trail records. This workflow reads audit trail from the Identity Store and writes them to the file identified by the **AuditTrailDB** connected directory. To configure the workflow, right-click **ExtractAuditTrail** and then select **Configure**:

- The Export Properties tab lets you define the **Base Object**, **Subset** and **Search Filter** for audit trail record retrieval. All objects included in the search are handled by the workflow.
- The Import Properties tab contains a link to the file object where you can define the path and filename prefix of the audit trail files, for example: C:\MetahubData\AuditTrailDB\auditTrail_
  In this case, the files are written to the directory **C:\MetahubData\AuditTrailDB**. During each run, a file with the name *prefix timestamp*.xml is generated.
- The Set Tracing tab allows you to define the trace parameters for the audit trail workflow.

You can define several audit trail workflows that handle different parts of your Identity Store in parallel or at different times.

13.4.6. Verifying Signatures

You can verify the signatures of written audit files with an external tool.

Unpack the zip file:

*installation_path***tools\utilities\runSignatureCheck.zip**

Run the batch script:

runSignatureCheck.bat *nameOfFileToCheck*

For UNIX platforms, use the corresponding **sh** command.

13.5. Managing Status Reports

For each DirX Identity domain, you can configure status reports that reflect the current status of objects in the identity store.

This section explains how to generate reports.See the *DirX Identity Customization Guide* for information on how to customize status reports.

13.5.1. Generating Status Reports

DirX Identity Provisioning provides several ways to produce status reports about the objects in the Identity Store (Provisioning Configuration). You can produce them:

- With the Web Center, to produce object status reports. The type of status reports you can produce depends on the access policies that you have set up.

- Dynamically, by hand, using DirX Identity Manager (Provisioning) to generate a status report and display it online or write it to a file, as desired.
- Automatically on a regular basis, by scheduling a Tcl-based workflow that generates the status reports (one job for each report type).



There are two different workflows for the Connectivity and Provisioning configuration.

- With the runReport tool to generate a report in the Connectivity configuration.



There is currently no runReport tool to generate reports in the Provisioning configuration. For details about how to configure and run this tool, see the section "Using the Run Report Tool" in the chapter "Using DirX Identity Utilities" in the *DirX Identity User Interfaces Guide*.

Status reports that are generated and written to files (either through DirX Identity Manager (Provisioning) or through a status report generator workflow) can be output in the defined format; this selection is made when you configure the workflow for generating the report, and is made on a per-job basis. XML status reports are useful for import into standard report generators like Crystal Reports, or to Microsoft Excel or a database to enable statistical analysis. HTML reports are "ready to view".

14. Managing Domains

Domain management consists of the following tasks:

- Performing the configuration of a newly created domain
- Using object collections
- Monitoring workflows

14.1. Configuring a Domain

Configuring a domain consists of the following tasks:

- Specifying the administrators who are to manage the Provisioning users, the privilege structure, and the target systems and define their access rights.
- Customizing the standard object descriptions for users, roles, permissions, target systems and folders and (optionally) customizing their presentation in DirX Identity Manager (Provisioning) - this task is only necessary if you want to use different attributes or default values for these Provisioning objects.
- Managing critical attributes / parameters that influences via policies or match rules the assignment of privileges.
- Specifying the user attributes that can be used as permission parameters.
- Setting up role parameters.
- Adjusting domain-specific parameters such as the maximum time that an object in the "to be deleted" state can remain in the DirX Identity store (Provisioning Configuration).
- Setting up the report templates for the domain
- Setting up and adjusting the links between the Provisioning and Connectivity domain.

14.1.1. Defining the Administrators

Defining the administrators means defining the domain administrator(s), user administrator(s), role administrator(s) target system administrator(s), informational user(s) and auditors for the domain. For each person who is to perform one or more of these administrative tasks, you must:

- Create a user account
- Specify the administrative roles assigned to the person

14.1.1.1. Pre-configured Administrative Roles and Access Rights

DirX Identity Provisioning sets up each domain with several user groups that represent roles for DirX Identity Provisioning domain administration. DirX Identity Provisioning creates an entry for the domain administrator (**cn=DomainAdmin,cn=** *domain_name*) and the administrator group entries shown in the following table in **cn=DirXmetaRole,cn=TargetSystems,cn=** *domain_name*.

Administrator Group	RDN
Domain Administrators (DA)	cn=DomainAdmins
User Administrators (UA)	cn=UserAdmins
Role Administrators (RA)	cn=RoleAdmins
Target System Administrators (TA)	cn=TSAdmins
Informational Users (IU)	cn=InformationalUsers
Auditors	cn=Auditors



The role administrator is the privilege administrator. The name has not been changed for compatibility with earlier DirXmetaRole versions.

These administrator groups exist in the DirX Identity Provisioning target system that is provided in each DirX Identity Provisioning domain and are ready for assignment to users (the DirX Identity Provisioning target system acts as a virtual target system for the storage and maintenance of DirX Identity Provisioning administrators).

DomainAdmin is a member of all groups. This user must make the other Provisioning users members of these administrator groups according to their responsibilities.

Anonymous users do not have access rights to read entries or attributes in the domain. Every user that has authenticated with name and password (simple authentication) is permitted to change his password.

Members of the administrator groups that have authenticated with name and password are permitted to read all entries and attributes. For modify rights to entries and attributes, the group members are granted access rights according to the following table.

Directory object	DA	UA	RA	TA	IU	AU
Users subtree	all/*1	all/all	all/all	all/all	-	-
Assignments subtree	all/*1	all/all	all/all	-	-	-
Role catalogue subtree	all/*1	-	all/all	-	-	-
Permissions subtree	all/*1	-	all/all	-	-	-
Target systems subtree	all/*1	-	-	all/all	-	-
Configuration subtree	all/all	-	-	-	-	-
dxrTargetSystemAccount object class	all/*1	all/all	all/all	all/all	-	-
dxrTargetSystemGroup object class	all/all	all/all	all/all	all/all	-	-

In the table:

- The notation **all/all** means all access rights for **all** entry operations and all access rights for **all** attributes
- The notation **all/*1** means all access rights for **all** entry operations and all access rights for all attributes excluding the rights for making assignments (user-role, user-

permission, user-group, role-role, role-permission, permission-group, account-user)

You use the DirX Identity Manager Provisioning users view to create the users that correspond to DirX Identity Provisioning administrators. You use DirX Identity Manager Provisioning Data view to set passwords for users. You then grant them the necessary access rights to the objects in the Identity Store (Provisioning Configuration) by making them members of the defined administrator groups. You assign the administrator groups to the users either directly, by creating user-group assignments, or indirectly, with the DirX Identity Provisioning user-role-permission assignment process. For example, you can create a target system administrator permission and assign it the target system administrator group, then create a target system administrator role and assign the target system administrator permission to it, and then assign the target system administrator role to one or more users, who are then granted the access rights represented by the target system administrator group in the Identity Store (Provisioning Configuration).

14.1.1.2. Domain Super Administrators

The DirX Identity Provisioning administrators just described are only allowed to access and modify Provisioning objects. The DirX Identity Provisioning system also requires an administrator role that is allowed to modify the schema and access rights in the domain; this type of administrator is called the Domain Super Administrator. DirX Identity Provisioning provides the following accounts for Domain Super Administrators:

cn=DomainAdmin,cn=*domain_name* (rights are restricted to *domain_name*)

cn=SystemAdmin,cn=DirXmetaRole-SystemDomain (rights for all domains)

14.1.2. Customizing Object Descriptions

DirX Identity Provisioning provides template descriptions, in Extensible Markup Language (XML) format, of all Identity Store (Provisioning Configuration) objects: users, roles, permissions, target systems, folders, accounts, groups, and so on. DirX Identity Provisioning allows you to customize these descriptions to the requirements of your domain. For example, you may want to use a customer-specific attribute in the description of a user object, or you may want to specify your own default values for attributes of these objects.

DirX Identity Provisioning also provides template property page descriptions for Provisioning objects, in Bean Markup Language (BML) format, that control the objects' presentation (tabs and properties) within DirX Identity Manager (Provisioning). You can also customize these templates to your requirements.

14.1.2.1. Viewing and Editing Template Descriptions

You can use DirX Identity Manager (Provisioning) to modify the DirX Identity Provisioning template object and property page descriptions that are specific to the domain. The description of each object is split into two parts: a system part and a customer extension part. The system part is likely to be replaced on upgrade installations. Therefore, enter your modifications into the customer extension parts. Use the *Domain Configuration* view and find these templates in the following folders underneath the entry *cn=Customer Extensions,cn=<domain root>*:

- Object Descriptions folder - stores the domain-specific object descriptions (users, roles, permissions, target systems and so on)
- Property Page Description folder - stores the domain-specific property page descriptions for users and group objects
- JavaScripts folder (optional) - stores the Java scripts that are referenced from within the object descriptions

The templates for target system objects (accounts, groups and the target system entries themselves) are located underneath each target system's Configuration folder in subfolders with the same names as used for the domain.

To open a description for editing with DirX Identity Manager (Provisioning):

1. Select it in the subtree to display its properties.
2. Select the Object Descriptions tab, and then click **Edit**.

Now you can use the object description editor to make changes to the description. The document *DirX Identity Customization Guide* describes the structure of object and property page descriptions and how to customize them.

Alternatively, you can use DirX Identity Manager (Provisioning) to write the description to a file, and then use your favorite XML editor to edit it within the file system. You must then import the file back into the Identity Store (Provisioning Configuration) (with DirX Identity Manager (Provisioning)) to update the template.



If you are adding domain-specific user or other attributes to the Provisioning object descriptions, you will also need to add these attribute types to the default LDAP directory schema supplied with DirX Identity Provisioning. You use the schema modification tools of the particular LDAP directory to accomplish these tasks. For example, if you are using the DirX LDAP directory, you use the DirX administration tools for modifying the schema. The *DirX Identity Customization Guide* provides more information about these tasks.

14.1.2.2. Upgrading Customized Descriptions

Each upgrade installation of DirX Identity typically supplies new versions of the standard domain (and target system) object and property page descriptions. It overwrites the system specific descriptions located underneath the folders *Object Descriptions*, *Property Page Descriptions* and *Target Systems* direct underneath the domain root, but keeps the entries underneath *Customer Extensions* un-changed.

Hence, if you haven't modified the system entries, you need not adapt your configuration after upgrade installations.

14.1.3. Managing Critical Parameters

You can use user attributes in DirX Identity in different context. If a user attribute influences via policies or match rules the assignment of privileges, this attribute must be seen as

critical attribute. Changing such an attribute changes the access rights in the provisioned target systems.

Attributes of this type are used:

- In the matching rules of permission.
- As role parameters for a role. These attributes are used as permission parameters during the privilege resolution process.
- In a provisioning rule.



These attributes are currently named "Permission Parameters". In the future, this name will be changed to "Critical Parameters".

The domain administrator must specify these user attributes at the domain object.

We recommend handling these attributes as follows:

- Extend the relevant object descriptions for the user object with these parameters.
- Define these attributes at the domain object in the Permission Parameters tab.
- You can handle these attributes in a special way if you set up a **User Attribute Policy**. It allows you to control changes of these attributes via request workflows.
- You can set up matching rules in roles or permissions that use these parameters (see the next chapters for more information). Setup of matching rules requires extensions to the corresponding group objects (adjust the object descriptions).
- You can set up filters in provisioning rules that use these parameters.

14.1.4. Specifying Permission Parameters

The domain administrator must specify the user attributes that can be used as permission parameters. Use the Permission Parameter tab of the domain object to set up this list. The attributes selected will be valid throughout the domain (not just in some administrative area of the permissions subtree).

To manage the domain's list of permission parameters:

1. Select the domain root of the Domain Configuration tree. DirX Identity Manager (Provisioning) displays the list of user attributes that are currently defined as permission parameters.
2. Click **Edit**.

Now you can:

- Add new user attributes to the permission parameter list
- Remove previously selected user attributes from the permission parameter list

When you click **Save**, DirX Identity Manager (Provisioning) validates that all user attributes that have been removed as permission parameters are not still in use in the matching rule

fields of permission objects in the domain. Manager returns an error if de-selected permission parameters are still in use in permission matching rules.

When you add a user attribute as a permission parameter, you (or the responsible administrator) must:

- Define the matching rules that use the permission parameter in the permission or role objects
- Ensure that the new permission parameter is visible in the Provisioning user object (only if used in permission match rules)
- Ensure that the permission parameter values are set for the users (only if used in permission match rules)
- Extend the object description of target system groups to contain the permission parameter
- Extend the property page description of the target system groups to display the permission parameter
- Supply permission parameter values in the target system groups

The *DirX Identity Customization Guide* describes this process in more detail.

14.1.5. Setting up Role Parameters

Role parameters are a powerful mechanism for reducing the number of roles in your Identity system. See the chapter "Managing the Privilege Resolution Process" for a basic description of how role parameters work. See also the section "Access Policies for Hierarchical Role Parameters" for another aspect of this mechanism.

You can use various types of role parameter objects:

- **String** - a list of role parameter values. The source for such a role parameter object can either come from a fixed list of values or from a proposal list.
- **Text** - a pure text string. Can be used for comments or additional non-structured information.
- **Integer** - a list of integer values.
- **DN** (distinguished name) - a list of DNs.
- **Hierarchical DN** (distinguished name) - a list of hierarchical DNs that can be used as hierarchical ordered tree of values.

The DN and Hierarchical DN types are particularly powerful options. The differences are:

- At the Web Center assignment user interface (Assign Privileges) DNs are presented as a linear list, Hierarchical DNs as tree.
- Hierarchical DN values can be protected via access policies.

You can specify single-valued or multivalued role parameters. Parameters of type Text you can set either for mandatory or optional input. All other types are always mandatory. You can also set an optional default value.

14.1.6. Setting Domain Parameters

One task of setting up a domain is to establish values for the parameters that control DirX Identity Provisioning operation for the domain. To display these parameters, select the root of the domain configuration subtree, then click the Options tab. Click **Edit** to change the values.

You can set the following parameters:

- Global parameters like the domain type or information about the corresponding Connectivity domain.
- Numerous feature flags that influence the operation of services that work for this domain, for example the segregation of duties flag or various flags to enable or disable policies.
- Permission parameters that are used within this domain.
- Global timing parameters.
- Parameters to configure auditing.

For a detailed description of these parameters, see the context-sensitive help of the domain object.

14.1.7. Setting up the Report Templates

DirX Identity Provisioning allows you to generate the following types of reports:

- For DirX Identity Provisioning users, their assigned privileges; this type of report is called the "User" report
- For roles:
  - Their authorized users, which are the users directly assigned to the roles (not all the affected users of a role); this type of report is called the "Authorized Users" report
  - Their junior roles and permissions, including aggregated permissions; this type of report is called the "Roles" report
- For permissions, their assigned groups; this type of report is called the "Permissions" report
- For each target system:
  - Its accounts, including assigned groups (the "Account" report)
  - Its groups, including account members (the "Group" report)

DirX Identity Provisioning provides templates for each type of report in the following locations:

- In the Reports folder within the domain configuration subtree of the domain root
- In the Reports folder within the Configuration subtree of each target system

You (the domain administrator) are permitted to customize these report templates to the requirements of your domain. For example, you can change the search base parameter for

a report type to reduce the volume of information generated, or you can select different object attributes to be output for the report. You can also create your own report types. The *DirX Identity Customization Guide* provides more information about how to customize the standard DirX Identity Provisioning reports and how to create your own report types.

You can use DirX Identity Provisioning to make your customizations directly to the templates in the Identity Store (Provisioning Configuration), or you can export the template data to a file in the file system, and use an XML editor to edit it in the file system. You must then import the file back into the Identity Store (Provisioning Configuration) (with DirX Identity Provisioning) to update the templates.

We recommend that you follow this procedure when customizing the standard report templates:

1. Create new report template objects
2. Copy the contents of the standard templates into the new objects
3. Customize the new objects

14.1.8. Relationship to the Connectivity Domain

DirX Identity is designed for multi-tenant operation within a single domain or by setting up multiple domains. The following figure illustrates the multi-domain scenario:

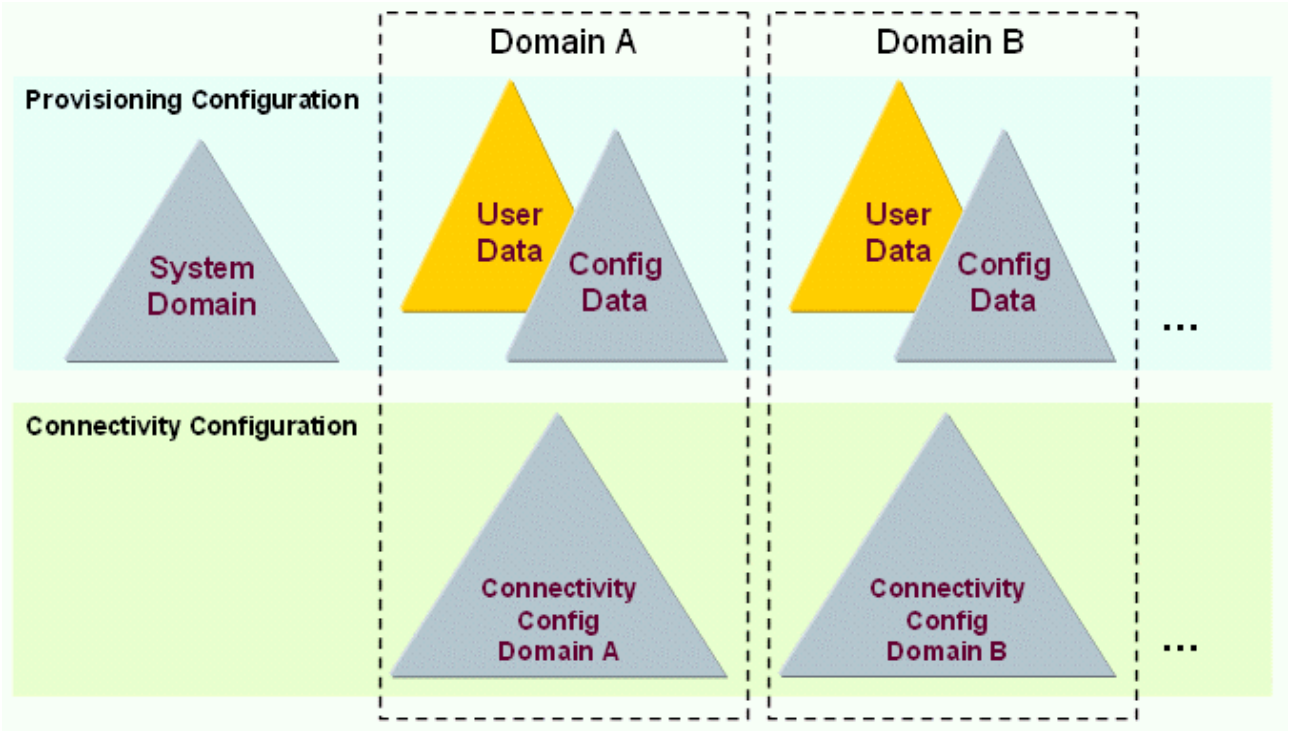


Figure 59. DirX Identity Domain Architecture

As shown in the figure, the **Provisioning Configuration** consists of a **System Domain** and one or more domains (called customer domains). All domains are located in one physical directory server. The System Domain contains all templates for target systems and other non-domain specific data. The customer domains contain clearly separated data that is not accessible from other domains. It contains user (identity) data as well as configuration data.

Configuration data consists of, for example, the role structure, policies and domain data.

Each customer domain in the Provisioning view is related to a Connectivity domain in the **Connectivity Configuration**. The connectivity domain contains all data necessary for synchronization from source systems and to target systems, such as IP addresses, bind profiles and other system-specific data. To separate this data between domains, we recommend separating it into different directory servers.

If you run only one domain, you can keep the Provisioning Configuration and the Connectivity Configuration in one server. If you intend to run several domains, you can keep the Provisioning Configuration and the Connectivity Configuration for one domain in one server. For an additional domain, you need an extra server for Connectivity Configuration data.

To allow separation of the domain parts, the relationship between the Provisioning Configuration and the Connectivity Configuration must be defined in the configuration data. The following figure illustrates this relationship:

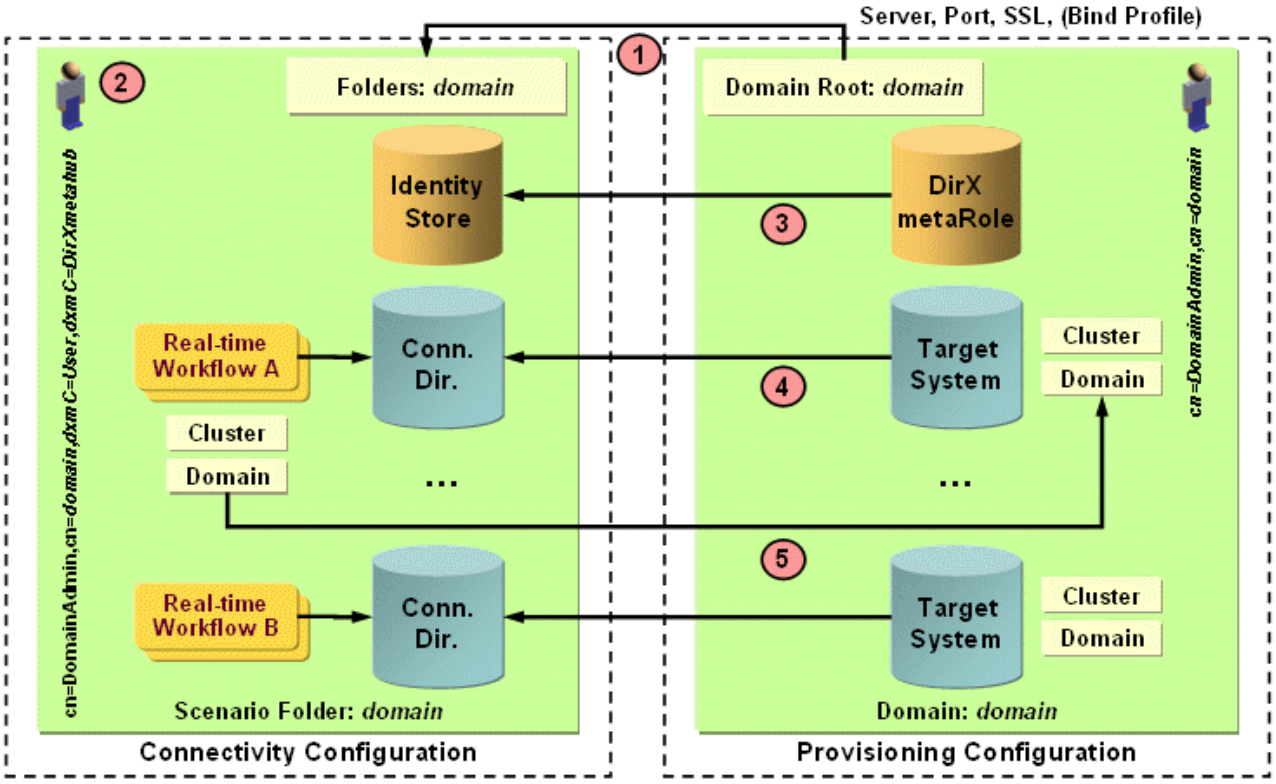


Figure 60. Relationship between Connectivity and Provisioning Domains

The relationship between the domain parts is as follows:

1. The **Domain** object of each Provisioning Domain points to the corresponding Connectivity Domain. See the **Connectivity Configuration Database** definition that defines the **Server**, **Port** and **SSL** parameters.



The target system wizard creates domain specific folders (with the name of the domain) where necessary in the Connectivity Configuration, for example under Connected Directories, Jobs,

Workflows and Schedules. This eases automatic management by the target system wizard and also other components like the Java-based Server. Do not change these folder structures by hand!

2. The software uses the user cn=DomainAdmin,cn=*domain* as the bind account in the Provisioning Configuration. For the Connectivity Configuration, the software uses another fixed user that is located in this domain: cn=DomainAdmin,cn=*domainName*,dxmC=Users,dxmC=DirXmetahub



If you change the password of the DomainAdmin, don't forget to change the password of this entry accordingly. See the section "Managing Administrative Accounts" in the *DirX Identity Connectivity Administration Guide* for more information.

3. The **DirXmetaRole** target system object on the Provisioning side points to the corresponding **Identity Store** on the Connectivity side. This relationship is defined via the **Connected Directory** field in the **General** tab of the target system definition.
4. Each **target system object** on the Provisioning side points to the corresponding **connected directory** on the Connectivity side.This relationship is defined via the **Connected Directory** field in the **General** tab of the target system definition.
5. Real-time synchronization workflows contain a **Type**, **Cluster** and **Domain** field that must be identical to the definition of the corresponding fields at the target system object.

All these relationships are handled by the target system wizard on the Provisioning side as well as during copying real-time workflows.You can also adjust and change the data manually.

Additional Hints

- The scenario object in the Connectivity Configuration contains a link to the Identity Store that must be set correctly.Otherwise, the target system wizard does not work.
- You can also set up multiple tenants within a single Provisioning domain; see the DirX Identity Use Case Document *Using Domains* for details.

14.2. Using Object Collections

Object collections are an easy way to define a set of objects that you want to transfer between DirX Identity Provisioning domains.Object collections allow you to define a set of objects to be frequently exchanged between domains; for example, between a test environment and a productive environment or only once between two different domains.

For more information, see the chapter "Transporting Data" in the *DirX Identity User Interfaces Guide*.

14.3. Monitoring the Workflows

You use the DirX Identity Manager (Connectivity) to monitor the workflows described in section "Default Connectivity Applications" in the *DirX Identity Application Development Guide*.The workflows generate status entries into a status area within the Identity Store.You can view these entries using DirX Identity Manager (Connectivity) Monitor View.

A domain administrator is responsible for handling the following errors:

- DirX Identity Connectivity failed to start either the entire workflow or a single activity within the workflow.
- The workflow or an activity within it failed to complete.
- An activity failed to add, modify, or delete an object or an object attribute.

The administrator can take one of the following actions, depending on the information contained in the error report:

- Do nothing, and wait for the next scheduled workflow run. This action especially makes sense if the problem cause was a temporary error.
- Start the workflow by hand with DirX Identity Manager (Connectivity).
- Try to repair data inconsistencies using DirX Identity Manager (Provisioning) or the LDAP directory tools.
- Request that the responsible administrator corrects the problem.

15. Managing the Provisioning System

Managing the DirX Identity Provisioning System consists of the following tasks:

- Installing and uninstalling DirX Identity. See the *DirX Identity Installation Guide* for more information.
- Configuring DirX Identity Provisioning. See the *DirX Identity Installation Guide* for more information.
- Setting LDAP limits
- Starting the DirX Identity Manager
- Monitoring the provisioning system
- Tuning the provisioning system

15.1. Logging in To DirX Identity Manager

Perform the following steps to log into DirX Identity Manager (Provisioning).The following steps show how to log into the Sample Domain.If you intend to log into another domain, use your domain name wherever **My-Company** is displayed in this procedure:

- Select **Start → Programs → DirX Identity V** *x.x* **→ Manager**.
- Click the **Provisioning** view group.
- In the **Server** field, click the down arrow and select **My-Company** from the list. Ensure that **User DN** is set to **cn=DomainAdmin,cn=My-Company**. If it is not, click the down arrow and select it from the list.
- If **My-Company** does not appear in the **Server** list, click ... to open the **Manage Server Profiles** dialog. Click **New** to create a new server profile. Enter the following values into the fields in the **Server** dialog:

  Name = "My-Company with Domain Admin"
  Description = "Profile for My-Company to enter with Domain Admin account"
  Host = "localhost"
  Port = "389"
  BaseDN = "cn=My-Company"
  DefaultUserDN = "cn=DomainAdmin,cn=My-Company"

- Click **OK** to store the profile. Click **Close** to close the profile list window.
- Select **My-Company** in the **Server** field. Be sure that the **User DN** field is set to **cn=DomainAdmin,cn=My-Company**. If not, select it from the list.
- Enter the password **dirx**, and click **OK**.

After a few seconds, the DirX Identity Manager displays its views window.

Manager attempts to log in to the directory server shown in the **Server** field of the login dialog.If the login operation is successful, Manager closes the login dialog and displays its main window. If the login operation fails, Manager displays an error message and places

the cursor in the password field.

You can click **Cancel** to cancel the login procedure.Manager closes the login dialog and opens an empty main window.You can perform a new login sequence if you change the view or by performing 'Login' at the top level node of the current view.

For details how to customize the login dialog, to create or maintain server profiles see the section "Basic Patterns" in the DirX Identity online help (click **Help** in the login dialog).

15.2. Setting LDAP Client Limits

A **size limit** specifies the maximum number of entries the LDAP server is supposed to return in a search operation.A value of 0 means that the number of entries is unlimited.

A **time limit** specifies the maximum time (in seconds) the LDAP server is supposed to take for processing a search operation.A value of 0 means that the time is unlimited.

Size and time limits are used with the following operations:

- Displaying objects in tree
- Displaying assignable objects (an incomplete result will be indicated)
- Displaying senior roles (an incomplete result will not be indicated)

The following operations are performed as unlimited operations:

- Searching for assigned objects (for example, the accounts that are assigned to a user, the groups of which an account is a member).

Warning: Please be aware that the LDAP server may have global size and time restrictions for searches that DirX Identity Manager can’t exceed. This condition could result in severe configuration database inconsistencies.

The agents do not use any size and time limits. If the LDAP directory server supports the controls “paged read” or “virtual list”, they perform paged reading when searching the LDAP directory. Otherwise, they return to standard search requests. Be sure that in this case the size and time limit of the directory server is set correctly, or the DirX Identity agents could encounter severe problems.

You can set LDAP client size and time limits in the bind profile, the DirX Identity Manager INI file, and in DirX Identity Provisioning domain settings. The next sections describe the settings you can make.

15.2.1. Setting Limits in the Bind Profile

You can set size and time limits in each bind profile that you use during start-up of the DirX Identity Manager. You can also access these settings via the Login command on a tree node in one of the DirX Identity Provisioning views or in the Data View. See the online help topics for details.

These settings have the highest priority.

15.2.2. Setting Limits in the DirX Identity Manager INI File

You can set the parameters ldap.maxresults and ldap.servertimelimit in the DirX Identity Manager INI file.See the section "dxi.cfg" in the *DirX Identity Connectivity Administration Guide* for details.

The settings have the second-highest priority.

15.2.3. Setting Limits in the DirX Identity Provisioning Domain

The DirX Identity Provisioning domain settings are the settings with the lowest priority.See the "Domain Properties" online help topic for details.Modifying size and time limits in the Options tab has immediate effect for the current Manager session.

15.3. Managing Consistency

This section explains how to make sure that objects in the Provisioning domain are consistent and that users have the correct access rights according their manually and automatically assigned privileges. These tasks include:

- Applying consistency rules on all objects
- Marking users that are affected by changes in the role hierarchy as "to be analyzed"
- Performing privilege resolution
- Performing object cleanup

The next sections describe the DirX Identity workflows that perform these tasks and how to configure them.

15.3.1. Using the Consistency Check Workflow

DirX Identity provides a Check Consistency workflow that applies consistency rules and hard-coded consistency checks to the objects of a Provisioning domain.

The workflow's configuration parameters include:

- Enabling the checking of users, roles and permissions, accounts and groups.
- Defining a search base and filter for each set of objects to further limit the set of objects to be checked.
- Activating the application of consistency rules and specifying a search base and filter to select only a subset of rules.
- Enabling the check for privileges to be deleted. With this process, the workflow searches for roles, permissions, and groups in state TBDEL. For each detected privilege, the workflow:
  - Removes the incoming assignments from users and/or senior privileges and sets the To Be Analyzed (dxrTBA) flag for the affected senior objects.
  - Deletes roles/permissions or sets their state to DELETED if history is configured.

- Sets the state of groups to DELETED.

Because checking all objects one after another can take a long time, you can set up several instances of this workflow to process different sets of objects and rules. This setup allows the workflows to run in parallel and on multiple IdS-J Servers on different systems. For example, you can perform hard-coded checks on users, roles/ permissions, and accounts/groups in parallel. It is even possible to run one workflow per target system. If the number of users and accounts is very high and there are enough systems and CPUs, the search base and filter settings allow you to further split up the set of objects and run even more workflows in parallel.



This workflow does not perform user-privilege resolution. If it changes an attribute that might result in the change of assigned groups for a user, the workflow only sets the dxrTBA flag to indicate that the user needs resolution and sends a resolve event to trigger the resolution in the Resolution Adapter. You should check the query folders in Identity Manager regularly to find the problems reported by this workflow.

15.3.2. Using the Mark Affected Users Workflow

When the role hierarchy is changed - for example, a permission is added to a role or group is removed from a permission - or matching rules in roles and permissions are changed, all affected users must be resolved to calculate their updated set of groups. If a DirX Identity component such as the Business User Interface, Web Center, Identity Manager or the Provisioning services performs these changes, they normally set the dxrTBA flag of the changed objects to **true**.

The task of the Mark Affected Users workflow is to find such privileges with **dxrTBA = true**, find the users that might be affected by a change and set the **dxrTBA = true** flag for these users. The User Resolution workflow will then find these so-called “TBA-users” and send the resolve event to the Resolution Adapter. Therefore, it is best to run this workflow before User Resolution.

The Mark Affected Users workflow cannot be split up and run several instances in parallel.

15.3.3. Using the User Resolution Workflow

The User Resolution workflow:

- Applies provisioning rules to users
- Enables or removes user-privilege assignments where start or end dates are reached
- Finds users where the state has to be changed because one of the corresponding start or end dates has been changed
- Triggers user-privilege resolution for all matching users.

Configuration parameters allow you to limit the set of users and rules to be evaluated.

For all users that match the configured search base and filter, the workflow evaluates the provisioning rules in the given subtree for rules. When needed, it adds or removes

automatic privilege assignments according to the rules and the user attributes. It searches for user-privilege assignments that are either outdated (end date in the past) or need to be activated (start date reached). It searches for users where one of the start and end dates is reached and then changes their states accordingly. Finally, it sends a resolution message for each of them so that the Resolution Adapter resolves these users.



When the user filter is empty, the workflow sends resolve messages only for the users with **dxrTBA = true**.

15.3.4. Setting up Object Cleanup

Most of the DirX Identity objects are only flagged for deletion and are not immediately removed. Set up the according services to remove these objects from time to time. We recommend setting up these services to run every week. You are allowed to leave all objects in the directory, but doing this can decrease performance and reduce oversight.

The most important services for deletion are consistency checking, and deletion consistency rules. The next sections provide more detail about these services.

15.3.4.1. Consistency Checking

The Consistency Check workflow performs the following delete actions for the objects that match its filter:

- Deletes all users that are in state TBDEL and whose Delete Dates have expired.
- Deletes all accounts that are marked with **Managed in Target System Only** that are in state **Deleted** in the target system.
- Deletes accounts that are in state DELETED and whose End Dates have expired.
- Deletes all groups that are marked for **Managed in Target System Only** that are in state **Deleted** in the target system.
- Deletes groups that are in state DELETED or DISABLED and whose End Dates have expired.
- Deletes all assignment objects whose End Dates have expired.

15.3.4.2. Mark Affected Users

The Mark Affected Users workflow finds users that might be affected of changed privileges. (See section "Mark Affected Users" for details.)

15.3.4.3. Object Cleanup with Deletion Consistency Rules

DirX Identity is delivered with pre-configured consistency rules for the deletion of objects. The default workflow **CleanupObjects** executes these rules:

- **cleanupDeletedObjects**
  Handles the deletion of a huge set of object types.You can change the rule or extend it to handle your object type extensions.
- **cleanupOutDatedDelegations**

Removes outdated delegations.

- **cleanupOutDatedRequWflowInstances**
  Deletes outdated request workflow instances.

You can set up separate workflows to run any combination of these rules.

15.4. Monitoring DirX Identity Provisioning

The topics in this section describe:

- How to detect and handle inconsistencies in the DirX Identity Provisioning system
- How to use DirX Identity Provisioning logging to monitor the system

15.4.1. Detecting and Handling Inconsistencies in DirX Identity Provisioning

The Identity Store (Provisioning views group) is a complex matrix of inter-connected objects: users - roles - roles - permissions - groups - accounts - users.These objects are modified by such diverse applications as:

- One or more DirX Identity Manager installations
- One or more DirX Identity Web Center installations
- The regularly scheduled creation workflows
- Each target system's provisioning (synchronization and validation) workflows

DirX Identity Provisioning general function relies on the ability of a set of distributed applications to work together. The overall operation of this distributed system can be disrupted by some malfunction of its integral components or by another LDAP application.

As a result, DirX Identity Provisioning supplies the following mechanisms to help DirX Identity Provisioning administrators identify Provisioning Configuration inconsistencies and repair them quickly:

- Automatic detection of inconsistencies within DirX Identity Provisioning components
- Status report generation templates and workflows
- Query folders

15.4.1.1. How DirX Identity Provisioning Detects Inconsistencies

Inconsistencies are detected by

- DirX Identity Manager when a Provisioning object is changed
- The regularly running consistency workflow performed by service agent
- The validation workflows for each target system

15.4.1.2. How DirX Identity Manager Detects Inconsistencies

DirX Identity Manager (Provisioning views group) detects inconsistencies when the administrator modifies an object, as follows:

- When a user is modified, Manager re-calculates the access rights and thereby navigates the chain of references from the user to his assigned roles, permissions, groups and accounts. In case of errors (broken references: the destination of a link is not an existing object or an object of wrong type), Manager informs the administrator in a message box. If the administrator chooses to store the modifications despite the errors, Manager does not store the new – and presumably invalid – access rights, but flags the user as invalid ("dxrIsInconsistent" = true), stores the error message in "dxrError" and sets the "dxrErrorExpDate". The new access rights are at the latest enforced at this expiration date. This is one of the tasks of the consistency workflow. In the meantime, the administrator can repair the reason for the error and resolve the user anew.
- Manager not only issues error messages, but also warnings. A typical reason for a warning is when the user is granted a permission, but none of the assigned groups, because the match rule fails.
- When a role, permission or group is edited, Manager performs some local checks, but by default does not re-calculate the access rights of the affected users. This operation is postponed until the next consistency workflow runs. Local checks especially include valid match rules and the links to the next descendants.

15.4.1.3. How the Domain Consistency Workflow Detects Inconsistencies

The domain service (consistency) workflow especially checks for:

- Broken references (the destination of a link is not an existing object or an object of wrong type).
- Error expiration date in user objects. In this case, it performs a privilege resolution and stores the result regardless of whether there are any errors.
- Start dates of users that have been created but not yet activated (users in the NEW state).
- End dates of users (the end of the user's lifetime).
- Start deactivation dates of users.
- End deactivation dates of users.
- End dates of users or accounts to be deleted. The objects are deleted.
- End dates of disabled accounts: they are marked for deletion (set to state DELETED) and the new end date is calculated according the MaxTimeToDelete domain parameter.
- Invalid roles or permissions: no descendants (neither junior role, permission or group assigned), invalid match rule.
- Out-dated user-to-role, user-to-permission or user-to-group assignments, which are de-activated.
- Modified roles, permissions and groups. The workflow searches for the affected users and performs a privilege resolution for each of them. This task can be very time-

consuming and is therefore not performed in the Manager.

- Accounts without any user assigned that are not marked as "target system local".

15.4.1.4. How the Consistency and Validation Workflows Detect Inconsistencies

When the consistency workflow recognizes inconsistencies it cannot repair, it stores appropriate instructions for the administrator in the dxrToDo attribute of the affected object.

The validation workflows automatically check for:

- Accounts or groups that have been deleted in the target system
- Accounts or groups that have been added to the target system
- Group-account memberships that have been added or deleted in the target system

Like the consistency workflow, the validation workflows store such deviations with appropriate messages in the dxrToDo attribute of the affected account or group object.

15.4.1.5. Using Status Reports to Detect Inconsistencies

DirX Identity Provisioning provides a set of standard status report templates and a set of status report generation workflows that domain administrators and target system administrators use to generate periodic status reports about the Provisioning objects that they are responsible for managing. The topic "Setting up the Status Report Templates" describes the DirX Identity Provisioning-provided report types. Administrators can also create their own status report types; see the *DirX Identity Customization Guide* for further details.

Administrators should generate status reports on a regular basis. These reports can help them to detect strange combinations because they offer an overview across a number of objects at a glance. Administrators can fine-tune the status report templates so that they extract only those objects in a specified state or with specified attribute value combinations; see the *DirX Identity Customization Guide* for further details

15.4.1.6. Using Query Folders to Detect Inconsistencies

DirX Identity Provisioning provides a default set of query folders in each DirX Identity Provisioning view and each target system that you can use to search for and identify objects with:

- Error or ToDo entries
- Inconsistent users
- Users to be deleted
- Unassigned accounts (accounts without any user assigned)

When you open a query folder, it performs a search on the DirX Identity store (Provisioning Configuration) according to the values in its Filter tab and displays the results as objects underneath the query folder.

You can easily create new query folders or modify the default folders to deliver object sets that meet special conditions and thus highlight exceptional objects in a prominent location.

15.4.1.7. Provisioning Object Attributes that Identify Inconsistencies

Objects like user, role, permission, account or group have "dxrError" and "dxrToDo" attributes. They inform you about errors that have occurred and of actions you need to perform. The "dxrState" and "dxrTsState" attributes in the account and group objects allow you to compare the states in DirX Identity Provisioning and in the target system. Especially important is the "dxrIsInconsistent" attribute of the user, which indicates users where assigned roles are not consistent with the calculated access rights (see below).

15.4.1.8. Handling Detected Inconsistencies

DirX Identity Provisioning cannot automatically repair all detected anomalies. DirX Identity Provisioning does a lot to relieve the administrator of this burden by enforcing start and end dates, enforcing access rights of inconsistent users, reapplying deleted group–account assignments and so on. But determining how to correct a broken reference, or how to repair a match rule with invalid permission parameters (for example, delete it?) requires the knowledge of a human administrator.

Consequently, the DirX Identity Provisioning administrator(s) must watch for exceptional situations on a regular basis using the standard (or customized) query folders or reports.

Administrators must delete broken references and objects with the wrong object and must then re-create them. By default, the object class is not displayed and editable in DirX Identity Manager. The administrator can customize the object and property description to allow him to make such changes with Manager or he can use another LDAP client provided with the database to edit these attributes by hand. In the case of assignments, this requires detailed knowledge of the way in which DirX Identity Provisioning stores them in the directory.

DirX Identity Provisioning deactivates outdated assignments, but does not remove them. This gives the administrator the opportunity to be informed of the existence of past assignments and extend them easily, but leaves him with the task of manually removing them.

Matching rules that use attributes that are no longer specified as permission or role parameters remain active, which enables a smooth migration. On the other hand, they carry the danger of security breaches. The role administrator must decide when to modify the rule and enforce the permission parameter changes. The reports can help to determine which users are affected by this permission or role and optionally have to care for proper attribute values.

Special care must be taken of the validation results stored in the account and group objects. Objects deleted in the target system must be deleted manually in DirX Identity Provisioning: the same objects normally cannot be re-created. If accounts or group assignments are needed, DirX Identity Provisioning automatically creates new ones when the corresponding user is resolved. But groups must be created by the administrator manually. In any case, these should be an issue for the DirX Identity Provisioning target

system administrator and the local target system administrators.

15.4.2. DirX Identity Provisioning Logging

The following DirX Identity Provisioning components perform logging functions:

- DirX Identity Manager
- DirX Identity Service Agent

15.4.2.1. DirX Identity Manager Logging

DirX Identity Manager trace logging is controlled by keys in the file *install_path***gui\bin\dxi.cfg**.

See the section "Using DirX Identity Manager → Customizing DirX Identity Manager → dxi.cfg" in the *DirX Identity Connectivity Administration Guide* to understand how to configure the manager logging.

If you don't specify anything, error messages are written into the file **system**.*nnn*.**log**.

On start-up, DirX Identity Manager removes existing *prefix*.*nnn*.**log** files to prevent inconsistencies between old and new trace messages.

15.4.2.2. DirX Identity Service Agent Logging

DirX Identity service agent logging is controlled via its initialization (**ini)** file. See the section "About the Maintenance Agent INI File" in the chapter "Managing the DirX Identity Provisioning System" for information about this file.

15.5. Tuning the Provisioning System

DirX Identity contains a lot of comfortable features and concepts that consume a lot of computer power.

The user interface (UI) applications (DirX Identity Manager, Business User Interface (BUI), Web Center) display complex information for users that are composed of the content of many objects (users, roles, permissions, groups, accounts). When assigning and resolving privileges, the role hierarchy must be navigated, role and permission parameters evaluated, SoD must be checked, sometimes approval workflows must be started or tickets created, and the real-time provisioning actions must be triggered.The time for performing all these actions is highly dependent on the number of objects (users, roles, permissions, groups), the use of parameters (role and permission parameters) and on the complexity of structures (role hierarchies, hierarchical group structures).

To minimize the time for an end user waiting in front of a UI application, assigning a privilege and resolving the assignments to the access rights in the target systems have been split. The client, namely the UI applications, only perform what is necessary before storing the requested changes: check consistency of input data, check SoD, start approval workflows and create tickets where necessary. If privilege resolution is necessary, they send a resolve message that is processed by a Resolution Adapter in an IdS-J server

asynchronously in the background.This Resolution Adapter resolves the users, calculates the new access rights, and triggers the real-time provisioning.

Due to the variety of possible combinations of all the relevant parameters, there is no general approach that guarantees acceptable performance.To optimize your system, you can adjust some parameters, switch off some of the features or move intensive calculations to the background.

This section gives some hints on how you can tune the system for a specific customer environment. Read it carefully and experiment with the various settings to adapt the system for optimum performance. Identity Manager, BUI and Web Center are controlled by these settings.

15.5.1. Increasing the Number of Resolution Adapter Threads

Each IdS-J Server hosts a Resolution Adapter. It receives messages to perform privilege resolution for a selected user from the queue **dxm.request.user.resolve**. By default, the adapter starts two listeners. This means with *n* Java-based servers per domain, *n* * 2 different users can be resolved at the same time.

To resolve more users in parallel, you can increase the number of listeners per server:

- In DirX Identity Manager, select the Provisioning view group and then the Domain Configuration view.
- Select the top-level object and click the Privilege Resolution tab.
- Set the number of resolution listeners per IdS-J Server in the respective field; for example, 3.
- Activate the changes by restarting every IdS-J Server.

15.5.2. Tuning the User Edit Operation

To detect inconsistencies in a user entry, DirX Identity Manager must read all tabs of a user entry when **Edit** is clicked. If inconsistencies are detected, the relevant tabs are flagged and the user must correct them, otherwise the **Save** button will not be enabled.

It is important that roles, permissions, groups, and accounts are read with all relevant attributes during the first search operation. Otherwise, time-consuming re-reading of entries can occur. This behavior can be controlled by the LoadAttributes section in the **User.xml** object descriptions.

15.5.2.1. Changing the Customer-specific User.xml Object Description

If you have extended the role, permission, group or account object in your environment, perform these steps:

- Open Domain Configuration → Customer Extensions → Object Descriptions.
- Select **User.xml** and click **Edit**.
- Enter a section like this one:

```
<extension>
<loadAttributes>
<roles name="{<your list of extension attributes>}" />
<permissions  name="{<your list of extension attributes>}" />
<groups name="{<your list of extension attributes>}" />
<accounts name="{<your list of extension attributes>}" />
</loadAttributes>
</extension>
```

- Save the object.
- Restart the Manager.

This list will be merged with the list in the previous section.

To test for unnecessary re-reads:

- Set the **trace.level** in the file **dxi.cfg** to 9 (remember the previous value).
- Restart the Manager.
- Click a typical user. Click **Edit**.
- Open the file **system.000.log** in the folder *install_path***gui\log**. There will be messages like this one:
  DBG(STG100): complete o=MultiMarket,dc=Customers,cn=Users,cn=My-Company (reason: o)
- Search for the string **reason:**.
- Check for multiple occurrences of the same string (for example, during read of roles, permissions, groups). If you find them, enter this attribute to the loadAttributes section.
- Restart the Manager.
- Repeat the test until multiple occurrences are no longer visible.
- Reset the **trace.level** in the file **dxi.cfg**,
- Restart the Manager.

15.5.3. Tuning the Tree Operation

Configure the attributes being requested when searching the node's children to optimize the tree expansion. Define the requested attributes in the node's object description in the <children> tag of the <loadAttributes> extension tag.

To optimize expansion of a tree node, run the identity manager with trace level 9 and observe DBG - messages 'complete *object_dn* (reason: *attribute name*)' in the identity manager's system log that are reported when the tree is expanded. In case those messages occur, add the *attribute name* reported in the message to the load attributes.

Example:

When expanding an organizational unit (or an organization), the following messages can occur:

```
DBG(STG001): complete <cn=aUser>,cn=Users,cn=My-Company (reason:
employeenumber)
```

To avoid reading each user as a whole, when the tree is expanded, just add employeeNumber to the load attributes for searching children in Org.xml and in OrgUnit.xml:

```
<extension>
<loadAttributes>
  <children name="{ou,sn,givenName,dxrState,employeeNumber}" />
</loadAttributes>
</extension>
```



Be sure that you observe the messages resulting from tree expansion.

- Start DirX Identity Manager with trace level 9.
- Expand the tree node to be optimized.
- Quit the Manager and check the messages. Only messages about the objects in the expanded tree are relevant.
- Repeat this procedure until no more "complete" messages occur.



It is possible to use a JavaScript to create the names displayed in the tree. However, this method does not perform well when dealing with thousands of children in one node. A better strategy is to fill an attribute "displayName" at the user being maintained with each user edit and to display this attribute.



The attribute must be added to the load attributes as described above.

15.5.4. Managing Long Delays

Clicking **Edit** for a user object in DirX Identity Manager or selecting **Assign Privileges** in the Web Center can cause long delays if the switch "assign.InitialSearch=true" is set in the **dxi.cfg** file. The delays result from reading a large number of privileges to populate the lists for the privileges to assign. Each of these entries must be checked by an access control operation.

By default, the lists are empty. The user must perform a search operation by defining the filter and clicking the corresponding button. This action can significantly affect performance depending on the number of existing privileges and the complexity of access

policies in the configuration.

This behavior is enforced in the Web Center and is the default for the Manager.

15.5.5. Ignoring Nested Group Searches

By default, DirX Identity searches for nested groups in each target system during resolution and check processes. This requires additional time especially with large groups and thus can lower performance significantly.

For target systems that do not use nested groups, you can set a flag to switch off these unnecessary searches:

- Start the DirX Identity Manager.
- Login to the **Provisioning View**.
- Check the **Ignore Nested groups** check box in the **Advanced** tab of the target system.
- Apply this setting to each target system that does not require nested groups.
- To activate the changes in the Manager, restart it.
- To activate the changes in the Web Center, restart Tomcat.

15.5.6. Activating XXL Groups

A search for assigned groups might result in low performance if a target system has only a few groups with a high number of members (this limit depends on the directory server technology in use).

You can activate the XXL group flag at all target systems that meet this condition:

- Start the DirX Identity Manager.
- For each target system, select the option **XXL Groups** in the **Advanced** tab.
- To activate the changes in the Manager, restart it.
- To activate the changes in the Web Center, restart Tomcat.

Appendix A: Context-Sensitive Help

- Users View
- Business Objects View
- Ticket View
- Privileges View
- Policies View
- Certification Campaign View
- Request Workflows View
- Target Systems View
- Auditing View
- Domain Configuration View
- Miscellaneous

A.1. Users View

The **Users** view shows the tree of user entries together with their organizational folders like countries, localities, domain components, organizations and organizational units.It may also hold a set of query folders for filtering a set of user entries according to various criteria.

For each selected node in the tree, the respective property pages are shown on the right-hand side.Click **Edit** to modify property values.Note that saving the properties of user entries will result in another resolution of their relations to groups and roles.

Related Topics

- User
- Country
- Locality
- Domain Component
- Organization
- Organizational Unit

A.1.1. Users

This tab shows the general properties for a user folder. The user folder contains either user entries or ordering folders corresponding to countries, localities, domain components, organizations or organizational units (departments, teams).

The items shown here are the following:

Name - the displayed name of the users folder.

Description - the description for the users folder.

Related Topics

- Country
- Locality
- Domain component
- Organization
- Organizational Unit

A.1.1.1. User - General Properties

The user object represents a user of an IT infrastructure. DirX Identity Provisioning manages the user's access to all parts of the IT infrastructure.

Use this tab to display the general properties of the user object. Since it is fully configurable by the domain administrator, the items shown here may vary from installation to installation or even from domain to domain. It is also possible to set up several user types that show a different set of properties.

The following properties explain the default configuration of DirX Identity:

General (Identification) Properties

Name (mandatory) - used as the relative distinguished name (RDN) for the user entry and identifies the user uniquely within the sub-tree. Once set during adding a new user entry, it can no longer be changed by simply editing this field after the initial save operation. Use the **Rename** command of the user entry's context menu instead. A change of the common name is also possible by running the user integration workflow.

First Name (optional) - the first name of the user.

Middle Name (optional) - the middle name of the user.

Salutation (optional) - the salutation of the user. Values for different countries are available.

Title (optional) - the title of this user.

Last Name (mandatory) - the last name of the user. This item corresponds to the "surname" attribute saved in the Identity Store.

Gender (optional) - the gender of this user.

Day of birth (optional) - the day of birth for this user.

Master (optional) - the name of the directory that masters the user entry.

Employee number (optional) - the employee number.

Identifier (optional) - the user's global unique identifier.

Description (optional) - a description for the user. This field is often useful when user entries are listed in tables with only their names but with a description column to better identify them.

Employee Type (optional) - one of the following user's employee types:

- **Contractor** - an (outside) contractor.
- **Customer** - a customer.
- **Internal** - an internal employee.
- **Supplier** - a supplier.

Business Category (optional) - the business the user works for.

Related Topics

- Multi-Value Editor
- User
- Relationships
- Operational Properties
- Communication
- Authentication
- Organization
- Location
- Context
- Assigned Roles
- Assigned Permissions
- Assigned Groups
- Accounts
- Orders
- SoD Exceptions
- Risk Parameters

A.1.1.2. User - Relationships

Use this tab to define relationships to other user objects.

Owner (optional) - the owner of the user entry. You can use this field in two ways:

- To define the user administrator that can handle this user (set up an access policy to make this work).
- To define the responsible user for a persona (see the section "Handling Multiple Accounts" in the chapter "Managing Target Systems" for more information).

Manager (optional) - the user's manager. Used for user type Internal Employee.

Secretary (optional) - the user's secretary.

Representative (optional - the user's representative.

Sponsor (optional) - the user's sponsor. Used mainly for user type Contractor.

Note that these links are very useful when setting up participants for request workflows. For more information see the section "Participant Calculation" in the *DirX Identity Application Development Guide*.

Related Topics

- User
- Operational Properties
- Communication
- Authentication
- Organization
- Location
- Context
- Assigned Roles
- Assigned Permissions
- Assigned Groups
- Accounts
- Orders
- SoD Exceptions
- Risk Parameters

A.1.1.3. User - Operational

Use this tab to display the operational parameters necessary for correct operation on this user within DirX Identity.

Status fields

To be analyzed (read-only) - indicates when checked that this entry has been changed. For example, an import workflow added or modified some attributes. The privilege resolution or the consistency checker uses this flag to detect and resolve records that are not up-to-date. After the check or resolution operation completes, the flag is cleared.

Is Inconsistent - indicates whether (unchecked) or not (checked) the stored access rights of the user are consistent with the assigned privileges. DirX Identity Provisioning checks the flag (sets the entry's flag to TRUE), if it fails to resolve the assigned privileges due to errors in the privilege structure. In this case, the user keeps the previous access rights, but DirX Identity Provisioning sets an end date for this state: the **Error expiration date**. The

resolution error is stored in the **Error** field. So the administrator has time to repair the error. The flag is unchecked if the entry is consistent.

Use as Template - indicates when checked that DirX Identity will not resolve privileges assigned to this user. A likely reason for the checked flag is that the user object has been copied from another object. In this case, the user is marked as a template until appropriate changes have been made to the copied user. If the user is set correctly, uncheck the flag to enforce privilege resolution and target system provisioning.

+ A template user may also have been deliberately created to be used as a basis for creating additional accounts that are fully controlled by DirX Identity. You can use the owner link to point from this copied user to another (primary) user. This reference can help you to understand which secondary user entries belong to a primary one.

Status - shows the status of the user entry. For more information, see the section "Managing States" in the chapter "Managing Provisioning". Possible values are:

- **TEMPLATE -** the settings made for this user are used as a template.
- **NEW -** the user is a new user, the privilege assignments are not activated yet.
- **ENABLED** - the user is established; the corresponding user object is up to date and valid, the privilege assignments are active.
- **DISABLED** - the privilege assignments for this user are currently disabled, the user does not have any rights in the domain.
- **TBDEL** - the user object shall be deleted from the DirX Identity Provisioning system. This action is taken when all assigned accounts in the target systems are deleted or the end date for deletion is reached.

Lifetime Start - shows the date on which the privilege assignments to the user will become active for the first time (usually the date on which the user enters the company).

Lifetime End - shows the date on which the privilege assignments to the user will become inactive for the last time (usually the date on which the user leaves the company).

Deactivation Start - shows the date on which the privilege assignments will become inactive for the next time.

Deactivation End - shows the date on which the privilege assignments will be reactivated again for the next time.

Delete - shows the date on which the entry will finally be deleted despite any existing accounts in target systems.

Create time stamp - shows the date and time at which this entry was created.

Modify time stamp - shows the date and time at which this entry was last changed.

Notification fields

Notification Level - controls suppression of the first e-mail of the e-mail notification of an approval workflow. Possible values are:

- **0** - the first e-mail is sent.
- **1** - the first e-mail is not sent (but all other administrative or repetition e-mails).

Tasks fields

To do - shows a list of manual working items for the administrator. DirX Identity Provisioning stores the results of consistency checks it cannot repair automatically. The items are overwritten on the next consistency check.

Possible messages in this field besides warnings from the service agent are:
INF:530:Missing values for attributes: *attributelist*.
*INF:531:More than one peer account exists for account DN=**dn*.
*INF:532:No peer account exists for account DN=**dn*.

Error - stores the list of error messages that give the detailed reason for inconsistent states or **To do** entries. They are cleared automatically when a privilege resolution succeeds.

Error expiration date - shows the date on which the access rights of the user will be updated despite any resolution errors. See the **Is Inconsistent** field.

Related Topics

- User
- Relationships
- Communication
- Authentication
- Organization
- Location
- Context
- Assigned Roles
- Assigned Permissions
- Assigned Groups
- Accounts
- Orders
- SoD Exceptions
- Risk Parameters

A.1.1.4. User - Communication

Use this tab to display communication attributes for a user object.

E-Mail (optional) - the user's e-mail addresses.

Phone (optional) - the user's main phone number (business phone number).

Fax (optional) - the user's fax number.

Mobile (optional) - the user's mobile number.

Home (optional) - the user's home phone number.

Web address (optional) - the Web address of the Internet home page for this user.

Preferred language (optional) - the preferred language of this user, which controls, for example, the language used in an e-mail if this user is referenced in the To field. For more information, see the chapter "Using Variable Substitution" in the *DirX Identity Application Development Guide*.

Note that the phone number might be used by default when DirX Identity Provisioning creates a new account for the user. The account name is generated using the user's surname, given name and phone number. See the object description for the accounts.

Related Topics

- User
- Relationships
- Operational Properties
- Authentication
- Organization
- Location
- Context
- Assigned Roles
- Assigned Permissions
- Assigned Groups
- Accounts
- Orders
- SoD Exceptions
- Risk Parameters

A.1.1.5. User - Authentication

Use this tab to display the authentication properties for this user.

Password Management

One-Way Password (read-only) - the user password (userPassword attribute). Authentication against the Identity Store is performed using this password. The password is usually not readable.

Encrypted Password (read-only) - the encrypted (or scrambled) password of this user (dxmPassword attribute). This attribute is used for password synchronization to target

systems.

Password Changed Time (read-only) - the time at which the password was last changed.

Password Expiration Notified (read-only) - the time at which the Password Notification service notified the system about the password expiration.

Password History (read-only) - the password history (one-way encrypted). This field shows only that there is a password history.

Password Reset (read-only) - if active, the password was reset by the administrator.

Password Policy - a link to the relevant password policy. If no link is set, the default policy is used by the system.

Logins with Password

This group of fields is related to logins with password.

Failure Count (read-only) – the number of failed login attempts. The number is reset on any successful login attempt.

Last Failure Time (read-only) – the time of the last failed login attempt. The time is cleared on any successful login attempt.

Locked Until (read-only) – the time until which login attempts by the user are rejected.

Challenge Response Authentications

This group of fields is related to challenge-response authentications.

Challenges and Responses (read-only) - the challenge and response questions (one-way encrypted). This field shows only that challenge and response questions have been set up.

Failure Count (read-only) – the number of failed authentication attempts. The number is reset on any successful login attempt.

Last Failure Time (read-only) – the time of the last failed authentication attempt. The time is cleared on any successful login attempt.

Locked Until (read-only) – the time until which authentication attempts by the user are rejected.

SASL EXTERNAL Bind

Mirrored User - a link to the mirrored user for SASL EXTERNAL binds. For details, see the chapter "Configure DirX Identity - Provisioning" in the Use Case Document *DXI Smart Card Login Manager*.

Certificate

Certificate (read-only) - the certificate of the user.

Related Topics

- User
- Relationships
- Operational Properties
- Communication
- Organization
- Location
- Context
- Assigned Roles
- Assigned Permissions
- Assigned Groups
- Accounts
- Orders
- SoD Exceptions
- Risk Parameters

A.1.1.6. User - Organization

Use this tab to display the organization and organizational unit properties of a user. These properties can be linked with the relevant business objects.

Organizations

Organization Link (optional) - sets a link to an organization object (business object).

Organization (optional) - defines the organization to which the user belongs. If the **Organization Link** is set, this attribute is read-only and is controlled by the linked object.

More Organizations (optional) - defines more organization links for this user.

Organizational Units

Org. Unit Link (optional) - sets a link to an organizational unit object (business object).

Organizational Unit (optional) - the organizational unit to which the user belongs. If the **Org. Unit Link** is set, this attribute is read-only and controlled by the linked object.

Department Number (optional) - the department number of the organizational unit. If the **Org. Unit Link** is set, this attribute is read-only and controlled by the linked object.

More Organizational Units (optional) - defines more organizational unit links for this user.

Cost Units

Cost Unit (optional) - defines the cost unit to which the user belongs.

More Cost Units (optional) - defines more cost units for this user.

Related Topics

- User
- Relationships
- Operational Properties
- Communication
- Authentication
- Location
- Context
- Assigned Roles
- Assigned Permissions
- Assigned Groups
- Accounts
- Orders
- SoD Exceptions
- Risk Parameters

A.1.1.7. User - Location

User this tab to display and set location information associated with the user.

Location Link (optional) - sets a link to a location object (business object).

Country (optional) - the country of the user. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

State (optional) - the state of the user. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

Postal Code (optional) - the postal code of the location. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

Location (optional) - the location(s) of the user. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

Street (optional) - the street of the location. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

Room (optional) - the room where this user is located.

Postal Address (optional) - the postal address of this location. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

More Locations (optional) - defines more location links for this user.

Related Topics

- User
- Relationships
- Operational Properties
- Communication
- Authentication
- Organization
- Context
- Assigned Roles
- Assigned Permissions
- Assigned Groups
- Accounts
- Orders
- SoD Exceptions
- Risk Parameters

A.1.1.8. User - Context

This tab is an example of other context information. Pre-configured links are:

Contexts (optional) - a link to any type of context object (business object).

Related Topics

- User
- Relationships
- Operational Properties
- Communication
- Authentication
- Organization
- Location
- Assigned Roles
- Assigned Permissions
- Assigned Groups
- Accounts
- Orders
- SoD Exceptions
- Risk Parameters

A.1.1.9. User - Assigned Roles

Use this tab to view and manage the roles assigned to a user. The tab displays the list of roles currently assigned to the user.

For each role, the following properties are shown:

Role - the role's name as it is displayed throughout the DirX Identity Provisioning system. If the role's name ends with an asterisk character (*), the role has assigned one or more junior roles. These junior roles are automatically assigned to the user as well.

Description - the description for the role.

Start Date - the date on which this role became or will become active for the user; that is, the date on which the user was/is granted the permissions assigned to this role.

End Date - the date on which this role became or will become inactive for the user; that is, the date on which the permissions granted to the user was or will be revoked. If the re-approval flag is set, the end date shown is the date of the re-approval. You can change the end date to shorten the period to the next re-approval but you cannot lengthen it.

Reapproval - whether (checked) or not (unchecked) the assignment requires regular re-approval. The next re-approval will occur at the shown end date. Unchecking this flag removes the privilege from re-approval at the shown end date (re-approvals no longer occur).

Role Parameters - shows the first few assigned role parameters. Use the  button to view all role parameter settings for this assignment.

Assigned by - how the assignment was made. Possible values are:

- **BO** - the role was assigned by privilege inheritance from a business object.
- **UF** - the role was assigned by user facet inheritance from a user facet.
- **manual** - the role was assigned by hand.
- **rule** - the role was assigned by a rule.

State - the state of the assignment. Possible values include:

- **ToBeApproved** - the assignment is still in approval. The corresponding access rights (groups) are not yet assigned.
- **ENABLED** - the assignment is enabled. The user has the corresponding access rights (groups) assigned.
- **DELETED** - the assignment is deleted. The corresponding access rights (groups) are not yet or no longer assigned.

  See the section "Managing States" in the chapter "Managing Provisioning" to understand the relationship to the states of other objects.

  Note that DirX Identity shows assignments in approval as gray lines. Valid values for such entries are:

- **Add** - this assignment is not yet active.
- **Mod** - parts of this assignment (some attributes) were changed.
- **Del** - this assignment is to be deleted.

To add or remove roles, use the assignment editor, which is displayed when you click **Edit**. To examine the details of a particular role, click ... to the right of its row.

Note that a line is displayed in gray if this assignment is still in approval. Thus you can see a white line that defines the current state and maybe several gray lines that define changes for this assignment that are still in approval.

Related Topics

- User
- Relationships
- Operational Properties
- Communication
- Authentication
- Organization
- Location
- Context
- Assigned Permissions
- Assigned Groups
- Accounts
- Orders
- SoD Exceptions
- Risk Parameters

A.1.1.10. User - Assigned Permissions

Use this tab to view and manage the permissions assigned to a user. The tab displays the list of permissions currently assigned to the user.

For each permission, the following properties are shown:

Permission - the permission's name as it is displayed throughout the DirX Identity Provisioning system.

Description - the description for the permission.

Start Date - the date on which this permission became or will become active for the user.

End Date - the date on which this permission became or will become inactive for the user. If the re-approval flag is set, the end date shown is the date of the re-approval. You can change the end date to shorten the period to the next re-approval but you cannot

lengthen it.

Reapproval - whether (checked) or not (unchecked) this assignment requires regular re-approval. The next re-approval occurs at the shown end date. If you uncheck this flag, privilege re-approval is removed at the shown end date (re-approvals no longer occur after the end date is reached).

Assigned by - how the permission assignment was made. Possible values are:

- (empty) - the permission is inherited from a role.
- **BO** - the permission was assigned by privilege inheritance from a business object.
- **UF** - the permission was assigned by user facet inheritance from a user facet.
- **manual** - the permission was assigned by hand.
- **rule** - the permission was assigned by a rule.

State - The state of the assignment. Possible values are:

- **ToBeApproved** - the assignment is still in approval. The corresponding access rights (groups) are not yet assigned.
- **INHERITED** - the assignment is inherited from a role assignment.
- **DELETED** - the assignment is deleted. The corresponding access rights (groups) are not yet or no longer assigned.

  See the section "Managing States" in the chapter "Managing Provisioning" to understand the relationship to the states of other objects.

  Note that DirX Identity shows assignments in approval as gray lines. Valid values for such entries are:
- **Add** - the assignment is not yet active.
- **Mod** - parts of the assignment (some attributes) are to be changed.
- **Del** - the assignment is to be deleted.

To add or remove permissions manually, use the assignment editor, which is displayed when you click **Edit**. To inspect the details of a particular permission, click … to the right of its row.

Note that a line is displayed in gray if this assignment is still in approval. Thus you can see a white line that defines the current state and maybe several gray lines that define changes for this assignment that are still in approval.

Related Topics

- User
- Relationships
- Operational Properties
- Communication

- Authentication
- Organization
- Location
- Context
- Assigned Roles
- Assigned Groups
- Accounts
- Orders
- SoD Exceptions
- Risk Parameters

A.1.1.11. User - Assigned Groups

Use this tab to view and manage the groups assigned to a user. The tab displays the list of groups currently assigned to the user.

For each group, the following properties are shown:

Target system - the name of the target system to which the group belongs.

Group - the group's name as it is displayed throughout the DirX Identity Provisioning system.

Description - the description for the group.

Start Date - the date on which this direct group assignment became or will become inactive for the user.

End Date - the date on which this direct group assignment became or will become inactive for the user. If the re-approval flag is set, the end date shown is the date of the re-approval. You can change the end date to shorten the period to the next re-approval but you cannot lengthen it.

Reapproval - whether (checked) or not (unchecked) this assignment will require regular re-approval. The next re-approval will occur at the end date shown. Unchecking this flag removes the privilege from re-approval at the shown end date (re-approvals are no longer required after this date is reached).

Assigned by - how the group assignment was made:

- (empty) - the group was assigned by a permission or role through privilege assignment.
- **BO** - the group was assigned by privilege inheritance from a business object.
- **UF** - the group was assigned by user facet inheritance from a user facet.
- **manual** - the group was assigned by hand.
- **rule** - the group was assigned by a rule through policy execution.

State - the status of the group assignment. Possible values are:

- **ADD** - the assignment is requested by DirX Identity Provisioning and shall be created in the target system by the synchronization workflow.
- **DELETED** - DirX Identity Provisioning requests the assignment to be deleted in the target system by the synchronization workflow. When it is deleted in the target system, the workflow removes the assignment physically from the **dxrMemberDel** attribute.
- **ENABLED** - the assignment is established both in DirX Identity Provisioning and in the target system: they are “in-sync”.
- **IMPORTED** - the assignment was created in the target system, but not yet requested in DirX Identity Provisioning. DirX Identity Provisioning does not delete it (ignores it), but it may be an indicator to the administrator that the user has access rights not granted by the assigned roles. DirX Identity Provisioning switches the state to **ENABLED** when the user is granted this access right. If it is revoked afterwards, the state switches to **DELETED** and not back to **IMPORTED**!

  See the section "Managing States" in the chapter "Managing Provisioning" to understand the relationship to the states of other objects.

  Note that DirX Identity shows assignments in approval as gray lines. Valid values for such entries are:
- **Add** - the assignment is not yet active.
- **Mod** - parts of the assignment (some attributes) are to be changed.
- **Del** - the assignment is to be deleted.
- **INACTIVE** - this state is a virtual one that may have two causes:
- The end date of this assignment has passed.
- The privilege is already assigned (for example, by the policy execution service), but a privilege resolution has not yet run. Run the privilege resolution to solve this problem.
- The privilege is already assigned but the assignment is still in approval. Completing the approval workflow will change the state.

The reason for this state is that the dxrGroupLink at the user has been populated, but the member attributes at the group or account are not yet populated.

To add or remove groups, use the assignment editor, which is visible when you click **Edit**. To examine the details of a particular group, click … on the right of its row.

Note that a line is displayed in gray if this assignment is still in approval. Thus you can see a white line that defines the current state and maybe several gray lines that define changes for this assignment that are still in approval.

Related Topics

- User
- Relationships

- Operational Properties
- Communication
- Authentication
- Organization
- Location
- Context
- Assigned Roles
- Assigned Permissions
- Accounts
- Orders
- SoD Exceptions
- Risk Parameters

A.1.1.12. User - Accounts

Use this tab to display the list of accounts currently assigned to the user.

For each account, the following properties are shown:

Account - the account's name as it is displayed throughout the DirX Identity Provisioning system.

Description - the description for the account.

State - the status of the account. Possible values are:

- **ENABLED** - the account shall exist, be created in the target system and be enabled.
- **DISABLED** - the account shall exist, be created in the target system and be disabled.
- **DELETED** - the account shall be deleted in the target system.
- **IMPORTED** - the account was created in the target system, but is not yet requested by DirX Identity Provisioning.

Target System - the target system to which this account belongs.

State in TS (target system) - the status of the account in the respective target system. This status may differ from the status of the account entry in DirX Identity Provisioning as long as the information is not synchronized. Possible values are:

- **ENABLED -** the account exists in the target system and is enabled.
- **DISABLED -** the account exists in the target system and is disabled.
- **DELETED** - the object has been deleted in the target system without DirX Identity Provisioning having requested it! This state is only set by the validation workflow along with a message in the **dxrToDo** attribute reminding the administrator to inspect this situation.

- **NONE** - replaces the empty value when the object is created by the DirX Identity Manager or agent.

  See the section "Managing States" in the chapter "Managing Provisioning" to understand the relationship to the states of other objects.

Priv - whether the account is an assigned privileged account (checked) or a personal account that is a member in the assigned groups that define the access rights in the connected systems (unchecked). A privileged account can be used by many persons in parallel. Examples are the Administrator account in Windows or the root account in UNIX. Assignment of a privileged account means:
1) You can read the password of this account in clear text to log in to the corresponding target system.
2) Your certificates are copied to the account in the connected system, which allows you to log in with the corresponding card.

You cannot make direct assignment of accounts in this tab. To examine the details of a particular account, click [...] to the right of the respective table row.

Related Topics

- User
- Relationships
- Operational Properties
- Communication
- Authentication
- Organization
- Location
- Context
- Assigned Roles
- Assigned Permissions
- Assigned Groups
- Orders
- SoD Exceptions
- Risk Parameters

A.1.1.13. User - Orders

Use this tab to display all currently pending changes for this user. This information is kept in order objects (mostly part of running request workflows).

The upper part of this tab shows pending attribute modification requests, while the lower part shows pending privilege assignments or privilege assignment attribute changes.

The **Attribute Modifications** pane shows a line for each relevant attribute. The columns are:

Due Date - the date on which this change will be valid. If no value is supplied, the change depends on a pending attribute approval step of a running request workflow.

Attribute - the name of the attribute.

Old values - the old value(s) of the attribute.

New values - the new value(s) of the attribute.

- opens a detail page for this line.

The **Assignments to add (+), modify (*) or delete (-)** pane shows a line for each assignment or change. The columns are:

Op(eration) - the type of line. Possible values are:

(+) - the privilege is to be assigned (to be added).

(*) - the assignment is subject to a change. One or more of the attributes - for example, the start or end date or a role parameter - has changed.

(-) - the privilege is to be removed (to be deleted).

Due date - the date on which this assignment or assignment change will be valid. If no value is supplied, the change is dependent on a pending attribute approval step of a running request workflow.

Type - the privilege type (role, permission or group).

Name - the privilege name.

Attribute - the name of the attribute to be changed.

Old values - the old value(s) of the attribute.

New values - the new value(s) of the attribute.

Assigned by - the type of assignment (manual or by rule).

- opens a detail page for this line. It shows the corresponding request workflow instance.

Note that the assignment information is also visible in the Assigned Roles, Assigned Permissions and Assigned Group tabs in a slightly different format.

Related Topics

- User
- Relationships
- Operational Properties
- Communication
- Authentication

- Organization
- Location
- Context
- Assigned Roles
- Assigned Permissions
- Assigned Groups
- Accounts
- SoD Exceptions
- Risk Parameters

A.1.1.14. User - SoD Exceptions

Use this tab to display all segregation of duty (SoD) exceptions for this user. SoD exceptions are SoD conflicts that have been approved successfully via an assignment approval workflow.

The upper part of this tab shows approved SoD violations, the lower part shows pending SoD violations.

For each SoD exception line in the **Approved SoD violations** table, the following columns are shown:

Policy - the SoD policy that caused this exception.

Privilege - the privilege that was assigned to the user.

Activity - the approval activity of the corresponding request workflow.

Reason - the reason why the approver accepted this SoD conflict.

Who - the approver.

When - the date of approval.

For each SoD exception line in the **Pending SoD violations** table, the following columns are shown:

Policy - the SoD policy that caused this exception.

Privilege - the privilege that was assigned to the user.

Click ... to open a detail page for this line.

Related Topics

- User
- Relationships
- Operational Properties

- Communication
- Authentication
- Organization
- Location
- Context
- Assigned Roles
- Assigned Permissions
- Assigned Groups
- Accounts
- Orders
- Risk Parameters

A.1.1.15. User - Risk Parameters

Use this tab to display the risk assessment properties for a user.

The **Overall Risk** section displays the computed overall risk level and the compound score. It includes the following properties:

Risk Level - the overall risk level for the user. Possible values are **Low**, **Medium** or **High**.

Compound Score - the calculated compound score for the user. All of the user's standard scores are used to compute the compound score.

The **Risk Factors** section provides the values for every risk factor defined in the domain's risk policy. Each risk factor is listed by name; for example, **GroupMemberships**, **ImportedAccounts**, **PrivilegedAccounts**. Risk factors shown as **RiskFactor*_number_ - for example, *RiskFactor6** - represent risk factors that have not been configured and have so not been processed by the Risk Calculation workflow. For every risk factor, the following properties are shown:

Raw Value - the raw value for the risk factor. This value is the computed score of the factor at the user.

Standard Score - the standard score for the risk factor. This value is a normalized score for the factor at the user.

See the section "Managing Risk" in the chapter "Managing Compliance" for details on risk calculation.

Related Topics

- User
- Relationships
- Operational Properties
- Communication

- Authentication
- Organization
- Location
- Context
- Assigned Roles
- Assigned Permissions
- Assigned Groups
- Accounts
- Orders
- SoD Exceptions
- Risk Policy - General

A.1.2. User Facets

A.1.2.1. User Facet - General Properties

User facets are alternative representations of a user. User facets are intended for users with different capacities or positions. For example, suppose you have a student who works as a teaching assistant and as a tutor. In some instances, you might be interested in the privileges that derive from his capacity as a student; in other cases, you might want to know about the privileges that derive from his position as a tutor. You can model these different profiles by creating two user facets; both of which are linked to the real user. If you are only interested in the privileges granted from his capacity as a tutor, then look at the "tutor" user facet. At the user, you can view all of the user's privileges with the resulting accounts.

This tab shows all general properties of the user facet object. Since it is fully configurable by the domain administrator, the items shown here may vary from installation to installation or even from domain to domain.

Note that a user facet is related via its owner link to its user. Depending on the processes in a company, it may make sense to use the master mechanism so that the user facet inherits the properties of the user. In this case, the text entry fields are not editable but are automatically populated from the owner.

The following properties explain the default configuration of DirX Identity:

General (Identification) Properties

Name (mandatory) - used as the relative distinguished name (RDN) for the user facet entry and identifies the user facet uniquely within the sub-tree. Once set during adding a new user facet entry, it can no longer be changed by simply editing this field after the initial save operation. Use the Rename command of the user facet entry's context menu instead. A change of the common name is also possible by running the user facet integration workflow.

First Name (optional) - the first name of the user facet.

Middle Name (optional) - the middle name of the user facet.

Salutation (optional) - the salutation of the user facet. Values for different countries are available.

Title (optional) - the title of this user facet. Mastered from the user.

Last Name (mandatory) - the last name of the user facet. This item corresponds to the "surname" attribute saved in the Identity Store.

Gender (optional) - the gender of this user facet. Mastered from the user.

Day of birth (optional) - the day of birth for this user facet.

Master (optional) - the name of the directory that masters the user facet entry.

Employee number (optional) - the employee number.

Identifier (optional) - the user facet's global unique identifier.

Description (optional) - A description for the user facet. This is often useful, when user facet entries are listed in tables just with their names but with a description column to better identify them.

Employee Type (optional) - one of the following user facet's employee types:

- **Contractor** - an (outside) contractor.
- **Customer** - a customer.
- **Internal** - an internal employee.
- **Supplier** - a supplier.

Business Category (optional) - the business the user facet works for.

Related Topics:

- Multi-Value Editor
- User Facet - Relationships
- User Facet - Operational Properties
- User Facet - Communication
- User Facet - Authentication
- User Facet - Organization
- User Facet - Location
- User Facet - Context
- User Facet - Assigned Roles
- User Facet - Assigned Permissions

- User Facet - Assigned Groups
- User Facet - Orders
- User Facet - SoD Exceptions

A.1.2.2. User Facet - Relationships

Use this tab to define relationships to other user facet objects.

Owner (mandatory) - the owner of the user facet entry. It references the user facet's related user.

Manager (optional) - the user facet's manager. Used for user facet type Internal Employee. Mastered from the user.

Secretary (optional) - the user facet's secretary.

Representative (optional) - the user facet's representative.

Sponsor (optional) - the user facet's sponsor. Used mainly for user type Contractor.

Note that these links are very useful when setting up participants for request workflows. For more information, see the section "Participant Calculation" in the *DirX Identity Application Development Guide*.

Related Topics

- User Facet - General Properties
- User Facet - Operational Properties
- User Facet - Communication
- User Facet - Authentication
- User Facet - Organization
- User Facet - Location
- User Facet - Context
- User Facet - Assigned Roles
- User Facet - Assigned Permissions
- User Facet - Assigned Groups
- User Facet - Orders
- User Facet - SoD Exceptions

A.1.2.3. User Facet - Operational Properties

Use this tab to display the operational parameters necessary for correct operation on the user facet entry within DirX Identity.

Status fields

To be analyzed (read-only) - indicates when checked that this entry has been changed. For example, an import workflow added or modified some attributes. The privilege resolution or the consistency checker uses this flag to detect and resolve records that are not up-to-date. After the check or resolution operation completes, the flag is cleared.

Is Inconsistent - indicates whether (unchecked) or not (checked) the stored access rights of the user facet are consistent with the assigned privileges. DirX Identity Provisioning checks the flag (sets the entry's flag to TRUE), if it fails to resolve the assigned privileges due to errors in the privilege structure. In this case, the user facet keeps the previous access rights, but DirX Identity Provisioning sets an end date for this state: the **Error expiration date**. The resolution error is stored in the **Error** field. So the administrator has time to repair the error. The flag is unchecked if the entry is consistent.

Use as Template - indicates when checked that DirX Identity will not resolve privileges assigned to this user facet. A likely reason for the checked flag is that the user facet object has been copied from another object. In this case, the user facet is marked as a template until appropriate changes have been made to the copied user facet. If the user facet is set correctly, uncheck the flag to enforce privilege resolution and target system provisioning.

+ A template user facet may also have been deliberately created to be used as a basis for creating additional accounts that are fully controlled by DirX Identity. You can use the owner link to point from this user facet its main identity. This reference can help you to understand which secondary user facet entries belong to a main identity one.

Status - shows the status of the user facet entry. For more information, see the section "Managing States". Possible values are:

- **TEMPLATE -** the settings made for this user facet are used as a template.
- **NEW -** the user facet is a new user facet; the privilege assignments are not activated yet.
- **ENABLED** - the user facet is established; the corresponding user facet object is up to date and valid, the privilege assignments are active.
- **DISABLED** - the privilege assignments for this user facet are currently disabled, the user facet does not have any rights in the domain.
- **TBDEL** - the user facet object shall be deleted from the DirX Identity Provisioning system. This action is taken when all assigned accounts in the target systems are deleted or the end date for deletion is reached.

Lifetime Start - shows the date on which the privilege assignments to the user facet will become active for the first time (usually the date on which the user facet's privileges must be activated for the main identity).

Lifetime End - shows the date on which the privilege assignments to the user facet will become inactive for the last time (usually the date on which the user facet's purpose has been satisfied so that the main identity no longer needs its privileges). Note that when the status of the main identity changes to TBDEL, the user facet's state changes as well and its EndDate is set.

Deactivation Start - shows the date on which the privilege assignments will become inactive for the next time.

Deactivation End - shows the date on which the privilege assignments will be reactivated again for the next time.

Delete - shows the date on which the entry will finally be deleted despite any existing accounts in target systems.

Create time stamp - shows the date and time at which this entry was created.

Modify time stamp - shows the date and time at which this entry was last changed.

Notification fields

Notification Level - controls suppression of the first e-mail of the e-mail notification of an approval workflow. Possible values are:

- **0** - the first e-mail is sent.
- **1** - the first e-mail is not sent (but all other administrative or repetition e-mails).

Tasks fields

To do - shows a list of manual working items for the administrator. DirX Identity Provisioning stores the results of consistency checks it cannot repair automatically. The items are overwritten on the next consistency check.

Possible messages in this field besides warnings from the service agent are:
INF:530:Missing values for attributes: *attributelist*.
*INF:531:More than one peer account exists for account DN=**dn*.
*INF:532:No peer account exists for account DN=**dn*.

Error - stores the list of error messages that give the detailed reason for inconsistent states or **To do** entries. They are cleared automatically when a privilege resolution succeeds.

Error expiration date - shows the date on which the access rights of the user will be updated despite any resolution errors. See the **Is Inconsistent** field.

Related Topics

- User Facet - General
- User Facet - Relationships
- User Facet - Communication
- User Facet - Authentication
- User Facet - Organization
- User Facet - Location
- User Facet - Context
- User Facet - Assigned Roles
- User Facet - Assigned Permissions
- User Facet - Assigned Groups

- User Facet - Orders
- User Facet - SoD Exceptions

A.1.2.4. User Facet - Communication

Use this tab to display communication attributes for a user facet object.

E-Mail (optional) - the user facet's e-mail addresses. Mastered from the user.

Phone (optional) - the user facet's main phone number (business phone number).

Fax (optional) - the user facet's fax number.

Mobile (optional) - the user facet's mobile number. Mastered from the user.

Home (optional) - the user facet's home phone number. Mastered from the user.

Web address (optional) - the Web address of the Internet home page for this user facet.

Preferred language (optional) - the preferred language of this user facet (mastered from the user), which controls, for example, the language used in an e-mail if this user facet is referenced in the To field. For more information, see the chapter "Using Variable Substitution" in the *DirX Identity Application Development Guide*.

Note that the phone number might be used by default when DirX Identity Provisioning creates a new account for the user facet. The account name is generated using the user facet's surname, given name and phone number. See the object description for the accounts.

Related Topics

- User Facet - General Properties
- User Facet - Relationships
- User Facet - Operational Properties
- User Facet - Authentication
- User Facet - Organization
- User Facet - Location
- User Facet - Context
- User Facet - Assigned Roles
- User Facet - Assigned Permissions
- User Facet - Assigned Groups
- User Facet - Orders
- User Facet - SoD Exceptions

A.1.2.5. User Facet - Authentication

Use this tab to display the authentication properties for this user facet.

Encrypted Password - the encrypted (or scrambled) password of this user facet (dxmPassword attribute). This attribute is used for password synchronization to target systems. Authentication against the Identity Store is performed via the stored one-way password (userPassword attribute).

Password Account Locked Time (read-only) - the time at which the account was locked.

Password Changed Time (read-only) - the time at which the password was last changed.

Password Expiration Notified (read-only) - the time at which Password Notification service notified the system about the password expiration.

Password Failure Time (read-only) - the time at which the password was entered incorrectly.

Password History (read-only) - the password history (one-way encrypted). This field shows only that there is a password history.

Password Reset (read-only) - if active, the password was reset by the administrator.

Password Policy - a link to the relevant password policy. If no link is set, the default policy is used by the system.

Challenges and Responses (read-only) - the challenge and response questions (one-way encrypted). This field shows only that challenge and response questions have been set up.

Certificate (read-only) - the certificate of the user facet.

Related Topics

- User Facet - General Properties
- User Facet - Relationships
- User Facet - Operational Properties
- User Facet - Communication
- User Facet - Organization
- User Facet - Location
- User Facet - Context
- User Facet - Assigned Roles
- User Facet - Assigned Permissions
- User Facet - Assigned Groups
- User Facet - Orders
- User Facet - SoD Exceptions

A.1.2.6. User Facet - Organization

Use this tab to display the organization and organizational units properties of a user facet. These properties can be linked with the relevant business objects.

Organizations

Organization Link (optional) - sets a link to an organization object (business object).

Organization (optional) - defines the organization to which the user facet belongs. If the **Organization Link** is set, this attribute is read-only and is controlled by the linked object.

Organizational Units

Org. Unit Link (optional) - sets a link to an organizational unit object (business object).

Organizational Unit (optional) - the organizational unit to which the user facet belongs. If the **Org. Unit Link** is set, this attribute is read-only and controlled by the linked object.

Department Number (optional) - the department number of the organizational unit. If the **Org. Unit Link** is set, this attribute is read-only and controlled by the linked object.

More Organizational Units (optional) - defines more organizational unit links for this user facet.

Related Topics

- User Facet - General Properties
- User Facet - Relationships
- User Facet - Operational Properties
- User Facet - Communication
- User Facet - Authentication
- User Facet - Location
- User Facet - Context
- User Facet - Assigned Roles
- User Facet - Assigned Permissions
- User Facet - Assigned Groups
- User Facet - Orders
- User Facet - SoD Exceptions

A.1.2.7. User Facet - Location

Use this tab to display and set location information associated with the user facet.

Location Link (optional) - sets a link to a location object (business object).

Country (optional) - the country of the user facet. If the Location Link is set, this attribute is

read-only and is controlled by the linked object.

State (optional) - the state of the user facet. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

Postal Code (optional) - the postal code of the location. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

Location (optional) - the location(s) of the user facet. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

Street (optional) - the street name of the location. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

Room (optional) - the room where this user facet is located.

Postal Address (optional) - the postal address of this location. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

Related Topics

- User Facet - General Properties
- User Facet - Relationships
- User Facet - Operational Properties
- User Facet - Communication
- User Facet - Authentication
- User Facet - Organization
- User Facet - Context
- User Facet - Assigned Roles
- User Facet - Assigned Permissions
- User Facet - Assigned Groups
- User Facet - Orders
- User Facet - SoD Exceptions

A.1.2.8. User Facet - Context

This tab is an example of other context information. Pre-configured links are:

Contexts (optional) - a link to any type of context object (business object).

Related Topics

- User Facet - General Properties
- User Facet - Relationships
- User Facet - Operational Properties
- User Facet - Communication

- User Facet - Authentication
- User Facet - Organization
- User Facet - Location
- User Facet - Assigned Roles
- User Facet - Assigned Permissions
- User Facet - Assigned Groups
- User Facet - Orders
- User Facet - SoD Exceptions

A.1.2.9. User Facet - Assigned Roles

Use this tab to view and manage the roles assigned to a user facet. The tab displays the list of roles currently assigned to the user facet.

For each role, the following properties are shown:

Role - the role's name as it is displayed throughout the DirX Identity Provisioning system. If the role's name ends with an asterisk character (*), the role has assigned one or more junior roles. These junior roles are automatically assigned to the user as well.

Description - the description for the role.

Start Date - the date on which this role became or will become active for the user facet; that is, when the user facet was/is granted the permissions assigned to this role.

End Date - the date on which this role became or will become inactive for the user facet; that is, when the permissions granted to the user facet was or will be revoked. If the re-approval flag is set, the end date shown is the date of the re-approval. You can change the end date to shorten the period to the next re-approval but you cannot lengthen it.

Reapproval - whether (checked) or not (unchecked) this assignment will be regularly reapproved. The next re-approval will occur at the shown end date. Unchecking this flag removes the privilege at the shown end date (re-approvals no longer occur).

Role Parameters - shows the first few assigned role parameters. Use the … button to view all role parameter settings for this assignment.

Assigned by - how the assignment was made. Possible values are:

- **BO** - the role was assigned by privilege inheritance from a business object.
- **manual** - the role was assigned by hand.
- **rule** - the role was assigned by a rule.

State - the state of the assignment:

- **ToBeApproved** - the assignment is still in approval. The corresponding access rights (groups) are not yet assigned.

- **ENABLED** - the assignment is enabled. The user facet has the corresponding access rights (groups) assigned.
- **DELETED** - the assignment is deleted. The corresponding access rights (groups) are not yet or no longer assigned.

  See the section "Managing States" in the chapter "Managing Provisioning" to understand the relationship to the states of other objects.

  Note that DirX Identity shows assignments in approval as gray lines. Valid values for such entries are:
- **Add** - this assignment is not yet active.
- **Mod** - parts of this assignment (some attributes) were changed.
- **Del** - this assignment is to be deleted.

To add or remove roles, use the assignment editor, which is displayed when you click **Edit**. To examine the details of a particular role, click ... to the right of its row.

Note that a line is displayed in gray if this assignment is still in approval. Thus you may see a white line that defines the current state and possibly several gray lines that define changes for this assignment that are still in approval.

Related Topics

- User Facet - General Properties
- User Facet - Relationships
- User Facet - Operational Properties
- User Facet - Communication
- User Facet - Authentication
- User Facet - Organization
- User Facet - Location
- User Facet - Context
- User Facet - Assigned Permissions
- User Facet - Assigned Groups
- User Facet - Orders
- User Facet - SoD Exceptions

A.1.2.10. User Facet - Assigned Permissions

Use this tab to view and manage the permissions assigned to a user facet. The tab shows the list of permissions currently assigned to the user facet.

For each permission, the following properties are displayed:

Permission - the permission's name as it is displayed throughout the DirX Identity

Provisioning system.

Description - the description of the permission.

Start Date - the date on which this permission became or will become active for the user facet.

End Date - the date on which this permission became or will become inactive for the user facet. If the re-approval flag is set, the shown end date is the date of the re-approval. You can change the end date to shorten the period to the next re-approval but you cannot lengthen it.

Reapproval - whether (checked) or not (unchecked) this assignment requires regular re-approval. The next re-approval occurs at the shown end date. If you uncheck this flag, privilege re-approval is removed at the shown end date (re-approvals no longer occur after the end date is reached).

Assigned by - how the permission assignment was made. Possible values are:

- (empty) - This permission is inherited from a role.
- **BO** - the permission was assigned by privilege inheritance from a business object.
- **manual** - the permission was assigned by hand.
- **rule** - This permission was assigned by a rule.

State - the state of the assignment:

- **ToBeApproved** - the assignment is still in approval. The corresponding access rights (groups) are not yet assigned.
- **INHERITED** - this assignment is inherited from a role assignment.
- **DELETED** - the assignment is deleted. The corresponding access rights (groups) are not yet or no longer assigned.

  See the section "Managing States" in the chapter "Managing Provisioning" to understand the relationship to the states of other objects.

  Note that DirX Identity shows assignments in approval as gray lines. Valid values for such entries are:
- **Add** - this assignment is not yet active.
- **Mod** - parts of this assignment (some attributes) are to be changed.
- **Del** - this assignment is to be deleted.

To add or remove permissions manually, use the assignment editor, which is visible when you click **Edit**. To examine the details of a particular permission, click … to the right of its row.

Note that a line is displayed in gray if this assignment is still in approval. Thus you can see a white line that defines the current state and maybe several gray lines that define changes for this assignment that are still in approval.

Related Topics

- User Facet - General Properties
- User Facet - Relationships
- User Facet - Operational Properties
- User Facet - Communication
- User Facet - Authentication
- User Facet - Organization
- User Facet - Location
- User Facet - Context
- User Facet - Assigned Roles
- User Facet - Assigned Groups
- User Facet - Orders
- User Facet - SoD Exceptions

A.1.2.11. User Facet - Assigned Groups

Use this tab to view and manage the groups assigned to a user facet. The tab displays the list of groups currently assigned to the user facet.

For each group, the following properties are shown:

Target system - the name of the target system to which the group belongs.

Group - the group's name as it is displayed throughout the DirX Identity Provisioning system.

Description - the description of the group.

Start Date - the date on which this direct group assignment became or will become inactive for the user facet.

End Date - the date on which this direct group assignment became or will become inactive for the user facet. If the re-approval flag is set, the shown end date is the date of the re-approval. You can change the end date to shorten the period to the next re-approval but you cannot lengthen it.

Reapproval - whether (checked) or not (unchecked) this assignment will require regular re-approval. The next re-approval will occur at the end date shown. Unchecking this flag removes the privilege from re-approval at the shown end date (re-approvals are no longer required after this date is reached).

Assigned by - how the assignment was made:

- (empty) - the group was assigned by a permission or role through privilege assignment.
- **BO** - the group was assigned by privilege inheritance from a business object.

- **manual** - the group was assigned by hand.
- **rule** - the group was assigned by a rule through policy execution.

State - the status of the group assignment. Possible values are:

- **ADD** - the assignment is requested by DirX Identity Provisioning and shall be created in the target system by the synchronization workflow.
- **DELETED** - DirX Identity Provisioning requests the assignment to be deleted in the target system by the synchronization workflow. When it is deleted in the target system, the workflow removes the assignment physically from the **dxrMemberDel** attribute.
- **ENABLED** - the assignment is established both in DirX Identity Provisioning and in the target system: they are "in-sync".
- **IMPORTED** - the assignment was created in the target system, but not yet requested in DirX Identity Provisioning. DirX Identity Provisioning does not delete it (ignores it), but it may be an indicator to the administrator, that the user facet has access rights not granted by the assigned roles. DirX Identity Provisioning switches the state to **ENABLED**, when the user facet is granted this access right. If it is revoked afterwards, the state switches to **DELETED** and not back to **IMPORTED**!

  See the section "Managing States" in the chapter "Managing Provisioning" to understand the relationship to the states of other objects.

  Note that DirX Identity shows assignments in approval as gray lines. Valid values for such entries are:
- **Add** - the assignment is not yet active.
- **Mod** - parts of this assignment (some attributes) are to be changed.
- **Del** - the assignment is to be deleted.
- **INACTIVE** - the state is a virtual one that may have two causes:
- The end date of this assignment has passed.
- The privilege is already assigned (for example, by the policy execution service), but a privilege resolution has not yet run. Run the privilege resolution to solve this problem.
- The privilege is already assigned but the assignment is still in approval. Completing the approval workflow will change the state.

The reason for this state is that the dxrGroupLink at the user facet is already populated, but the member attributes at the group or account are not yet populated.

To add or remove groups manually, use the assignment editor, which is visible when you click **Edit**. To examine the details of a particular group, click ... to the right of its row.

Note that a line is displayed in gray if this assignment is still in approval. Thus you can see a white line that defines the current state and maybe several gray lines that define changes for this assignment that are still in approval.

Related Topics

- User Facet - General Properties
- User Facet - Relationships
- User Facet - Operational Properties
- User Facet - Communication
- User Facet - Authentication
- User Facet - Organization
- User Facet - Location
- User Facet - Context
- User Facet - Assigned Roles
- User Facet - Assigned Permissions
- User Facet - Orders
- User Facet - SoD Exceptions

A.1.2.12. User Facet - Orders

Use this tab to display all currently pending changes for this user facet. This information is kept in order objects (mostly part of running request workflows).

The upper part of this tab shows pending attribute modification requests, the lower part shows pending privilege assignments or privilege assignment attribute changes.

The **Attribute Modifications** pane shows a line for each relevant attribute. The columns are:

Due Date - the date on which this change will be valid. If no value is supplied, the change depends on a pending attribute approval step of a running request workflow.

Attribute - the name of the attribute.

Old values - the old value(s) of the attribute.

New values - the new value(s) of the attribute.


- opens a detail page for this line.

The **Assignments to add (+), modify (*) or delete (-)** pane shows a line for each assignment or change. The columns are:

Op(eration) - the type of line. Possible values are:

(+) - the privilege is to be assigned (to be added).

(*) - the assignment is subject to a change. One or more of the attributes - for example, the start or end date or a role parameter - has changed.

(-) - the privilege is to be removed (to be deleted).

Due date - the date on which this assignment or assignment change will be valid. If no

value is supplied, the change depends on a pending attribute approval step of a running request workflow.

Type - the privilege type (role, permission or group).

Name - the privilege name.

Attribute - the name of the attribute to be changed.

Old values - the old value(s) of the attribute.

New values - the new value(s) of the attribute.

Assigned by - the type of the assignment (manual or by rule).

 - opens a detail page for this line. It shows the corresponding request workflow instance.

Note that the assignment information is also visible in the Assigned Roles, Assigned Permissions and Assigned Group tabs in a slightly different format.

Related Topics

- User Facet - General Properties
- User Facet - Relationships
- User Facet - Operational Properties
- User Facet - Communication
- User Facet - Authentication
- User Facet - Organization
- User Facet - Location
- User Facet - Context
- User Facet - Assigned Roles
- User Facet - Assigned Permissions
- User Facet - Assigned Groups
- User Facet - SoD Exceptions

A.1.2.13. User Facet - SoD Exceptions

Use this tab to display all segregation of duty (SoD) exceptions for this user facet. SoD exceptions are SoD conflicts that have been approved successfully via an assignment approval workflow.

Note that the scope of SOD checking is limited to single users or user facets only. Thus. if a user facet has a privilege assigned that violates an SOD policy for another privilege being assigned to its user or to another user facet of its user, no violation is detected.

The upper part of this tab shows approved SoD violations, the lower part shows pending SoD violations.

For each SoD exception line in the **Approved SoD violations** table, the following columns are shown:

Policy - the SoD policy that caused this exception.

Privilege - the privilege that was assigned to the user facet.

Activity - the approval activity of the corresponding request workflow.

Reason - the reason why the approver accepted this SoD conflict.

Who - the approver.

When - the date of approval.

For each SoD exception line in the **Pending SoD violations** table, the following columns are shown:

Policy - the SoD policy that caused this exception.

Privilege - the privilege that was assigned to the user facet.

Click [...] to open a detail page for this line.

Related Topics

- User Facet - General Properties
- User Facet - Relationships
- User Facet - Operational Properties
- User Facet - Communication
- User Facet - Authentication
- User Facet - Organization
- User Facet - Location
- User Facet - Context
- User Facet - Assigned Roles
- User Facet - Assigned Permissions
- User Facet - Assigned Groups
- User Facet - Orders

A.1.2.14. User Facet - Risk Parameters

Use this tab to display the risk assessment properties for the user facet.

The **Overall Risk** section displays the computed overall risk level and the compound score. It includes the following properties:

Risk Level - the overall risk level for the user facet. Possible values are **Low**, **Medium** or

High.

Compound Score - the calculated compound score for the user facet. All of the user facet's standard scores are used to compute the compound score.

The **Risk Factors** section provides the values for every risk factor defined in the domain's risk policy. Each risk factor is listed by name; for example, **GroupMemberships**, **ImportedAccounts**, **PrivilegedAccounts**. Risk factors shown as **RiskFactor*_number_ - for example, *RiskFactor6** - represent risk factors that have not been configured and have so not been processed by the Risk Calculation workflow. For every risk factor, the following properties are shown:

Raw Value - the raw value for the risk factor. This value is the computed score of the factor at the user facet.

Standard Score - the standard score for the risk factor. This value is a normalized score for the factor at the user facet.

See the section "Managing Risk" in the chapter "Managing Compliance" for details on risk calculation.

Related Topics

- User Facet - General Properties
- User Facet - Relationships
- User Facet - Operational Properties
- User Facet - Communication
- User Facet - Authentication
- User Facet - Organization
- User Facet - Location
- User Facet - Context
- User Facet - Assigned Roles
- User Facet - Assigned Permissions
- User Facet - Assigned Groups
- User Facet - Orders

Risk Policy - General

A.1.3. Persona

A.1.3.1. Persona - General Properties

Personas are alternative representations of a user. The user works in different functions in the company, for example as an administrator or as a project manager. The accounts and the access rights for each representation might be quite different and also auditing should be able to distinguish between these functions of a user.

The persona object represents a profile of a user of an IT infrastructure. DirX Identity Provisioning manages the persona's access to all parts of the IT infrastructure.

This tab shows all general properties of the persona object. Since it is fully configurable by the domain administrator, the items shown here may vary from installation to installation or even from domain to domain.

Note that a persona is related via its owner link to its user. Depending on the processes in a company, it makes sense to inherit properties of the user to the persona by the master mechanism. In that case, the text entry fields are not editable but are automatically filled from the owner.

The following properties explain the default configuration of DirX Identity:

General (Identification) Properties

Name (mandatory) - used as the relative distinguished name (RDN) for the persona entry and identifies the persona uniquely within the sub-tree. Once set during adding a new persona entry, it can no longer be changed by simply editing this field after the initial save operation. Use the **Rename** command of the persona entry's context menu instead. A change of the common name is also possible by running the persona integration workflow.

First Name (optional) - the first name of the persona.

Middle Name (optional) - the middle name of the persona.

Salutation (optional) - the salutation of the persona. Values for different countries are available.

Title (optional) - the title of this persona. Mastered from the user.

Last Name (mandatory) - the last name of the persona. This item corresponds to the "surname" attribute saved in the Identity Store.

Gender (optional) - the gender of this persona. Mastered from the user.

Day of birth (optional) - the day of birth for this persona.

Master (optional) - the name of the directory that masters the persona entry.

Employee number (optional) - the employee number.

Identifier (optional) - the persona's global unique identifier.

Description (optional) - A description for the persona. This is often useful, when persona entries are listed in tables just with their names but with a description column to better identify them.

Employee Type (optional) - one of the following persona's employee types:

- **Contractor** - an (outside) contractor.
- **Customer** - a customer.

- **Internal** - an internal employee.
- **Supplier** - a supplier.

Business Category (optional) - the business the persona works for.

Related Topics:

- Persona - General Properties
- Persona - Relationships
- Persona - Operational Properties
- Persona - Communication
- Persona - Authentication
- Persona - Organization
- Persona - Location
- Persona - Context
- Persona - Assigned Roles
- Persona - Assigned Permissions
- Persona - Assigned Groups
- Persona - Accounts
- Persona - Orders
- Persona - SoD Exceptions

A.1.3.2. Persona - Relationships

Use this tab to define relationships to other user objects.

Owner (mandatory) - the owner of the persona entry. It references the persona's related user.

Manager (optional) - the persona's manager. Used for persona type Internal Employee. Mastered from the user.

Secretary (optional) - the persona's secretary.

Representative (optional - the persona's representative.

Sponsor (optional) - the persona's sponsor. Used mainly for user type Contractor.

Note that these links are very useful when setting up participants for request workflows. For more information, see the section "Participant Calculation" in the *DirX Identity Application Development Guide*.

Related Topics

- Persona - General Properties

- Persona - Operational Properties
- Persona - Communication
- Persona - Authentication
- Persona - Organization
- Persona - Location
- Persona - Context
- Persona - Assigned Roles
- Persona - Assigned Permissions
- Persona - Assigned Groups
- Persona - Accounts
- Persona - Orders
- Persona - SoD Exceptions

A.1.3.3. Persona - Operational Properties

Use this tab to display the operational parameters necessary for correct operation on the persona entry within DirX Identity.

Status fields

To be analyzed (read-only) - indicates when checked that this entry has been changed. For example, an import workflow added or modified some attributes. The privilege resolution or the consistency checker uses this flag to detect and resolve records that are not up to date. These operations clear the flag after the check or resolution completes.

Is Inconsistent - indicates whether (unchecked) or not (checked) the stored access rights of the persona are consistent with the assigned privileges. DirX Identity Provisioning checks this field (sets the entry's flag to TRUE) if it fails to resolve the assigned privileges due to errors in the privilege structure. In this case, the persona keeps the previous access rights, but DirX Identity Provisioning sets an end date for this state: the **Error expiration date**. The resolution error is stored in the Error field so that the administrator has time to repair the error.

Use as Template - indicates when checked that DirX Identity will not resolve privileges assigned to this persona. When this flag is checked, a likely reason is that the persona object has been copied from another one. In this case, the persona is marked as a template until appropriate changes have been made to the copied persona. If the persona is set correctly, uncheck the flag to enforce privilege resolution and target system provisioning.

+ A template persona may also have been deliberately created to be used as a basis for creating additional accounts that are fully controlled by DirX Identity. You can use the owner link to point from this persona to its main identity. This reference can help you to understand which secondary persona entries belong to a main identity one.

Status - the status of the persona entry. For more information, see the section "Managing States". Possible values are:

- **TEMPLATE** - the settings made for this persona are used as a template.
- **NEW** - the persona is a new persona, the privilege assignments are not activated yet.
- **ENABLED** - the persona is established; the corresponding persona object is up to date and valid, the privilege assignments are active.
- **DISABLED** - the privilege assignments for this persona are currently disabled, the persona does not have any rights in the domain.
- **TBDEL** - the persona object shall be deleted from the DirX Identity Provisioning system. This action is taken when all assigned accounts in the target systems are deleted or the end date for deletion is reached.

Lifetime Start - shows the date on which the privilege assignments to the persona will become active for the first time (usually the date on which the persona's privileges have to become active for the main identity).

Lifetime End - shows the date on which the privilege assignments to the persona will become inactive for the last time (usually the date on which the persona's purpose is fulfilled, so that the main identity no longer requires its privileges). Note that when the status of the main identity changes to **TBDEL**, the persona's state changes as well, and its EndDate is set.

Deactivation Start - shows the date on which the privilege assignments will become inactive for the next time.

Deactivation End - shows the date on which the privilege assignments will be reactivated again for the next time.

Delete - shows the date on which the entry will be finally deleted despite any existing accounts in target systems.

Create time stamp - shows the date and time at which this entry was created.

Modify time stamp - shows the date and time at which this entry was last changed.

Notification fields

Notification Level - controls the suppression of the first e-mail of the e-mail notification of an approval workflow:

- **0** - the first e-mail is sent.
- **1** - the first e-mail is not sent (but all other administrative or repetition e-mails).

Tasks fields

To do - shows a list of manual working items for the administrator. DirX Identity Provisioning stores the results of consistency checks it cannot repair automatically. The items are overwritten on the next consistency check.

Possible messages in this field besides warnings from the service agent are:
INF:530:Missing values for attributes: *attributelist*.

*INF:531:More than one peer account exists for account DN=*dn*.
*INF:532:No peer account exists for account DN=*dn*.

Error - stores a list of error messages that show the detailed reason for inconsistent states or **To do** entries. They are cleared automatically when a privilege resolution succeeds.

Error expiration date - shows the date on which the access rights of the persona will be updated despite any resolution errors. See the **Is Inconsistent** field.

Related Topics

- Persona - General Properties
- Persona - Relationships
- Persona - Communication
- Persona - Authentication
- Persona - Organization
- Persona - Location
- Persona - Context
- Persona - Assigned Roles
- Persona - Assigned Permissions
- Persona - Assigned Groups
- Persona - Accounts
- Persona - Orders
- Persona - SoD Exceptions

A.1.3.4. Persona - Communication

Use this tab to display communication attributes for a persona object.

E-Mail (optional) - the persona's e-mail addresses. Mastered from the user.

Phone (optional) - the persona's main phone number (business phone number).

Fax (optional) - the persona's fax number.

Mobile (optional) - the persona's mobile number. Mastered from the user.

Home (optional) - the persona's home phone number. Mastered from the user.

Web address (optional) - the Web address of the Internet home page for this persona.

Preferred language (optional) - the preferred language of this persona (mastered from the user), which controls, for example, the language used in an e-mail if this persona is referenced in the To field. For more information, see the chapter "Using Variable Substitution" in the *DirX Identity Application Development Guide*.

Note that the phone number might be used by default when DirX Identity Provisioning

creates a new account for the persona. The account name is generated using the persona's surname, given name and phone number. See the object description for the accounts.

Related Topics

- Persona - Relationships
- Persona - Operational Properties
- Persona - Communication
- Persona - Authentication
- Persona - Organization
- Persona - Location
- Persona - Context
- Persona - Assigned Roles
- Persona - Assigned Permissions
- Persona - Assigned Groups
- Persona - Accounts
- Persona - Orders
- Persona - SoD Exceptions

A.1.3.5. Persona - Authentication

Use this tab to display the authentication properties for this persona.

Encrypted Password - the encrypted (or scrambled) password of this persona (dxmPassword attribute). This attribute is used for password synchronization to target systems. Authentication against the Identity Store is performed via the stored one-way password (userPassword attribute).

Password Account Locked Time (read-only) - the time at which the account was locked.

Password Changed Time (read-only) - the time at which the password was last changed.

Password Expiration Notified (read-only) - the time at which Password Notification service notified the system about the password expiration.

Password Failure Time (read-only) - the time at which the password was entered incorrectly.

Password History (read-only) - the password history (one-way encrypted). This field shows only that there is a password history.

Password Reset (read-only) - if active, the password was reset by the administrator.

Password Policy - a link to the relevant password policy. If no link is set, the default policy is used by the system.

Challenges and Responses (read-only) - the challenge and response questions (one-way encrypted). This field shows only that challenge and response questions have been set up.

Certificate (read-only) - the certificate of the persona.

Related Topics

- Persona - General Properties
- Persona - Relationships
- Persona - Operational Properties
- Persona - Communication
- Persona - Authentication
- Persona - Organization
- Persona - Location
- Persona - Context
- Persona - Assigned Roles
- Persona - Assigned Permissions
- Persona - Assigned Groups
- Persona - Accounts
- Persona - Orders
- Persona - SoD Exceptions

A.1.3.6. Persona - Organization

Use this tab to display the organization and organizational units properties of a persona. These properties can be linked with the relevant business objects.

Organizations

Organization Link (optional) - sets a link to an organization object (business object).

Organization (optional) - defines the organization to which the persona belongs. If the **Organization Link** is set, this attribute is read-only and is controlled by the linked object.

Organizational Units

Org. Unit Link (optional) - sets a link to an organizational unit object (business object).

Organizational Unit (optional) - the organizational unit to which the persona belongs. If the **Org. Unit Link** is set, this attribute is read-only and controlled by the linked object.

Department Number (optional) - the department number of the organizational unit. If the **Org. Unit Link** is set, this attribute is read-only and controlled by the linked object.

More Organizational Units (optional) - defines more organizational unit links for this persona.

Related Topics

- Persona - General Properties
- Persona - Relationships
- Persona - Operational Properties
- Persona - Communication
- Persona - Authentication
- Persona - Location
- Persona - Context
- Persona - Assigned Roles
- Persona - Assigned Permissions
- Persona - Assigned Groups
- Persona - Accounts
- Persona - Orders
- Persona - SoD Exceptions

A.1.3.7. Persona - Location

Use this tab to display and set location information associated with the persona.

Location Link (optional) - sets a link to a location object (business object).

Country (optional) - the country of the persona. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

State (optional) - the state of the persona. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

Postal Code (optional) - the postal code of the location. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

Location (optional) - the location(s) of the persona. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

Street (optional) - the street name of the location. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

Room (optional) - the room where this persona is located.

Postal Address (optional) - the postal address of this location. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

Related Topics

- Persona - General Properties
- Persona - Relationships

- Persona - Operational Properties
- Persona - Communication
- Persona - Authentication
- Persona - Organization
- Persona - Context
- Persona - Assigned Roles
- Persona - Assigned Permissions
- Persona - Assigned Groups
- Persona - Accounts
- Persona - Orders
- Persona - SoD Exceptions

A.1.3.8. Persona - Context

This tab is an example of other context information. Pre-configured links are:

Contexts (optional) - a link to any type of context object (business object).

Related Topics

- Persona - General Properties
- Persona - Relationships
- Persona - Operational Properties
- Persona - Communication
- Persona - Authentication
- Persona - Organization
- Persona - Location
- Persona - Assigned Roles
- Persona - Assigned Permissions
- Persona - Assigned Groups
- Persona - Accounts
- Persona - Orders
- Persona - SoD Exceptions

A.1.3.9. Persona - Assigned Roles

Use this tab to view and manage the roles assigned to a persona. The tab displays the list of roles currently assigned to the persona.

For each role, the following properties are shown:

Role - the role's name as it is displayed throughout the DirX Identity Provisioning system. If the role's name ends with an asterisk character (*), the role has assigned one or more junior roles. These junior roles are automatically assigned to the user as well.

Description - the description for the role.

Start Date - the date on which this role became or will become active for the persona; that is, when the persona was/is granted the permissions assigned to this role.

End Date - the date on which this role became or will become inactive for the persona; that is, when the permissions granted to the persona was or will be revoked. If the re-approval flag is set, the end date shown is the date of the re-approval. You can change the end date to shorten the period to the next re-approval but you cannot lengthen it.

Reapproval - whether (checked) or not (unchecked) this assignment will be regularly reapproved. The next re-approval will occur at the shown end date. Unchecking this flag removes the privilege at the shown end date (re-approvals no longer occur).

Role Parameters - shows the first few assigned role parameters. Use the … button to view all role parameter settings for this assignment.

Assigned by - how the assignment was made. Possible values are:

- **BO** - the role was assigned by privilege inheritance from a business object.
- **manual** - the role was assigned by hand.
- **rule** - the role was assigned by a rule.

State - the state of the assignment:

- **ToBeApproved** - the assignment is still in approval. The corresponding access rights (groups) are not yet assigned.
- **ENABLED** - the assignment is enabled. The persona has the corresponding access rights (groups) assigned.
- **DELETED** - the assignment is deleted. The corresponding access rights (groups) are not yet or no longer assigned.

  See the section "Managing States" in the chapter "Managing Provisioning" to understand the relationship to the states of other objects.

  Note that DirX Identity shows assignments in approval as gray lines. Valid values for such entries are:

- **Add** - this assignment is not yet active.
- **Mod** - parts of this assignment (some attributes) were changed.
- **Del** - this assignment is to be deleted.

To add or remove roles, use the assignment editor, which is displayed when you click **Edit**. To examine the details of a particular role, click … to the right of its row.

Note that a line is displayed in gray if this assignment is still in approval. Thus you may see a white line that defines the current state and possibly several gray lines that define changes for this assignment that are still in approval.

Related Topics

- Persona - General Properties
- Persona - Relationships
- Persona - Operational Properties
- Persona - Communication
- Persona - Authentication
- Persona - Organization
- Persona - Location
- Persona - Context
- Persona - Assigned Permissions
- Persona - Assigned Groups
- Persona - Accounts
- Persona - Orders
- Persona - SoD Exceptions

A.1.3.10. Persona - Assigned Permissions

Use this tab to view and manage the permissions assigned to a persona. The tab shows the list of permissions currently assigned to the persona.

For each permission, the following properties are displayed:

Permission - the permission's name as it is displayed throughout the DirX Identity Provisioning system.

Description - the description of the permission.

Start Date - the date on which this permission became or will become active for the persona.

End Date - the date on which this permission became or will become inactive for the persona. If the re-approval flag is set, the shown end date is the date of the re-approval. You can change the end date to shorten the period to the next re-approval but you cannot lengthen it.

Reapproval - whether (checked) or not (unchecked) this assignment requires regular re-approval. The next re-approval occurs at the shown end date. If you uncheck this flag, privilege re-approval is removed at the shown end date (re-approvals no longer occur after the end date is reached).

Assigned by - how the permission assignment was made. Possible values are:

- (empty) - This permission is inherited from a role.
- **BO** - the permission was assigned by privilege inheritance from a business object.
- **manual** - the permission was assigned by hand.
- **rule** - This permission was assigned by a rule.

State - the state of the assignment:

- **ToBeApproved** - the assignment is still in approval. The corresponding access rights (groups) are not yet assigned.
- **INHERITED** - this assignment is inherited from a role assignment.
- **DELETED** - the assignment is deleted. The corresponding access rights (groups) are not yet or no longer assigned.

  See the section "Managing States" in the chapter "Managing Provisioning" to understand the relationship to the states of other objects.

  Note that DirX Identity shows assignments in approval as gray lines. Valid values for such entries are:
- **Add** - this assignment is not yet active.
- **Mod** - parts of this assignment (some attributes) are to be changed.
- **Del** - this assignment is to be deleted.

To add or remove permissions manually, use the assignment editor, which is visible when you click **Edit**. To examine the details of a particular permission, click … to the right of its row.

Note that a line is displayed in gray if this assignment is still in approval. Thus you can see a white line that defines the current state and maybe several gray lines that define changes for this assignment that are still in approval.

Related Topics

- Persona - General Properties
- Persona - Relationships
- Persona - Operational Properties
- Persona - Communication
- Persona - Authentication
- Persona - Organization
- Persona - Location
- Persona - Context
- Persona - Assigned Roles
- Persona - Assigned Groups
- Persona - Accounts

- Persona - Orders
- Persona - SoD Exceptions

A.1.3.11. Persona - Assigned Groups

Use this tab to view and manage the groups assigned to a persona. The tab displays the list of groups currently assigned to the persona.

For each group, the following properties are shown:

Target system - the name of the target system to which the group belongs.

Group - the group's name as it is displayed throughout the DirX Identity Provisioning system.

Description - the description of the group.

Start Date - the date on which this direct group assignment became or will become inactive for the persona.

End Date - the date on which this direct group assignment became or will become inactive for the persona. If the re-approval flag is set, the shown end date is the date of the re-approval. You can change the end date to shorten the period to the next re-approval but you cannot lengthen it.

Reapproval - whether (checked) or not (unchecked) this assignment will require regular re-approval. The next re-approval will occur at the end date shown. Unchecking this flag removes the privilege from re-approval at the shown end date (re-approvals are no longer required after this date is reached).

Assigned by - how the assignment was made:

- (empty) - the group was assigned by a permission or role through privilege assignment.
- **BO** - the group was assigned by privilege inheritance from a business object.
- **manual** - the group was assigned by hand.
- **rule** - the group was assigned by a rule through policy execution.

State - the status of the group assignment. Possible values are:

- **ADD** - the assignment is requested by DirX Identity Provisioning and shall be created in the target system by the synchronization workflow.
- **DELETED** - DirX Identity Provisioning requests the assignment to be deleted in the target system by the synchronization workflow. When it is deleted in the target system, the workflow removes the assignment physically from the **dxrMemberDel** attribute.
- **ENABLED** - the assignment is established both in DirX Identity Provisioning and in the target system: they are "in-sync".
- **IMPORTED** - the assignment was created in the target system, but not yet requested in DirX Identity Provisioning. DirX Identity Provisioning does not delete it (ignores it), but it may be an indicator to the administrator, that the persona has access rights not

granted by the assigned roles. DirX Identity Provisioning switches the state to **ENABLED**, when the persona is granted this access right. If it is revoked afterwards, the state switches to **DELETED** and not back to **IMPORTED**!

See the section "Managing States" in the chapter "Managing Provisioning" to understand the relationship to the states of other objects.

Note that DirX Identity shows assignments in approval as gray lines. Valid values for such entries are:

- **Add** - the assignment is not yet active.
- **Mod** - parts of this assignment (some attributes) are to be changed.
- **Del** - the assignment is to be deleted.
- **INACTIVE** - the state is a virtual one that may have two causes:
- The end date of this assignment has passed.
- The privilege is already assigned (for example, by the policy execution service), but a privilege resolution has not yet run. Run the privilege resolution to solve this problem.
- The privilege is already assigned but the assignment is still in approval. Completing the approval workflow will change the state.

The reason for this state is that the dxrGroupLink at the persona is already populated, but the member attributes at the group or account are not yet populated.

To add or remove groups manually, use the assignment editor, which is visible when you click **Edit**. To examine the details of a particular group, click ... to the right of its row.

Note that a line is displayed in gray if this assignment is still in approval. Thus you can see a white line that defines the current state and maybe several gray lines that define changes for this assignment that are still in approval.

Related Topics

- Persona - General Properties
- Persona - Relationships
- Persona - Operational Properties
- Persona - Communication
- Persona - Authentication
- Persona - Organization
- Persona - Location
- Persona - Context
- Persona - Assigned Roles
- Persona - Assigned Permissions
- Persona - Accounts
- Persona - Orders

- Persona - SoD Exceptions

A.1.3.12. Persona - Accounts

Use this tab to display the list of accounts currently assigned to the persona.

For each account, the following properties are shown:

Account - the account's name as it is displayed throughout the DirX Identity Provisioning system.

Description - the description of the account.

State - the status of the account. Possible values are:

- **ENABLED** - the account shall exist, be created in the target system and be enabled.
- **DISABLED** - the account shall exist, be created in the target system and be disabled.
- **DELETED** - the account shall be deleted in the target system.
- **IMPORTED** - the account was created in the target system, but is not yet requested by DirX Identity Provisioning.

Target System - the target system to which this account belongs.

State in TS (target system) - the status of the account in the respective target system. This value may differ from the status of the account entry in DirX Identity Provisioning as long as the information is not synchronized. Possible values are:

- **ENABLED** - the account exists in the target system and is enabled.
- **DISABLED** - the account exists in the target system and is disabled.
- **DELETED** - the object has been deleted in the target system without DirX Identity Provisioning having requested it! This state is only set by the validation workflow together with a message in the dxrToDo attribute, remembering the administrator to inspect this situation.
- **NONE** - replaces the empty value when the object is created by the DirX Identity Manager or agent.

  See the section "Managing States" in the chapter "Managing Provisioning" to understand the relationship to the states of other objects.

Priv - whether the account is an assigned privileged account (checked) or a personal account that is a member in the assigned groups that define the access rights in the connected systems (unchecked). A privileged account can be used by many persons in parallel. Examples are the Administrator account in Windows or the root account in UNIX. Assignment of a privileged account means:
1) You can read the password of this account in clear text to log in to the corresponding target system.
2) Your certificates are copied to the account in the connected system, which allows you to log in with the corresponding card.

You cannot make direct assignment of accounts in this tab. To examine the details of a particular account, click ... to the right of the respective table row.

Related Topics

- Persona - General Properties
- Persona - Relationships
- Persona - Operational Properties
- Persona - Communication
- Persona - Authentication
- Persona - Organization
- Persona - Location
- Persona - Context
- Persona - Assigned Roles
- Persona - Assigned Permissions
- Persona - Assigned Groups
- Persona - Orders
- Persona - SoD Exceptions

A.1.3.13. Persona - Orders

Use this tab to display all currently pending changes for this persona. This information is kept in order objects (mostly part of running request workflows).

The upper part of this tab shows pending attribute modification requests, the lower part shows pending privilege assignments or privilege assignment attribute changes.

The **Attribute Modifications** pane shows a line for each relevant attribute. The columns are:

Due Date - the date on which this change will be valid. If no value is supplied, the change depends on a pending attribute approval step of a running request workflow.

Attribute - the name of the attribute.

Old values - the old value(s) of the attribute.

New values - the new value(s) of the attribute.

... - opens a detail page for this line.

The **Assignments to add (+), modify (*) or delete (-)** pane shows a line for each assignment or change. The columns are:

Op(eration) - the type of line. Possible values are:

(+) - the privilege is to be assigned (to be added).

(*) - the assignment is subject to a change. One or more of the attributes - for example, the start or end date or a role parameter - has changed.

(-) - the privilege is to be removed (to be deleted).

Due date - the date on which this assignment or assignment change will be valid. If no value is supplied, the change depends on a pending attribute approval step of a running request workflow.

Type - the privilege type (role, permission or group).

Name - the privilege name.

Attribute - the name of the attribute to be changed.

Old values - the old value(s) of the attribute.

New values - the new value(s) of the attribute.

Assigned by - the type of the assignment (manual or by rule).

- opens a detail page for this line. It shows the corresponding request workflow instance.

Note that the assignment information is also visible in the Assigned Roles, Assigned Permissions and Assigned Group tabs in a slightly different format.

Related Topics

- Persona - General Properties
- Persona - Relationships
- Persona - Operational Properties
- Persona - Communication
- Persona - Authentication
- Persona - Organization
- Persona - Location
- Persona - Context
- Persona - Assigned Roles
- Persona - Assigned Permissions
- Persona - Assigned Groups
- Persona - Accounts
- Persona - SoD Exceptions

A.1.3.14. Persona - SoD Exceptions

Use this tab to display all segregation of duty (SoD) exceptions for this persona. SoD exceptions are SoD conflicts that have been approved successfully via an assignment approval workflow.

Note that the scope of SOD checking is limited to single users or personas only. Thus. if a persona has a privilege assigned that violates an SOD policy for another privilege being assigned to its user or to another persona of its user, no violation is detected.

The upper part of this tab shows approved SoD violations, the lower part shows pending SoD violations.

For each SoD exception line in the **Approved SoD violations** table, the following columns are shown:

Policy - the SoD policy that caused this exception.

Privilege - the privilege that was assigned to the persona.

Activity - the approval activity of the corresponding request workflow.

Reason - the reason why the approver accepted this SoD conflict.

Who - the approver.

When - the date of approval.

For each SoD exception line in the **Pending SoD violations** table, the following columns are shown:

Policy - the SoD policy that caused this exception.

Privilege - the privilege that was assigned to the persona.

Click … to open a detail page for this line.

Related Topics

- Persona - General Properties
- Persona - Relationships
- Persona - Operational Properties
- Persona - Communication
- Persona - Authentication
- Persona - Organization
- Persona - Location
- Persona - Context
- Persona - Assigned Roles
- Persona - Assigned Permissions
- Persona - Assigned Groups
- Persona - Accounts
- Persona - Orders

A.1.3.15. Persona - Risk Parameters

Use this tab to display the risk assessment properties for the persona.

The **Overall Risk** section displays the computed overall risk level and the compound score. It includes the following properties:

Risk Level - the overall risk level for the persona. Possible values are **Low**, **Medium** or **High**.

Compound Score - the calculated compound score for the persona. All of the persona's standard scores are used to compute the compound score.

The **Risk Factors** section provides the values for every risk factor defined in the domain's risk policy. Each risk factor is listed by name; for example, **GroupMemberships**, **ImportedAccounts**, **PrivilegedAccounts**. Risk factors shown as **RiskFactor*_number_ - for example, *RiskFactor6** - represent risk factors that have not been configured and have so not been processed by the Risk Calculation workflow. For every risk factor, the following properties are shown:

Raw Value - the raw value for the risk factor. This value is the computed score of the factor at the persona.

Standard Score - the standard score for the risk factor. This value is a normalized score for the factor at the persona.

See the section "Managing Risk" in the chapter "Managing Compliance" for details on risk calculation.

Related Topics

- Persona - General Properties
- Persona - Relationships
- Persona - Operational Properties
- Persona - Communication
- Persona - Authentication
- Persona - Organization
- Persona - Location
- Persona - Context
- Persona - Assigned Roles
- Persona - Assigned Permissions
- Persona - Assigned Groups
- Persona - Accounts
- Persona - Orders
- Persona - SoD Exceptions
- Risk Policy - General

A.1.4. Functional Users

A.1.4.1. Functional User - General Properties

A functional user is a resource that is assigned to an identity; for example, a global or group mailbox, a physical room with a phone or a working student entry. The functional user either manages or is responsible for these resources.

A functional user's sponsor attribute links it to the responsible user. DirX Identity Provisioning manages the functional user's access to all parts of the IT infrastructure.

The Functional User - General Properties tab shows all of the general properties of the functional user object. Since a domain administrator is allowed to configure all aspects of a functional user, the items shown here may vary from installation to installation or even from domain to domain. The domain administrator can also set up several functional user types that show a different set of properties.

The following properties describe the default configuration of DirX Identity. Note that the functional user's default properties have been taken almost completely from the user object, since the use of functional users and their properties varies from customer to customer.

General (Identification) Properties

Name (mandatory) - used as the relative distinguished name (RDN) for the functional user entry and identifies the functional user uniquely within the sub-tree. Once this property is set and saved during the addition of a new functional user entry, you can no longer change it by simply editing this field. Use the Rename command of the functional user entry's context menu instead.

First Name (optional) - the first name of the functional user.

Middle Name (optional) - the middle name of the functional user.

Salutation (optional) - the salutation of the functional user. Values for different countries are available.

Title (optional) - the title of this functional user.

Last Name (mandatory) - the last name of the functional user. This item corresponds to the "surname" attribute saved in the Identity Store.

Note that the use of first name and last name varies from customer to customer. Some customers use it to display the names of the related sponsor, others use it to describe the type of resource (mailbox, meeting room, trainee,...) with those attributes.

Gender (optional) - the gender of this functional user.

Day of birth (optional) - the date of birth for this functional user.

Master (optional) - the name of the directory that masters the functional user entry.

Employee number (optional) - the employee number.

Identifier (optional) - the functional user's global unique identifier.

Description (optional) - the description for the functional user. This field is often useful when functional user entries are listed in tables just with their names but with a description column to better identify them.

Employee Type (optional) - one of the following functional user's employee types:

- **Contractor** - an (outside) contractor.
- **Customer** - a customer.
- **Internal** - an internal employee.
- **Supplier** - a supplier.

Business Category (optional) - the business for which the functional user works.

Related Topics

- Functional User - Relationships
- Functional User - Operational Properties
- Functional User - Communication
- Functional User - Authentication
- Functional User - Organization
- Functional User - Location
- Functional User - Context
- Functional User - Assigned Roles
- Functional User - Assigned Permissions
- Functional User - Assigned Groups
- Functional User - Accounts
- Functional User - Orders
- Functional User - SoD Exceptions
- Functional User - Risk Parameters

A.1.4.2. Functional User - Relationships

Use this tab to define the relationship of this functional user to other functional user objects.

Owner (optional) - the owner of the functional user entry. The standard attribute to reference its responsible user is the sponsor attribute, so you may use the owner attribute for own purposes.

Manager (optional) - the functional user's manager. Used for functional user type Internal

Employee.

Secretary (optional) - the functional user's secretary.

Representative (optional - the functional user's representative.

Sponsor (mandatory) - the functional user's responsible person.

Note that these links are very useful when setting up participants for request workflows. For more information, see the section "Participant Calculation" in the *DirX Identity Application Development Guide*.

Related Topics

- Functional User - Operational Properties
- Functional User - Communication
- Functional User - Authentication
- Functional User - Organization
- Functional User - Location
- Functional User - Context
- Functional User - Assigned Roles
- Functional User - Assigned Permissions
- Functional User - Assigned Groups
- Functional User - Accounts
- Functional User - Orders
- Functional User - SoD Exceptions
- Functional User - Risk Parameters

A.1.4.3. Functional User - Operational Properties

Use this tab to display the operational parameters necessary for correct operation within DirX Identity.

Status fields

To be analyzed (read-only) - indicates when checked that this entry has been changed. For example, an import workflow added or modified some attributes. The privilege resolution process and the consistency checker uses this flag to detect and resolve records that are not up to date. They reset this flag after completing the check or resolution operation.

Is Inconsistent - indicates whether (unchecked) or not (checked) the stored access rights of the functional user are consistent with the assigned privileges. DirX Identity Provisioning checks the flag (sets the entry's flag to TRUE), if it fails to resolve the assigned privileges due to errors in the privilege structure. In this case, the functional user keeps the previous access rights, but DirX Identity Provisioning sets an end date for this state: the **Error expiration date**. The resolution error is stored in the **Error** field so that the administrator has

time to repair the error. The flag is unchecked if the entry is consistent.

Use as Template - indicates when checked that DirX Identity will not resolve privileges assigned to this functional user. A likely reason for the box to be checked is that the functional user object has been copied from another object. In this case, the functional user is marked as a template until appropriate changes have been made to the copied functional user. If the functional user is set correctly, uncheck the flag to enforce privilege resolution and target system provisioning.

+ A template functional user may also have been deliberately created to be used as a basis for creating additional accounts that are fully controlled by DirX Identity. Use the sponsor link to point from this copied functional user to its main identity (user).

Status - shows the status of the functional user entry. The field can be one of the following values (for more information, see the section "Managing States" in the chapter "Managing Provisioning"):

- **TEMPLATE** - the settings done for this functional user are used as a template.
- **NEW** - it is a new functional user, the privilege assignments are not yet activated.
- **ENABLED** - the functional user is established; the corresponding functional user object is up to date and valid, the privilege assignments are active.
- **DISABLED** - the privilege assignments for this functional user are currently disabled, the functional user does not have any rights in the domain.
- **TBDEL** - the functional user object shall be deleted from the DirX Identity Provisioning system. This operation is performed when all assigned accounts in the target systems are deleted or the end date for deletion is reached.

Lifetime Start - shows the date on which the privilege assignments to the functional user will become active for the first time. This is normally the case when the resource modeled by the functional user becomes available.

Lifetime End - shows the date on which the privilege assignments to the functional user will become inactive for the last time (usually the date when the functional user is removed since the resource modeled by the functional user entry does no longer exist).

Deactivation Start - shows the date on which the privilege assignments will become inactive for the next time.

Deactivation End - shows the date on which the privilege assignments will be reactivated again for the next time.

Delete - shows the date when the entry will be finally deleted despite of existing accounts in target systems.

Create time stamp - shows the date and time at which this entry was created.

Modify time stamp - shows the date and time at which this entry was last changed.

Notification fields

Notification Level - controls suppression of the first e-mail of the e-mail notification of an approval workflow:

- **0** - the first e-mail is sent.
- **1** - the first e-mail is not sent (but all other administrative or repetition e-mails).

Tasks fields

To do - shows a list of manual working items for the administrator. DirX Identity Provisioning stores the results of consistency checks it cannot repair automatically. The items are overwritten upon the next consistency check.

+ Possible messages in this field besides warnings from the service agent are:
INF:530:Missing values for attributes: *attributelist*.*
INF:531:More than one peer account exists for account DN=*dn.*
INF:532:No peer account exists for account DN=*dn.*

Error - stores a list of error messages that provide the detailed reason for inconsistent states or **To do** entries. They are cleared automatically when a privilege resolution succeeds.

Error expiration date - shows the date on which the access rights of the functional user will be updated despite any resolution errors. See the **Is Inconsistent** field.

Related Topics

- Functional User - Relationships
- Functional User - Communication
- Functional User - Authentication
- Functional User - Organization
- Functional User - Location
- Functional User - Context
- Functional User - Assigned Roles
- Functional User - Assigned Permissions
- Functional User - Assigned Groups
- Functional User - Accounts
- Functional User - Orders
- Functional User - SoD Exceptions
- Functional User - Risk Parameters

A.1.4.4. Functional User - Communication

Use this tab to display communication attributes for a functional user object.

E-Mail (optional) - the functional user's e-mail addresses.

Phone (optional) - the functional user's main phone number (business phone number).

Fax (optional) - the functional user's fax number.

Mobile (optional) - the functional user's mobile number.

Home (optional) - the functional user's home phone number.

Web address (optional) - the Web address of the Internet home page for this functional user.

Preferred language (optional) - the preferred language of this functional user, which controls, for example, the language used in an e-mail if this functional user is referenced in the To field. For more information, see the chapter "Using Variable Substitution" in the *DirX Identity Application Development Guide*.

Note that the phone number might be used by default when DirX Identity Provisioning creates a new account for the functional user. The account name is generated using the functional user's surname, given name and phone number. See the object description for the accounts.

Related Topics

- Functional User - Relationships
- Functional User - Operational Properties
- Functional User - Authentication
- Functional User - Organization
- Functional User - Location
- Functional User - Context
- Functional User - Assigned Roles
- Functional User - Assigned Permissions
- Functional User - Assigned Groups
- Functional User - Accounts
- Functional User - Orders
- Functional User - SoD Exceptions
- Functional User - Risk Parameters

A.1.4.5. Functional User - Authentication

Use this tab to display the authentication properties for this functional user.

Encrypted Password - the encrypted (or scrambled) password of this functional user (dxmPassword attribute). This attribute is used for password synchronization to target systems. Authentication against the Identity Store is performed via the stored one-way password (userPassword attribute).

Password Account Locked Time (read-only) - the time when the account was locked.

Password Changed Time (read-only) - the time at which the password was last changed.

Password Expiration Notified (read-only) - the time at which the Password Notification service notified the system about the password expiration.

Password Failure Time (read-only) - the time at which the password was entered incorrectly.

Password History (read-only) - the password history (one-way encrypted). This field shows only that there is a password history.

Password Reset (read-only) - if active, the password was reset by the administrator.

Password Policy - a link to the relevant password policy. If no link is set, the default policy is used by the system.

Challenges and Responses (read-only) - the challenge and response questions (one-way encrypted). This field shows only that challenge and response questions have been set up.

Certificate (read-only) - the certificate of the functional user.

Related Topics

- Functional User - Relationships
- Functional User - Operational Properties
- Functional User - Communication
- Functional User - Organization
- Functional User - Location
- Functional User - Context
- Functional User - Assigned Roles
- Functional User - Assigned Permissions
- Functional User - Assigned Groups
- Functional User - Accounts
- Functional User - Orders
- Functional User - SoD Exceptions
- Functional User - Risk Parameters

A.1.4.6. Functional User - Organization

Use this tab to display the organization and organizational units properties of a functional user. These properties can be linked with the relevant business objects.

Organizations

Organization Link (optional) - sets a link to an organization object (business object).

Organization (optional) - defines the organization to which the functional user belongs. If

the **Organization Link** is set, this attribute is read-only and is controlled by the linked object.

More Organizations (optional) - defines more organization links for this functional user.

Organizational Units

Org. Unit Link (optional) - sets a link to an organizational unit object (business object).

Organizational Unit (optional) - the organizational unit to which the functional user belongs. If the **Org. Unit Link** is set, this attribute is read-only and is controlled by the linked object.

Department Number (optional) - the department number of the organizational unit. If the **Org. Unit Link** is set, this attribute is read-only and is controlled by the linked object.

More Organizational Units (optional) - defines more organizational unit links for this functional user.

Related Topics

- Functional User - Relationships
- Functional User - Operational Properties
- Functional User - Communication
- Functional User - Authentication
- Functional User - Location
- Functional User - Context
- Functional User - Assigned Roles
- Functional User - Assigned Permissions
- Functional User - Assigned Groups
- Functional User - Accounts
- Functional User - Orders
- Functional User - SoD Exceptions
- Functional User - Risk Parameters

A.1.4.7. Functional User - Location

User this tab to display and set location information associated with the user.

Location Link (optional) - sets a link to a location object (business object).

Country (optional) - the country of the functional user. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

State (optional) - the state of the functional user. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

Postal Code (optional) - the postal code of the location. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

Location (optional) - the location(s) of the functional user. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

Street (optional) - the street name of the location. If the Location Link is set, this attribute is read-only and is controlled by the linked object.

Room (optional) - the room where this functional user is located.

Postal Address (optional) - the postal address of this location. If the Location Link is set, this attribute is read only and is controlled by the linked object.

More Locations (optional) - you can define more location links for this functional user with this editor.

Related Topics

- Functional User - Relationships
- Functional User - Operational Properties
- Functional User - Communication
- Functional User - Authentication
- Functional User - Organization
- Functional User - Context
- Functional User - Assigned Roles
- Functional User - Assigned Permissions
- Functional User - Assigned Groups
- Functional User - Accounts
- Functional User - Orders
- Functional User - SoD Exceptions
- Functional User - Risk Parameters

A.1.4.8. Functional User - Context

This tab is an example of other context information. Pre-configured links are:

Contexts (optional) - a link to any type of context object (business object).

Related Topics

- Functional User - Relationships
- Functional User - Operational Properties
- Functional User - Communication
- Functional User - Authentication

- Functional User - Organization
- Functional User - Location
- Functional User - Assigned Roles
- Functional User - Assigned Permissions
- Functional User - Assigned Groups
- Functional User - Accounts
- Functional User - Orders
- Functional User - SoD Exceptions
- Functional User - Risk Parameters

A.1.4.9. Functional User - Assigned Roles

Use this tab to view and manage the roles assigned to a functional user. The tab displays the list of roles currently assigned to the functional user.

For each role, the following properties are shown:

Role - the role's name as it is displayed throughout the DirX Identity Provisioning system. If the role's name ends with an asterisk character (*), the role has assigned one or more junior roles. These junior roles are automatically assigned to the user as well.

Description - the description for the role.

Start Date - the date on which this role became or will become active for the functional user; that is, the date on which the functional user was/is granted the permissions assigned to this role.

End Date - the date on which this role became or will become inactive for the functional user; that is, the date on which the permissions granted to the functional user was or will be revoked. If the re-approval flag is set, the shown end date is the date of the re-approval. You can change the end date to shorten the period to the next re-approval but you cannot lengthen it.

Reapproval - whether (checked) or not (unchecked) the assignment requires regular re-approval. The next re-approval will occur at the displayed end date. Unchecking this field causes the privilege to be removed at the displayed end date (re-approvals no longer occur).

Role Parameters - shows the first few assigned role parameters. Click ... to view all role parameter settings for this assignment.

Assigned by - the type of assignment:

- **BO** - the role was assigned by privilege inheritance from a business object.
- **manual** - the role was assigned by hand.
- **rule** - the role was assigned by a rule.

State - the state of the assignment:

- **ToBeApproved** - the assignment is still in approval. The corresponding access rights (groups) are not yet assigned.
- **ENABLED** - the assignment is enabled. The functional user has the corresponding access rights (groups) assigned.
- **DELETED** - the assignment is deleted. The corresponding access rights (groups) are not yet or no longer assigned.

  See the section "Managing States" in the chapter "Managing Provisioning" to understand the relationship to the states of other objects.

  Note that DirX Identity shows assignments in approval as gray lines. Valid values for such entries are:
- **Add** - the assignment is not yet active.
- **Mod** - parts of the assignment (some attributes) were changed.
- **Del** - the assignment is to be deleted.

To add or remove roles, use the assignment editor, which is displayed when you click **Edit**. To examine the details of a particular role, click ... to the right of its row.

Note that a line is displayed in gray if this assignment is still in approval. Thus, you can see a white line that defines the current state and maybe several gray lines that define changes for this assignment that are still in approval.

Related Topics

- Functional User - Relationships
- Functional User - Operational Properties
- Functional User - Communication
- Functional User - Authentication
- Functional User - Organization
- Functional User - Location
- Functional User - Context
- Functional User - Assigned Permissions
- Functional User - Assigned Groups
- Functional User - Accounts
- Functional User - Orders
- Functional User - SoD Exceptions
- Functional User - Risk Parameters

A.1.4.10. Functional User - Assigned Permissions

Use this tab to view and manage the permissions assigned to a functional user. The tab displays the list of permissions currently assigned to the functional user.

For each permission, the following properties are displayed:

Permission - the permission's name as it is displayed throughout the DirX Identity Provisioning system.

Description - the description of the permission.

Start Date - the date on which this permission became or will become active for the functional user.

End Date - the date on which this permission became or will become inactive for the functional user. If the re-approval flag is set, the end date displayed is the date of the re-approval. You can change the end date to shorten the period to the next re-approval but you cannot lengthen it.

Reapproval - whether (checked) or not (unchecked) this assignment requires regular re-approval. The next re-approval occurs at the shown end date. If you uncheck this flag, privilege re-approval is removed at the shown end date (re-approvals no longer occur after the end date is reached).

Assigned by - how the permission assignment was made. Possible values are:

- (empty) - the permission is inherited from a role.
- **BO** - the permission was assigned by privilege inheritance from a business object.
- **manual** - the permission was assigned by hand.
- **rule** - the permission was assigned by a rule.

State - the state of the assignment:

- **ToBeApproved** - the assignment is still in approval. The corresponding access rights (groups) are not yet assigned.
- **INHERITED** - this assignment is inherited from a role assignment.
- **DELETED** - the assignment is deleted. The corresponding access rights (groups) are not yet or no longer assigned.

  See the section "Managing States" in the chapter "Managing Provisioning" to understand the relationship to the states of other objects.

  Note that DirX Identity shows assignments in approval as gray lines. Valid values for such entries are:
- **Add** - the assignment is not yet active.
- **Mod** - parts of this assignment (some attributes) are to be changed.
- **Del** - the assignment is to be deleted.

To add or remove permissions manually, use the assignment editor, which is visible when you click **Edit**. To examine the details of a particular permission, click … to the right of its row.

Note that a line is displayed in gray if this assignment is still in approval. Thus you can see a white line that defines the current state and maybe several gray lines that define changes for this assignment that are still in approval.

Related Topics

- Functional User - Relationships
- Functional User - Operational Properties
- Functional User - Communication
- Functional User - Authentication
- Functional User - Organization
- Functional User - Location
- Functional User - Context
- Functional User - Assigned Roles
- Functional User - Assigned Groups
- Functional User - Accounts
- Functional User - Orders
- Functional User - SoD Exceptions
- Functional User - Risk Parameters

A.1.4.11. Functional User - Assigned Groups

Use this tab to view and manage the groups assigned to a functional user. The tab displays the list of groups currently assigned to the functional user.

For each group, the following properties are shown:

Target system - the name of the target system to which the group belongs.

Group - the group's name as it is displayed throughout the DirX Identity Provisioning system.

Description - the description of the group.

Start Date - the date on which this direct group assignment became or will become inactive for the functional user.

End Date - the date on which this direct group assignment became or will become inactive for the functional user. If the re-approval flag is set, the shown end date is the date of the re-approval. You can change the end date to shorten the period to the next re-approval but you cannot lengthen it.

Reapproval - whether (checked) or not (unchecked) this assignment will require regular re-approval. The next re-approval will occur at the end date shown. Unchecking this flag removes the privilege from re-approval at the shown end date (re-approvals are no longer required after this date is reached).

Assigned by - how the assignment was made:

- (empty) - the group was assigned by a permission or role through privilege assignment.
- **BO** - the group was assigned by privilege inheritance from a business object.
- **manual** - the group was assigned by hand.
- **rule** - the group was assigned by a rule through policy execution.

State - the status of the group assignment. Possible values are:

- **ADD** - the assignment is requested by DirX Identity Provisioning and shall be created in the target system by the synchronization workflow.
- **DELETED** - DirX Identity Provisioning requests the assignment to be deleted in the target system by the synchronization workflow. When it is deleted in the target system, the workflow removes the assignment physically from the **dxrMemberDel** attribute.
- **ENABLED** - the assignment is established both in DirX Identity Provisioning and in the target system: they are "in-sync".
- **IMPORTED** - the assignment was created in the target system, but not yet requested in DirX Identity Provisioning. DirX Identity Provisioning does not delete it (ignores it), but it may be an indicator to the administrator that the persona has access rights not granted by the assigned roles. DirX Identity Provisioning switches the state to **ENABLED** when the persona is granted this access right. If it is revoked afterwards, the state switches to **DELETED** and not back to **IMPORTED**!

  See the section "Managing States" in the chapter "Managing Provisioning" to understand the relationship to the states of other objects.

  Note that DirX Identity shows assignments in approval as gray lines. Valid values for such entries are:
- **Add** - the assignment is not yet active.
- **Mod** - parts of the assignment (some attributes) are to be changed.
- **Del** - the assignment is to be deleted.
- **INACTIVE** - the state is a virtual one that may have two causes:
- The end date of this assignment has passed.
- The privilege is already assigned (for example, by the policy execution service), but a privilege resolution has not yet run. Run the privilege resolution to solve this problem.
- The privilege is already assigned, but the assignment is still in approval. Completing the approval workflow will change the state.

The reason for this state is that the dxrGroupLink at the functional user is already populated, but the member attributes at the group or account are not yet populated.

To add or remove groups manually, use the assignment editor, which is visible when you click **Edit**. To examine the details of a particular role, click … to the right of its row.

Note that a line is displayed in gray if this assignment is still in approval. Thus you can see a white line that defines the current state and maybe several gray lines that define changes for this assignment that are still in approval.

Related Topics

- Functional User - Relationships
- Functional User - Operational Properties
- Functional User - Communication
- Functional User - Authentication
- Functional User - Organization
- Functional User - Location
- Functional User - Context
- Functional User - Assigned Roles
- Functional User - Assigned Permissions
- Functional User - Accounts
- Functional User - Orders
- Functional User - SoD Exceptions
- Functional User - Risk Parameters

A.1.4.12. Functional User - Accounts

Use this tab to display the list of accounts currently assigned to the functional user.

For each account, the following properties are shown:

Account - the account's name as it is displayed throughout the DirX Identity Provisioning system.

Description - the description of the account.

State - the status of the account. Possible values are:

- **ENABLED** - the account shall exist, be created in the target system and be enabled.
- **DISABLED** - the account shall exist, be created in the target system and be disabled.
- **DELETED** - the account shall be deleted in the target system.
- **IMPORTED** - the account was created in the target system, but is not yet requested by DirX Identity Provisioning.

Target System - the target system to which this account belongs.

State in TS (target system) - the status of the account in the respective target system. This

value may differ from the status of the account entry in DirX Identity Provisioning as long as the information is not synchronized. Possible values are:

- **ENABLED** - the account exists in the target system and is enabled.
- **DISABLED** - the account exists in the target system and is disabled.
- **DELETED** - the object has been deleted in the target system without DirX Identity Provisioning having requested it! This state is only set by the validation workflow along with a message in the **dxrToDo** attribute reminding the administrator to inspect this situation.
- **NONE** - replaces the empty value when the object is created by the DirX Identity Manager or agent.

  See the section "Managing States" in the chapter "Managing Provisioning" to understand the relationship to the states of other objects.

Priv - whether the account is an assigned privileged account (checked) or a personal account that is a member in the assigned groups that define the access rights in the connected systems (unchecked). A privileged account can be used by many persons in parallel. Examples are the Administrator account in Windows or the root account in UNIX. Assignment of a privileged account means:
1) You can read the password of this account in clear text to log in to the corresponding target system.
2) Your certificates are copied to the account in the connected system, which allows you to log in with the corresponding card.

You cannot make direct assignment of accounts in this tab. To examine the details of a particular account, click … to the right of the respective table row.

Related Topics

- Functional User - Relationships
- Functional User - Operational Properties
- Functional User - Communication
- Functional User - Authentication
- Functional User - Organization
- Functional User - Location
- Functional User - Context
- Functional User - Assigned Roles
- Functional User - Assigned Permissions
- Functional User - Assigned Groups
- Functional User - Orders
- Functional User - SoD Exceptions
- Functional User - Risk Parameters

A.1.4.13. Functional User - Orders

Use this tab to display the currently pending changes for a functional user. This information is kept in order objects (which are mostly part of running request workflows).

The upper part of this tab shows pending attribute modification requests, while the lower part shows pending privilege assignments or privilege assignment attribute changes.

The **Attribute Modifications** pane shows a line for each relevant attribute. The columns are:

Due Date - the date on which this change will be valid. If no value is supplied, the change depends on a pending attribute approval step of a running request workflow.

Attribute - the name of the attribute.

Old values - the old value(s) of the attribute.

New values - the new value(s) of the attribute.

- opens a detail page for this line.

The **Assignments to add (+), modify (*) or delete (-)** pane shows a line for each assignment or change. The columns are:

Op(eration) - the type of line. Possible values are:

(+) - the privilege is to be assigned (to be added).

(*) - the assignment is subject to a change. An attribute - for example, the start or end date or a role parameter - has changed.

(-) - the privilege is to be removed (to be deleted).

Due date - the date on which this assignment or assignment change will be valid. If no value is supplied, the change depends on a pending attribute approval step of a running request workflow.

Type - shows the privilege type (role, permission or group).

Name - displays the privilege name.

Attribute - the name of the attribute to be changed.

Old values - the old value(s) of the attribute.

New values - the new value(s) of the attribute.

Assigned by - type of the assignment (manual or by rule).

- opens a detail page for this line, which shows the corresponding request workflow instance.

Note that the assignment information is also visible in the Assigned Roles, Assigned

Permissions and Assigned Group tabs in a slightly different format.

Related Topics

- Functional User - Relationships
- Functional User - Operational Properties
- Functional User - Communication
- Functional User - Authentication
- Functional User - Organization
- Functional User - Location
- Functional User - Context
- Functional User - Assigned Roles
- Functional User - Assigned Permissions
- Functional User - Assigned Groups
- Functional User - Accounts
- Functional User - SoD Exceptions
- Functional User - Risk Parameters

A.1.4.14. Functional User - SoD Exceptions

Use this tab to display all segregation of duty (SoD) exceptions for this functional user. SoD exceptions are SoD conflicts that have been approved successfully via an assignment approval workflow.

The upper part of this tab shows approved SoD violations, the lower part shows pending SoD violations.

For each SoD exception line in the **Approved SoD violations** table, the following columns are shown:

Policy - the SoD policy that caused this exception.

Privilege - the privilege that was assigned to the functional user.

Activity - the approval activity of the corresponding request workflow.

Reason - the reason why the approver accepted this SoD conflict.

Who - the approver.

When - the date of approval.

For each SoD exception line in the **Pending SoD violations** table, the following columns are shown:

Policy - the SoD policy that caused this exception.

Privilege - the privilege that was assigned to the functional user.

Click [...] to open a detail page for this line.

Related Topics

- Functional User - Relationships
- Functional User - Operational Properties
- Functional User - Communication
- Functional User - Authentication
- Functional User - Organization
- Functional User - Location
- Functional User - Context
- Functional User - Assigned Roles
- Functional User - Assigned Permissions
- Functional User - Assigned Groups
- Functional User - Accounts
- Functional User - Orders
- Functional User - Risk Parameters

A.1.4.15. Functional User - Risk Parameters

Use this tab to display the risk assessment properties for the functional user.

The **Overall Risk** section displays the computed overall risk level and the compound score. It includes the following properties:

Risk Level - the overall risk level for the functional user. Possible values are **Low**, **Medium** or **High**.

Compound Score - the calculated compound score for the functional user. All of the functional user's standard scores are used to compute the compound score.

The **Risk Factors** section provides the values for every risk factor defined in the domain's risk policy. Each risk factor is listed by name; for example, **GroupMemberships**, **ImportedAccounts**, **PrivilegedAccounts**. Risk factors shown as **RiskFactor*_number_ - for example, *RiskFactor6** - represent risk factors that have not been configured and have so not been processed by the Risk Calculation workflow. For every risk factor, the following properties are shown:

Raw Value - the raw value for the risk factor. This value is the computed score of the factor at the functional user.

Standard Score - the standard score for the risk factor. This value is a normalized score for the factor at the functional user.

See the section "Managing Risk" in the chapter "Managing Compliance" for details on risk calculation.

Related Topics

- Functional User - Relationships
- Functional User - Operational Properties
- Functional User - Communication
- Functional User - Authentication
- Functional User - Organization
- Functional User - Location
- Functional User - Context
- Functional User - Assigned Roles
- Functional User - Assigned Permissions
- Functional User - Assigned Groups
- Functional User - Accounts
- Functional User - Orders
- Functional User - SoD Exceptions
- Risk Policy - General

A.1.5. Assignment - General

Use this tab to display the general parameters of an assignment between a user and a role.

For each assignment, the following properties are shown:

User - the link to the user to which the privilege is assigned (read-only).

Privilege - the link to the privilege that is assigned to the user (read-only).

State - the state of the assignment (read-only):

- **ToBeApproved** - the privilege is assigned, but the corresponding workflow has not started yet (**Save** has not been clicked).
- **Add** - the privilege has still to be added. The corresponding approval workflow has not yet finished.
- **Modify** - the privilege has still to be modified. The corresponding approval workflow has not yet finished.
- **Delete** - the privilege has still to be deleted. The corresponding approval workflow has not yet finished.
- **ENABLED** - the assignment is enabled.

Assigned by - the type of assignment (read-only):

- **BO** - the role was assigned by privilege inheritance from a business object.
- **manual** - the role was assigned by hand.
- **rule** - the role was assigned by a rule.

Start date - the date on which this role became or will become active for the user; that is, when the user was/is granted the permissions assigned to this role.

End date - the date on which this role became or will become inactive for the user; that is, when the permissions granted to the user was or will be revoked.

Note: Lines that are still in approval are shown in gray. For a specific privilege, several lines can exist (for example, the actual assignment and several pending modifications or even a deletion).

Related Topics

- Assignment - RoleParams
- User - General Properties

A.1.6. Assignment - RoleParams

Use this tab to set all role parameters.

Set all parameters to the required values by entering values or selecting them from a drop-down list.

If the values are not correct, a traffic sign indicates the error. Move the mouse cursor over the sign to display a tooltip that explains the error in detail.

Parameters can be either single-valued or multi-valued. Text parameters can be optional. All other parameters are mandatory and always require setting a value.

Click **OK** to save the parameters, click **Cancel** to abort editing these parameters.

Related Topics

- Assignment - General
- User - General Properties

A.1.7. Country - General Properties

Use this tab to display the general properties for a country folder. The country folder contains one or more organization folders.

The tab displays the following items:

Name - the displayed name of the country folder.

Description - the description for the country folder.

Related Topics

Organization

A.1.8. Domain Component - General Properties

Use this tab to display the general properties for a domain component folder. The domain component folder contains either user entries or ordering folders corresponding to other domain components, organizations or organizational units (departments, teams).

The tab displays the following items:

Name - the displayed name of the domain component folder.

Description - the description for the domain component folder.

Related Topics

- User
- Organization
- Organizational Unit

A.1.9. Locality - General Properties

Use this tab to display the general properties for a locality folder. The locality folder contains either user entries or folders corresponding to organizational units.

The items shown in this tab include:

Name - the displayed name of the locality folder.

Description - the description for the locality folder.

Related Topics

- User
- Organizational unit

A.1.10. Organization - General Properties

Use this tab to display the general properties for a folder that separates user entries for different organizations. The folder can contain user entries or other ordering folders that correspond to localities or organizational units like departments or teams.

The properties shown in this tab include:

Name - the displayed name of the organization folder.

Description - the description for the organization folder.

Related Topics

- User
- Organizational Unit
- Locality

A.1.11. Organizational Unit - General Properties

Use this tab to display the general properties for a folder that separates user entries for different organizational units. The folder can contain user entries or folders that correspond to other organizational units.

The properties shown in this tab include:

Name - the displayed name of the organizational unit folder.

Description - the description for the organizational unit folder.

Related Topics

- User

A.2. Business Objects View

A.2.1. Business Objects

This tab shows the general properties for a business objects folder.

The items shown here include:

Name - the displayed name of the folder.

Description - the description for the folder.

Related Topics

- Managing Business Objects
- Context
- Cost Unit
- Country
- Location
- Organization
- Organizational Unit
- Project

A.2.2. Context

The context object is a generic item that can serve as any object. You can use this object's items or you can define your own specific items.

The items shown here include:

Name - the displayed name of the object.

Description - the description for the object.

Type - the type of this object.

Status - the status of the object.

Managers - the managers of this context object.

References - pre-configured references to other business objects or even other objects.

Note that you cannot assign roles with role parameters or privileges that are marked for approval to business objects. Web Center prohibits assignment of such privileges. Manager and **metacp** do not protect against the assignment of such privileges.

Related Topics

- Managing Business Objects
- Cost Unit
- Country
- Location
- Organization
- Organizational Unit
- Project

A.2.3. Cost Unit

Use the cost unit object to design cost-related organizational items.

The items shown here are the following:

Name - the displayed name of the object.

Description - the description for the object.

Status - the status of the object.

Managers - the managers of this cost unit.

References - pre-configured references to other business objects or even other objects.

Related Topics

- Managing Business Objects
- Context
- Country

- Location
- Organization
- Organizational Unit
- Project

A.2.4. Country

The items shown here include:

Country - the displayed name of the object.

Description - the description for the object.

References - pre-configured references to other business objects or other objects.

Related Topics

- Managing Business Objects
- Context
- Cost Unit
- Location
- Organization
- Organizational Unit
- Project

A.2.5. Location

The items shown here include:

Name - the displayed name of the object.

Description - the description for the object.

Status - the status of the object.

Postal Address - the postal address of the location.

Postal Code - the postal code of the location.

Street - the street of the location.

State - the state of the location.

Managers - the managers of this location.

References - pre-configured references to other business objects or even other objects.

Related Topics

- Managing Business Objects
- Context
- Cost Unit
- Country
- Organization
- Organizational Unit
- Project

A.2.6. Organization

The items shown here include:

Organization - the displayed name of the object.

Description - the description for the object.

Status - the status of the object.

Postal Address - the postal address of the organization.

Postal Code - the postal code of the organization.

Street - the street of the organization.

State - the state of the organization.

Managers - the managers of this organization.

References - pre-configured references to other business objects or even other objects.

+ Note that you cannot assign roles with role parameters or privileges that are marked for approval to business objects. Web Center prohibits assignment of such privileges. Manager and **metacp** do not protect against the assignment of such privileges.

Related Topics

- Managing Business Objects
- Context
- Cost Unit
- Country
- Location
- Organizational Unit
- Project

A.2.7. Organizational Unit

The items shown here include:

Name - the displayed name of the object.

Description - the description for the object.

Status - the status of the object.

Department Number - the department number or short name of the organizational unit.

Postal Address - the postal address of the organizational unit.

Postal Code - the postal code of the organizational unit.

Street - the street of the organizational unit.

Managers - the managers of this organizational unit.

References - pre-configured references to other business objects or even other objects.

Note that you cannot assign roles with role parameters or privileges that are marked for approval to business objects. Web Center prohibits assignment of such privileges. Manager and **metacp** do not protect against the assignment of such privileges.

Related Topics

- Managing Business Objects
- Context
- Cost Unit
- Country
- Location
- Organization
- Project

A.2.8. Project

The items shown here include:

Name - the displayed name of the object.

Description - the description for the object.

Status - the status of the object.

Type - the project type.

Managers - the managers of this project.

References - pre-configured references to other business objects or even other objects.

Related Topics

- Managing Business Objects
- Context
- Cost Unit
- Country
- Location
- Organization
- Organizational Unit

A.3. Ticket View

A.3.1. Ticket Folder

The items shown here include:

Name - the displayed name of the object.

Description - the description for the object.

Related Topics

- Ticket - General Properties
- Ticket - Status Information
- Ticket - Object
- Ticket - Assignments

A.3.2. Ticket - General Properties

The ticket object represents an object that holds temporary information; for example, the data for a scheduled change in the future. Specific processes are available that manage ticket objects.

Use this tab to display the general properties of a ticket object. Since the domain administrator can configure all ticket object properties, the items shown here may vary from installation to installation or even from domain to domain. The domain administrator can also set up several ticket types that show a different set of properties.

The following properties describe the default configuration provided with DirX Identity:

Name - the name of the object (the ticket).

Object type - the object type of the ticket's subject.

Operation - the operation that will be performed when the ticket is processed.

Owner - the person who created this ticket (who owns this ticket).

Request workflow (optional) - a corresponding request workflow that was started directly after ticket creation.

Subject - the object that is affected by this ticket.

Resources - resources that are to be assigned to the subject via this ticket.

Related Topics

- Ticket - Status Information
- Ticket - Object
- Ticket - Assignments

A.3.3. Ticket - Status Information

Use this tab to display status information about a ticket object. Fields include:

Ticket state - the current state of the ticket.

Due Date - the date on which the ticket is to be processed.

Expiration date - the time at which this ticket expires. This value is calculated by the ticket creator and can be empty.

End Date - the date on which the ticket was processed.

Status Expiration - the lifetime of this ticket instance. Set up a cleanup consistency workflow to delete these instances from time to time.

Error messages - a list of problems that occurred during ticket processing.

Related Topics

- Ticket - General Properties
- Ticket - Object
- Ticket - Assignments

A.3.4. Ticket - Object

Use this tab to view the set of attributes to be modified for this object (the subject). Columns include:

Due date - the date when this change will be performed. If empty, the change is performed immediately after approval.

Attribute - the name of the attribute.

Old value - the old value.

New value - the new value.

Note that the operation is indirectly defined by the value combination:

Add - old value is empty, new value is present.

Modify - both values are present.

Delete - old value is present, new value is empty.

Related Topics

- Ticket - General Properties
- Ticket - Status Information
- Ticket - Assignments

A.3.5. Ticket - Assignments

Use this tab to display the assignments of resources to this subject that are to be modified. Columns include:

Op - the operation: '+' stands for add, '-' for remove and 'o' for modify.

Due date - the date on which this change will be performed. If empty, the change is performed immediately after approval.

Type - the type of resource (privilege).

Name - the name of the resource.

Attribute - the attribute name.

Old values - the old value of a resource attribute.

New values - the new value of a resource attribute.

Assigned by - the type of assignment (manual or by rule).

Related Topics

- Ticket - General Properties
- Ticket - Status Information
- Ticket - Object

A.4. Privileges View

The **Privileges** view shows the set of roles and permissions currently in use and the user groups and their organizing folders for each target system.For each object type, a set of query folders may also be shown for filtering a subset of the respective object type according to various criteria.

For each selected item in the tree, the respective property pages are shown on the right-hand side.Click **Edit** to modify property values.Note that saving the properties of an entry will result in another resolution of its relationships to other objects.

Related Topics

- Role - General Properties
- Permissions
- Group Target Systems View

A.4.1. Roles

This tab shows the general properties for a role folder. The role folder contains either role entries or other role folders for improving the ordering and grouping of roles.

The property items shown here are the following:

Name - the displayed name of the roles folder.

Description - the description for the roles folder.

Related Topics

- Role - General Properties

A.4.1.1. Role - General Properties

The role object represents a role, a set of permissions needed to perform a particular task in an appropriate personal function of an enterprise, completely managed by the DirX Identity Provisioning system.

Use this tab to display all general properties of the role object. General properties include:

Identification Properties

Name - (mandatory) the name of the role as it is used for display and identification purposes throughout the DirX Identity Provisioning system. Once set during adding a new role entry, it can no longer be changed by simply editing this field after the initial save operation. Use the **Rename** command of the user entry's context menu instead.

Description - (optional) a description for the role. This is often useful when role entries are listed in tables just with their names but also with a description column to better identify them.

Role ID - (optional) a customer-specific identifier for the role.

User assignment possible - whether (checked) or not (unchecked) the role can be assigned to a user entry.

Once-only assignment - whether (checked) or not (unchecked) a user can assign the role just once even if it has role parameters.

Owner - the owners of this privilege. These entities are usually the persons that must approve if this privilege is assigned to a user.

Task fields

To do - a list of manual working items for the administrator. DirX Identity Provisioning stores the results of consistency checks it cannot repair automatically. The items are overwritten upon the next consistency check.

Error - a list of error messages that show the detailed reason for inconsistent states or **To do** entries. They are cleared automatically when a privilege resolution succeeds.

To be analyzed - a flag for the consistency workflow. When checked, the role has been changed and a privilege resolution must be performed .This flag is read-only for the administrator and is set and reset automatically by DirX Identity Provisioning workflows.

Related Topics

- Approval
- Certification
- Details
- Role Parameters
- Junior Roles
- Assigned Permissions
- Senior Roles
- Users

A.4.1.2. Role - Approval

Use this tab to view and manage the properties of the role object that define the approval process to be used when the role is assigned to another object (for more information about this feature, see the section Managing Request Workflows). These properties include:

For Assignment to Users - fields that apply to the assignment of a role to a user object. These fields include:

Requires approval - whether (checked) or not (unchecked) assigning this role to a user requires approval via request workflows.

Potential SoD conflict (read-only) - whether (checked) or not (unchecked) the role is part of one or more SoD policies, which means that a conflicting privilege exists.

Approval Workflows for

Assignment - the request workflow to be used for approval if this privilege is assigned to a user. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).

Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Modification - the request workflow to be used for approval if this privilege assignment is changed. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Removal - the request workflow to be used for approval if this privilege assignment is removed. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

SoD - the request workflow to be used for approval if, during assignment of this privilege, an SoD violation is detected. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

For Assignment to Senior Roles - fields that apply to the assignment of a role as a junior role to another role object (the senior role). These fields include:

Requires approval - whether (checked) or not (unchecked) assigning this role as a junior role to another role requires approval via request workflows.

Approval Workflows for

Assignment - the request workflow to be used for approval if this privilege is assigned as a junior role to another role. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Removal - the request workflow to be used for approval if this role is removed from another role. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the

workflow definition.

Related Topics

- Assignment Editor
- General Properties
- Certification
- Details
- Role Parameters
- Assigned Permissions
- Senior Roles
- Users

A.4.1.3. Role - Re-approval

Use this tab to view and manage the properties of the role object that define re-approval scenarios. These properties include:

Requires re-approval - whether (checked) or not (unchecked) all assignments of the role should be regularly re-approved. The DirX Identity re-approval process starts a re-approval workflow at the intervals specified in the other fields in this area.

Re-approval date - the next date for re-approval. Alternatively you can define a re-approval period.

Re-approval period - the frequency with which the re-approval workflows should be started. This value is only used when **Re-approval date** is not set.

Note: If neither **Re-approval date** nor **Re-approval period** are set, the default values from the domain object are used.

Workflow - the request workflow to be used for re-approval. If this field is left blank, the request workflow engine searches for a suitable workflow template. (See the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details.)
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Related Topics

- Assignment Editor
- General Properties
- Approval
- Details
- Role Parameters

- Assigned Permissions
- Senior Roles
- Users

A.4.1.4. Role - Details

Use this tab to view and manage the general properties of the role object. These properties include:

Sub Tasks - the list of activities to reach the global goal of a role. These are activities that users must perform when they are assigned to the role. The activities should be relevant for access management.

Responsibilities - the duties and responsibilities for a user assigned to this role (for example, controlling of budget, or signing of contracts).

Role Engineer - the **Role Engineer** (user or team) who is currently responsible for the specification of this role.

Role Admin - the **Role Administrator** (user or team) who is currently responsible for implementing the results of the life-cycle management processes for this role.

Role class (optional) - the type of the role. Possible values include:

- **Basic** - a general value for the role class.
- **Functional** - marks the role as a functional role.
- **Special** - marks the role as a special role (in terms of the customer's use).
- **Misc** - assigns a class value that is different from the other values given here.

Version - the role definition's version.

Custom State - the part of the life-cycle in which the role currently resides.

Language (optional) - the language for the role. Possible values are:

- **De(German)**
- En(English)

Reference (optional) - an attribute to be used in customer-specific environments.

Related Topics

- General Properties
- Approval
- Certification
- Role Parameters
- Junior Roles

- Assigned Permissions
- Senior Roles
- Users

A.4.1.5. Role - Role Parameters

Use this tab to view and manage match rules for role parameters to be used when this role is assigned to a user. You can define a set of match rules. Note that these match rules are added to the permission match rules when this role is assigned to a user.

When in **Edit** mode, add new lines to the list, edit existing ones or delete lines in the list.

For each line, the following properties are shown:

Name - the name of the role parameter. This is a link that points to a role parameter object in the Domain Configuration view.

Operator - the operator expression used for matching. Click the arrow button when the respective table cell is selected for input to view the list of available operators and then select one.

Object - the object of the match expression. Click the arrow button when the respective table cell is selected for input to select one of the following items:

- **Group** - the match rule is completed by checking which role parameter attribute was selected for the match and setting the respective group attribute in the **Attribute/Value** field.
- **Const** - a constant must be entered in the **Attribute/Value** field to complete the match rule.

Value - the attribute or value to be matched with the role parameter attribute given in **User attribute**. Enter a group attribute or a constant value here, depending on what you selected in **Object**.

Proposals (optional) - add a proposal list to define the set of available values. This proposal list overwrites the role parameter list definition. Use this feature if you want to use the same role parameter definition for several roles but with a different list of values that is presented during role assignment.

Some notes apply to this feature:

- Values supplied by the proposal list are used as delivered from the list, without type or consistency checking. You are responsible for using compatible proposal lists for your role parameters.
- The values provided by the proposal list are not subject to access policies.
- For hierarchical role parameters, if multiple roles use the same parameter, you are responsible for configuring the proposal lists for the parameters. A role parameter edit pop-up with the same parameters but different values is NOT provided.
- Role parameters can be restricted to being single valued. A single-value role parameter

value cannot exist in multiple role assignments whose periods of validity overlap.

Related Topics

- Permissions
- Assignment Editor
- Role Parameter
- General Properties
- Approval
- Certification
- Details
- Junior Roles
- Assigned Permissions
- Senior Roles
- Users

A.4.1.6. Role - Assigned Junior Roles

Use this tab to view and manage the junior roles assigned to the role. The tab displays the list of junior roles currently assigned to the role.

For each junior role, the following properties are shown:

Role - the name of the junior role.

Description - the description for the role.

To add or remove roles, use the assignment editor, which is visible when you click **Edit**. DirX Identity Provisioning checks automatically for role cycles when assigning junior roles (roles that lead to a role cycle are marked with a role icon  in the **Available Roles** list).

to examine the details of a particular junior role, click  to the right of its row.

Related Topics

- Assignment Editor
- General Properties
- Approval
- Certification
- Details
- Role Parameters
- Assigned Permissions
- Senior Roles
- Users

A.4.1.7. Role - Assigned Permissions

Use this tab to view and manage the permissions assigned to the role. The tab shows the list of permissions currently assigned to this role.

For each permission, the following properties are shown:

Permission - the name of the permission.

Description - the description for the permission.

Source - Possible values are:

- **<empty>** - the permission was assigned directly to this role with the assignment editor.
- **<role name>** - the permissions are inherited from the source role.

To add or remove permissions, use the assignment editor, which is visible when you click **Edit**. To examine the details of a particular permission, click ... to the right of its row.

Related Topics

- Permissions
- Assignment Editor
- General Properties
- Approval
- Certification
- Details
- Role Parameters
- Junior Roles
- Senior Roles
- Users

A.4.1.8. Role - Senior Roles

Use this tab to display the list of senior roles that are using this role as a junior role.

For each senior role the following properties are shown:

Role - the name of the senior role.

Description - the description for the role.

The senior roles are the backward references to all roles which have this role assigned as a junior role. You cannot use this tab to make a direct assignment. To display the details of a particular senior role, click ... to the right of its row.

Related Topics

- General Properties
- Approval
- Certification
- Details
- Role Parameters
- Junior Roles
- Assigned Permissions
- Users

A.4.1.9. Role - Users

Use this tab to display the list of users who are assigned this role.

For each user, the following properties are shown:

User - the name of the user.

Description - the description of the user.

To examine the details of a particular user, click [...] to the right of its row.

Related Topics

- Assignment Editor
- General Properties
- Approval
- Certification
- Details
- Role Parameters
- Assigned Permissions
- Senior Roles
- Role - Provisioning Rules

A.4.1.10. Role - Provisioning Rules

Use this tab to display the list of provisioning rules that reference this role as a privilege.

For each provisioning rule, the following properties are shown:

Name - the name of the provisioning rule.

Description - the description of the provisioning rule.

The provisioning rules listed here are backward references to all rules which have this role defined as a privilege via a privilege link. You cannot use this tab to make a direct

assignment. To examine the details of a particular provisioning rule, click ... to the right of its row.

Related Topics

- Assignment Editor
- General Properties
- Approval
- Certification
- Details
- Role Parameters
- Assigned Permissions
- Senior Roles
- Users

A.4.2. Permissions

This tab shows the general properties for a permission folder. The permission folder contains either permission entries or other permission folders for improving the ordering and grouping of permissions.

The property items shown here include:

Name - the displayed name of the permission folder.

Description - the description for the permission folder.

Related Topics

- Permission

A.4.2.1. Permission - General Properties

The permission object represents a right of accessing any resource or service of a particular system environment. As with the other objects, the permission object is managed by the DirX Identity Provisioning system.

Use this tab to display all general properties of a permission object. General properties include:

Identification Properties

Name (mandatory) - the name of the permission as it is used for display and identification purposes throughout the DirX Identity Provisioning system. Once set during adding a new permission entry, it can no longer be changed by simply editing this field after the initial save operation. Use the **Rename** command of the user entry's context menu instead.

Description (optional) - a description for the permission. This field is often useful when

permission entries are listed in tables with just their names but also with a description column to better identify them.

User assignment possible - whether (checked) or not (unchecked) the permission can be assigned to a user entry.

Owner - the owners of this privilege. These entities are usually the persons that must approve if this privilege is assigned to a user.

Approval Properties - for more information about this feature, see the section "Managing Request Workflows".

Requires approval - whether (checked) or not (unchecked) assigning this permission to a user requires approval via request workflows.

Potential SoD conflict (read-only) - the permission is part of one or more SoD policies, which means that a conflicting privilege exists.

Approval workflow - the request workflow to be used for approval of this privilege. If you do not enter a value into this field, the request workflow engine searches for a suitable workflow template (it uses the first workflow template that matches based on the WhenApplicable section of the workflow definition object).

Requires reapproval - whether (checked) or not (unchecked) all assignments of this role require regular re-approval: a re-approval workflow is started at defined intervals.

Reapproval date - the next date for re-approval.

Reapproval period - the frequency of the re-approval workflows. This value is only used when **Reapproval date** is not specified.

Note: If neither the **Reapproval date** nor the **Reapproval period** is set, the default values from the domain object are used. A **Reapproval period** of less than 1 day it is treated as not set and the re-approval period from the domain is used. (You cannot set a **Reapproval period** of **0** days).

Task Properties

To do - a list of manual working items for the administrator. DirX Identity Provisioning stores the results of consistency checks it cannot repair automatically. The items are overwritten upon the next consistency check.

Error - a list of error messages that show the detailed reason for inconsistent states or **To do** entries. They are cleared automatically when a privilege resolution succeeds.

To be analyzed - a flag for the consistency workflow. When checked, the permission has been changed and a privilege resolution must be performed. This flag is read-only for the administrator and is set and reset automatically by DirX Identity Provisioning workflows.

Related Topics

- Approval

- Certification
- Match Rules
- Assigned Groups
- Roles

A.4.2.2. Permission - Approval

Use this tab to view and manage the properties of the permission object that define the approval process to be used when the permission is assigned to another object (for more information about this feature, see the section "Managing Request Workflows"). These properties include:

User Assignment - fields that apply to the assignment of a permission to a user object. These fields include:

Requires approval - whether (checked) or not (unchecked) assigning this permission to a user requires approval via request workflows.

Potential SoD conflict (read-only) - whether (checked) or not (unchecked) the permission is part of one or more SoD policies, which means that a conflicting privilege exists.

Approval Workflows for

Assignment - the request workflow to be used for approval if this privilege is assigned to a user. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Modification - the request workflow to be used for approval if this privilege assignment is changed. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Removal - the request workflow to be used for approval if this privilege assignment is removed. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

SoD - the request workflow to be used for approval if an SoD violation was detected during assignment of this privilege. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the

section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Permission Assignment - fields that apply to the assignment of a permission to a role object. These fields include:

Requires approval - whether (checked) or not (unchecked) assigning this permission to a role requires approval via request workflows.

Approval Workflows for

Assignment (Default) - the request workflow to be used for approval if this privilege is assigned to a role. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Removal - the request workflow to be used for approval if this permission is removed from a role. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Related Topics

- General Properties
- Certification
- Match Rules
- Assigned Groups
- Roles

A.4.2.3. Permission - Re-approval

Use this tab to view and manage the properties of the permission object that define re-approval scenarios. These properties include:

Requires re-approval - whether (checked) or not (unchecked) all assignments of this permission should be regularly re-approved. The DirX Identity re-approval process starts a re-approval workflow at the intervals specified in the other fields in this area.

Re-approval date - the next date for re-approval. Alternatively, you can define a re-approval

period.

Re-approval period - the frequency with which the re-approval workflows should be started. This value is only used when **Re-approval date** is not set.

Note: If neither **Re-approval date** nor **Re-approval period** is set, the default values from the domain object are used.

Workflow - the request workflow to be used for re-approval. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Related Topics

- General Properties
- Approval
- Match Rules
- Assigned Groups
- Roles

A.4.2.4. Permission - Match Rules

Use this tab to view and manage the match rules defined for the permission.

For each match rule, the following properties are displayed:

And/Or - the combination operator, which can be **AND** or **OR**. Note that the first item always displays **THIS**. Changing the operator for one item changes it for the other items, too, except for the first item.

User attribute - the user attribute to be used for checking on a match. This value is one of the permission parameter attributes in the user entry's general property page. Click the arrow button when the respective table cell is selected for input to display the list of available attributes and then select one.

Operator - the operator expression to be used for matching. Click the arrow button when the respective table cell is selected for input to display the list of available operators and then select one.

Object - the object of the match expression. Click the arrow button when the respective table cell is selected for input and then select one of the following items:

- **Group** - the match rule is completed by checking which user attribute was selected for the match and setting the respective group attribute into the **Attribute/Value** field.
- **Const** - a constant must be entered into the **Attribute/Value** field to complete the

match rule.

Attribute/Value - the attribute or value to be matched with the user attribute value entered in **User attribute**. Enter a group attribute or a constant value depending on what you selected for **Object**.

Related Topics

- General Properties
- Approval
- Certification
- Assigned Groups
- Roles

A.4.2.5. Permission - Assigned Groups

Use this tab to view and manage the groups currently assigned to this permission.

For each group, the following properties are shown:

Group - the name of the group.

Target System - the target system in which this group is defined.

Description - the description for the group.

To add or remove groups, use the assignment editor, which is visible when you click **Edit**. To examine the details of a particular group, click ... to the right of its row.

Related Topics

- Assignment Editor
- General Properties
- Approval
- Certification
- Match Rules
- Roles

A.4.2.6. Permission - Roles

Use this tab to view the list of roles that are currently assigned to the permission.

For each role, the following properties are shown:

Role - the role's name as it is displayed throughout the DirX Identity Provisioning system.

Description - the description for the role.

You cannot use this tab to make direct assignment of roles to the permission. To examine the details of a particular role, click ... to the right of its row.

Related Topics

- Role - General Properties
- General Properties
- Approval
- Certification
- Match Rules
- Assigned Groups

A.4.2.7. Permission - Users

Use this tab to view the list of users that currently have this permission assigned.

For each user, the following properties are shown:

User - the user's name as it is displayed throughout the DirX Identity Provisioning system.

Description - the description for the user.

You cannot use this tab to make direct assignment of this permission to a user. To examine the details of a particular user, click ... to the right of its row.

Related Topics

- Role - General Properties
- General Properties
- Approval
- Certification
- Match Rules
- Assigned Groups
- Users
- Permission - Provisioning Rules

A.4.2.8. Permission - Provisioning Rules

Use this tab to display the list of provisioning rules that reference this permission as a privilege.

For each provisioning rule, the following properties are shown:

Name - the name of the provisioning rule.

Description - the description of the provisioning rule.

The provisioning rules listed here are backward references to all rules which have this permission defined as a privilege via a privilege link. You cannot use this tab to make a direct assignment. To examine the details of a particular provisioning rule, click ... to the right of its row.

Related Topics

- Role - General Properties
- General Properties
- Approval
- Certification
- Match Rules
- Assigned Groups
- Users

A.4.3. Groups

Use this tab to view the general properties for a group folder. The group folder first contains a subfolder for each target system which in turn contains either group entries or other subfolders for improving the ordering of groups.

The property items shown here include:

Name - the displayed name of the group folder.

Description - the description for the group folder.

You cannot add or delete groups from this tab. Use the **Target Systems** view instead.

For a description of the Attributes tabs, see the Accounts and Groups - Attributes section.

Related Topics

- Accounts and Groups - General
- Accounts and Groups - Attributes
- Accounts
- * Target System
- Multi-Value Editor
- General Properties
- Target System Specific
- Permissions
- Members
- Remote Members
- Member of Group

- Obligations

A.4.3.1. Group - General Properties

The group object represents a user group in the target system. Groups are used to grant access to system resources by giving the group the right to access the resource and adding the user who should also have this access.

Use this tab to view and manage the general properties of a group. The following standard properties are always shown:

Identification Properties

Name (mandatory) - the name of the group as it is displayed throughout the DirX Identity Provisioning system. Once set during the addition of a new group entry, it cannot be changed after the initial save operation.

Description (optional) - a description for the group. This is often useful when group entries are listed in tables with just their names but with a description column to better identify them.

User assignment possible - whether (checked) or not (unchecked) the group can be assigned to a user entry.

Group for Priv. Accounts - whether the group handles privileged accounts (checked) or personal accounts (unchecked). In this case, the **dxrUsedBy** link is set to the user entry. Note: a group can only handle privileged or personal accounts.

Owners - the owners of this privilege. These entities are usually the persons that must approve if this privilege is assigned to a user.

Related Topics

- Multi-Value Editor
- Approval
- ertification
- Target System Specific
- Operational
- Permission Parameters
- Permissions
- Privileged Members
- Members
- Remote Members
  M* ember of Group
- Obligations

A.4.3.2. Group - Approval

Use this tab to view and manage the properties of a group that define the approval process to be used when the group is assigned to another object (for more information about this feature, see the section "Managing Request Workflows"). These properties include:

User Assignment - properties that apply to the assignment of a group to a user object. These properties include:

Requires approval - whether (checked) or not (unchecked) assigning the group to a user requires approval via request workflows.

Potential SoD conflict (read-only) - whether (checked) or not (unchecked) the permission is part of one or more SoD policies, which means that a conflicting privilege exists.

Approval Workflows for

Assignment - the request workflow to be used for approval if this privilege is assigned to a user. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Modification - the request workflow to be used for approval if this privilege assignment is changed. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Removal - the request workflow to be used for approval if this privilege assignment is removed. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

SoD - the request workflow to be used for approval if, during assignment of this privilege, an SoD violation is detected. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Permission Assignment - properties that apply to the assignment of a group to a

permission object. These properties include:

Requires approval - whether (checked) or not (unchecked) assigning the group to a permission requires approval via request workflows.

Approval Workflows for

Assignment (Default) - the request workflow to be used for approval if this privilege is assigned to a permission. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Removal - the request workflow to be used for approval if this group is removed from a role. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Related Topics

- General Properties
- Certification
- Target System Specific
- Operational
- Permission Parameters
- Permissions
- Privileged Members
- Members
- Remote Members
- Member of Group
- Obligations

A.4.3.3. Group - Re-approval

Use this tab to view and manage the properties of the group that define re-approval scenarios. These properties include:

Requires re-approval - whether (checked) or not (unchecked) all assignments of this group must be regularly re-approved. The DirX Identity re-approval process starts a re-approval workflow at the intervals defined in the other fields in this area.

Re-approval date - the next date for re-approval. Alternatively, you can define a re-approval period.

Re-approval period - the frequency with which re-approval workflows should be started. This value is only used when the **Re-approval date** is not set.

Note: If neither **Re-approval date** nor **Re-approval period** are set, the default values from the domain object are used.

Workflow - the request workflow to be used for re-approval. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Related Topics

- General Properties
- Target System Specific
- Operational
- Permission Parameters
- Permissions
- Privileged Members
- Members
- Remote Members
- Member of Group
- Obligations

A.4.3.4. Group - *Target System Specific*

Use this tab to display and manage target-specific properties. These properties include:

Default target system-specific group properties

Dashboard

The Dashboard target system has no specific properties.

JDBC

The JDBC target system has the following specific properties:

JDBC Key - the value of the primary key in the JDBC target system.

LDAP

The LDAP target system has the following specific properties:

Primary Key (DN in TS) - the DN in the LDAP target system.

Mailing List

This target system has the following target system-specific properties:

EMail - the recipients' e-mail addresses.

Notes

This target system has the following specific properties:

List Name - the unique name of the group in the Notes target system.

Type - the type of the group. Valid values are:

- **Multi purpose**
- Mail only
- Access Control List only
- Deny List only
- Servers only

Administrators - the list of administrators in the Notes target system.

ODBC

This target system has the following specific properties:

ODBC Key - the value of the primary key in the ODBC target system.

Office 365

This target system has the following specific properties:

Primary Key (ID in TS) - the identifier of the role (security group or service plan) generated by Office 365.

Display Name - the display name attribute of the role (security group or service plan).

Mail Nickname - the mail nickname attribute for the security group.

Sku ID - the license identifier related to the selected service plan.

Group Type - the type of group (Security, Microsoft 365 – Public, Microsoft 365 – Private, Microsoft 365 – Hiddenmembership).
The value is stored in **dxrType** - SecurityGroup, MS365GroupPublic, MS365GroupPrivate, MS365GroupHiddenMembership.

RACF

This target system has no specific properties.

SAP NetWeaver UM

This target system has the following specific properties:

dxrName - the name of the SAP NetWeaver UM target system in Provisioning.

Primary Key (DN in TS) - the DN in the SAP NetWeaver UM target system.

SAP ECC UM

This target system has no specific properties.

Sipass

This target system has no specific properties.

UNIX-OpenICF

This target system has the following specific properties:

Group Name - the unique group name in a UNIX system.

GID Number - the unique GID in a UNIX system.

Windows 2008/2012

This target system has the following specific properties:

Type - the type of the group. A group can be a local, global or universal security group or distribution list. Possible values are:

- **local, Security**
- local, Distribution list
- global, Security
- global, Distribution list
- universal, Security
- universal, Distribution list

Internal Type - for internal use only. Usually, the groups have an empty value. DirX Identity Provisioning provides one special, pre-installed group named "dxr Mailbox Users", which can be used for granting an Exchange20xx mailbox to its members. This group is identified by its internal type **MAILBOX**. In its **On Assignment** and **On Revocation** tabs, it specifies naming rules that are applied to each member account added to or deleted from this group. These naming rules set / reset the mailbox-relevant attributes of the account.

Windows NT

This target system has the following specific properties:

Type - (optional) the type of the group. A group can be a local or global security group. Possible values are:

- **local**
- global

Related Topics

- Multi-Value Editor
- General Properties
- Approval
- Certification
- Operational
- Permission Parameters
- Permissions
- Privileged Members
- Members
- Remote Members
- Member of Group
- Obligations

A.4.3.5. Group - Operational

Use this tab to view and manage the attributes for the group that are related to standard DirX Identity operation.

Operational Attributes

State - the status of the group entry. Possible values are (for more information, see the section "Managing States" in the chapter "Managing Provisioning"):

- **ENABLED** - the group exists in the target system or shall be created.
- **DELETED** - the group shall be deleted in the target system.

End date - the date on which the group is to be deleted. When this date occurs, the group entry is deleted in DirX Identity Provisioning independent of its state in the target system.

State in target system - the status of the group in the respective target system. This value can be different from the status of the group entry as long as this information is not synchronized. Possible values are (for more information, see the section "Managing States" in the chapter "Managing Provisioning"):

- **ENABLED -** the account exists in the target system and is enabled.
- **DELETED** - the object has been deleted in the target system without DirX Identity Provisioning having requested it! This state is only set by the validation workflow along

with a message in the **dxrToDo** attribute reminding the administrator to inspect this situation.

- **NONE** - replaces the empty value when the object is created by the DirX Identity Manager or agent.

Timing Properties

Create time stamp - the date and time at which the object was created in the directory.

Modify time stamp - the date and time of the last modification of this object.

Task Properties

To do - a list of manual working items for the administrator. DirX Identity Provisioning stores the results of consistency checks that it cannot automatically repair in this field. The items here are overwritten on the next consistency check.

Error - a list of error messages that show the detailed reason for inconsistent states or **To do** entries. These messages are cleared automatically when a privilege resolution succeeds.

To be analyzed (read-only) - when checked, indicates to the consistency workflow that the group has been changed and that a privilege resolution must be performed for all affected users. This flag is read-only for the administrator and is set and reset automatically by DirX Identity Provisioning workflows.

Related Topics

Multi-Value Editor
General Properties
Approval
Certification
Target System Specific
Permission Parameters
Permissions
Privileged Members
Members
Remote Members
Member of Group
Obligations

A.4.3.6. Group - Permission Parameters

Use this tab to display the permission parameters for assigning the group. The privilege resolution process evaluates these attribute values for a user when the match rule of an assigned role or permission is applied. The user is granted the group when the attribute value of the group and the user (or user-to-role assignment) match.

For details, see the section "Managing Critical Parameters" in the "Managing Domain" chapter.

Related Topics

Multi-Value Editor
General Properties
Approval
Certification
Target System Specific
Operational
Permissions
Privileged Members
Members
Remote Members
Member of Group
Obligations

A.4.3.7. Group - Permissions

Use this tab to display the list of permissions currently assigned to this group.

For each permission, the following properties are shown:

Permission - the name of the permission.

Description - the description of the permission.

You cannot use this tab to make direct assignment of permissions. The list shows the backward references to all permissions which have this group assigned. To view the details of a permission, click  to the right of its row.

Related Topics

Multi-Value Editor
General Properties
Approval
Certification
Target System Specific
Operational
Permission Parameters
Privileged Members
Members
Remote Members
Member of Group
Obligations

A.4.3.8. Group - Privileged Members

Use this tab to display the list of privileged accounts currently assigned to the group.

Related Topics

Multi-Value Editor
General Properties
Approval

Certification
Target System Specific
Operational
Permission Parameters
Permissions
Members
Remote Members
Member of Group
Obligations

A.4.3.9. Group - Members

Use this tab to display the list of members currently assigned to this group.

For each member, the following properties are shown:

Account or group - the name of the account or group.

State of Assignment - the status of the member's assignment to the group. It can take one of the following values:

- **ENABLED**- the assignment is up to date and there are currently no problems with it.
- **IMPORTED** - the assignment was imported from the target system and was not granted by any privilege assignment. The state switches automatically to **ENABLED** if the group is granted via a privilege assignment. The administrator must handle these assignments, which are identified in the **ToDo** query folders in each target system. By selecting the assignment in the "Member of" tab of the account, the administrator can either delete this assignment or accept it and then switch it to state **IGNORE**.
- **IGNORE** - the administrator has accepted this imported assignment. The state switches automatically to **ENABLED** if the group is granted via a privilege assignment.
- **ADD** - the assignment is a new one and must still be synchronized with the target system.
- **DELETED** - the assignment is marked for deletion.

You cannot use this tab to make direct assignment of members. The access rights of groups can only be granted by the assigned privileges. To view the details of a member, click ... to the right of its row.

Related Topics

Multi-Value Editor
General Properties
Approval
Certification
Target System Specific
Operational
Permission Parameters
Permissions
Privileged Members

Remote Members
Member of Group
Obligations

A.4.3.10. Group - Remote Members

Some target systems are structured into different domains. In this case, groups from one domain can contain accounts or groups from a foreign domain. Use this tab to display the list of members from foreign domains currently assigned to this group.

For each member, the following properties are shown:

Account or group - the name of the account or group.

You cannot use this tab to make direct assignment of members. To examine the details of a member, click ... to the right of its row.

Related Topics

Multi-Value Editor
General Properties
Approval
Certification
Target System Specific
Operational
Permission Parameters
Permissions
Privileged Members
Members
Member of Group
Obligations

A.4.3.11. Group - Member of Group

Use this tab to display the groups of which this group is a member. The tab allows you to view the next higher level of groups in a hierarchical target system.

For each member, the following properties are shown:

Group - the name of the group to which this group belongs.

Description - the description of the group object.

State of Assignment - the status of the member's assignment to the group. Possible values are:

- **ENABLED**- the assignment is up to date and there are currently no problems.
- **IMPORTED** - the assignment was imported from the target system and not granted by any privilege assignment. The state switches automatically to **ENABLED**, if the group is granted via a privilege assignment. The administrator must resolve these assignments, which are shown in the **ToDo** query folder of each target system. By selecting the

assignment in the "Member of" tab of the account, the administrator can either delete this assignment or accept it and then switch its state to **IGNORE**.

- **IGNORE** - the administrator has accepted this imported assignment. The state switches automatically to **ENABLED** if the group is granted via a privilege assignment.
- **ADD** - the assignment is a new one and must still be synchronized with the target system.
- **DELETED** - the assignment is marked for deletion.

You cannot use this tab to make direct assignments of members. The access rights of groups can only be granted by the assigned privileges. To examine the details of a particular member, click ... to the right of its row.

Related Topics

Multi-Value Editor
General Properties
Approval
Certification
Target System Specific
Operational
Permission Parameters
Permissions
Privileged Members
Members
Remote Members
Obligations
Provisioning Rules

A.4.3.12. Group - Obligations

Obligations are a method for setting or resetting account attributes when an account becomes a member of a group or is removed from the group. Obligations can be local to a group or common to several groups.

The local obligations page contains these properties:

Obligation link - the link to a common obligation object of this target system (see the Target Systems View and in this view, the folder Configuration -> Obligations).

For the rest of the properties on this page, see the description of the common obligation object.

Related Topics

Multi-Value Editor
General Properties
Approval
Certification
Target System Specific
Operational

Permission Parameters
Permissions
Privileged Members
Members
Remote Members
Member of Group
Provisioning Rules

A.4.3.13. Group - Provisioning Rules

Use this tab to display the list of provisioning rules that reference this group as a privilege.

For each provisioning rule, the following properties are shown:

Name - the name of the provisioning rule.

Description - the description of the provisioning rule.

The provisioning rules listed here are backward references to all rules which have this group defined as a privilege via a privilege link. You cannot use this tab to make a direct assignment. To examine the details of a particular provisioning rule, click ... to the right of its row.

Related Topics

Multi-Value Editor
General Properties
Approval
Certification
Target System Specific
Operational
Permission Parameters
Permissions
Privileged Members
Members
Remote Members
Member of Group
Obligations

A.5. Policies View

The **Policies** view shows policies and delegations with their organizing folders.

For each selected item in the tree, the respective property pages are shown on the right-hand side.Click **Edit** to modify property values.

You can add, modify, copy, move and delete structuring folders, access policies, rules and operations.

You can only modify delegation and access right objects.Creation is only possible via the Web Center.

Related Topics

Rules
Operations

A.5.1. Policies

This tab shows the general properties for the policies folder.

The property items shown here are the following:

Name - the displayed name of the policies folder.

Description - the description for the policies folder.

Related Topics

Rules Folder
Operations Folder

A.5.1.1. Access Policies

Use this tab to display the properties for an access policies folder. It can contain other access policy folders or access policies objects.

The property items shown here are the following:

Name - the displayed name of the access policy folder.

Description - the description for the access policy folder.

Related Topics

- Access Policy

A.5.1.1.1. Access Policy - General

In DirX Identity Provisioning, an access policy represents an object that defines the access rights of users. Use this tab to view and manage the general properties of an access policy. These properties include:

Name - the name of the access policy object.

Description - a description for the access policy object.

Owner - the person that created this access policy.

Is active - the access policy is only used if this flag is set.

Is delegatable - whether (checked) or not (unchecked) the access rights resulting from this policy can be delegated to other users. This field allows you to restrict delegation to access policies that make sense to be delegated and that do not comprise huge numbers of objects. For example, delegating the access policy that all employees of a company can

read all other employees does not make sense (comprises a huge number of target objects) while delegating the modify right of the project manager for his team members makes sense (comprises only a few target objects).

Operation - the operation of the access policy. The following operations exist:

Object handling:
create - allows creating this type of object.
read - allows reading this type of object (can be used together with modify).
modify - allows modifying this type of object (can be used together with read).
delete - allows deleting this type of object.*
Privilege assignment handling:*
viewAssignments - allows viewing privilege assignments or accounts.
grant - allows assigning a privilege (role, permission, group) or using a specific set of role parameters. A request workflow is started if the privilege is marked accordingly.
assignDirect - allows assigning a privilege, but no request workflow is started even if the privilege is marked for approval. This operation works only in connection with a grant policy for the privilege.
approve - allows approving privilege assignments.*
Request workflow handling:*
execute - allows executing request workflows.
suspend - allows suspending a running request workflow.
resume - allows resuming a suspended request workflow.
stop - allows stopping a running request workflow.*
Request workflow participant handling:*
changeParticipant - allows changing a participant in a task.
showTasksOf - defines the users whose task lists you can view.
delegate - defines the users to whom you can delegate access rights.*
Password handling:*
setPassword - allows setting the password for this object.
readPassword - allows reading the password for this object.

Object - the object type the access policy protects. The operations allowed are enclosed in brackets.

Account - handles objects of type account (create, read, modify, delete, viewAssignment, setPassword, readPassword).*
Context* - handles business objects of type context (create, read, modify, delete).*
CostUnit* - handles business objects of type cost unit (create, read, modify, delete).*
Country* - handles business objects of type country (create, read, modify, delete).*
Group* - handles objects of type group (create, read, modify, delete, viewAssignment, grant, approve).*
Location* - handles business objects of type location (create, read, modify, delete).*
Menu* - handles objects of type Web Center menu (execute).*
Organization* - handles business objects of type organization (create, read, modify, delete).*
OrganizationalUnit* - handles business objects of type organizational unit (create, read, modify, delete).*
Password policy* - handles objects of type password policy (create, read, modify, delete).*
Permission* - handles objects of type permission (create, read, modify, delete, viewAssignment, grant, approve).*

Project* - handles business objects of type project (create, read, modify, delete).*
Provisioning rule* - handles objects of type provisioning rule (create, read, modify, delete).*
Report* - handles objects of type report (execute).*
Role* - handles objects of type role (create, read, modify, delete, viewAssignment, grant, approve).*
RoleParam* - handles role parameter values (grant).*
User* - handles objects of type user (create, read, modify, approve, setPassword).*
Target System* - handles objects of type target system (create, read, modify, delete).*
Workflow definition* - handles objects of type request workflow definition (create, read, modify, delete, execute).*
Workflow instance* - handles objects of type request workflow instance (read, modify, delete, suspend, resume, stop).

Role Parameter - visible when **RoleParam** is selected in **Object** and points to the role parameter definition object.

DirX Identity provides a set of object types that are protected via access policies. See the chapter "Customizing Access Policies" in the *DirX Identity Customization Guide* to enable more pre-defined object types or your custom object types for access control.

Related Topics

Subjects
Resources
Rules
Attributes Modify
Attributes Read

A.5.1.1.2. Access Policy - Subjects

Use this tab of an access policy to define the subjects (users) for whom the access policy applies.

The available attributes are:

Persons - a static list of persons (a list of links to these persons).

Groups of Persons - a static list of groups (a list of links to these groups that each contains a list of links to persons).

Persons Filter - a filter definition (Search base and Search filter) that allows defining a dynamic set of persons.

Exclude Persons - a static list of persons (a list of links to these persons) that is removed from the union of the results from Persons, Group of Persons and Person Filter.

All these sets of persons are combined into one set of persons as the result that is used by the access policy.

Related Topics

General

Resources
Rules
Attributes Modify
Attributes Read

A.5.1.1.3. Access Policy - Resources

Use this tab of an access policy to define the resources with which the access policy works; that is, the objects on which the subject users can operate.

The available attributes are:

Resources - a static list of resources (a list of links to these resources).

Resource Groups - a static list of groups (a list of links to these groups that each contains a list of links to resources).

Resource Filter - a filter definition (Search base and Search filter) that allows defining a dynamic set of resources.

Exclude Resources - a static list of resources (a list of links to these resources) that is removed from the union of the results from Resources, Group of Resources and Resource Filter.
Note: this field is not operable for access policies of object type RoleParam.

All these sets of resources are combined into one set of resources as the result that is used by the access policy.

Resources can be privileges (roles, permissions, groups) or users.

Related Topics

General
Subjects
Rules
Attributes Modify
Attributes Read

A.5.1.1.4. Access Policy - Attributes Read

Use this tab of an access policy to define the attributes which should be readable via the Business User Interface. Modifiable attributes are readable by default. It is not necessary to configure them here.

If no attributes are selected, all attributes are readable.

Related Topics

General
Subjects
Rules
Attributes Modify

A.5.1.1.5. Access Policy - Attributes Modify

Use this tab of an access policy to define the attributes which should be modifiable via the Business User Interface. Modifiable attributes are readable by default. It is not necessary to configure them in the Attributes Read tab.

If no attributes are selected, all attributes can be changed.

Related Topics

General
Subjects
Rules
Attributes Read

A.5.1.1.6. Access Policy - Rules

Use this tab of an access policy to define the rules the access policy uses. Using rules allows you to reduce the resource objects based on their relationship to the subject; for example, to only those users who are managed by the subject.

The available attributes are:

Effect - the effect of the rule. Currently, only Permit is possible as a value. This value cannot be changed (read-only).

Rule - a definition that evaluates attributes of the subject with attributes of the resource. All objects that satisfy the rule are retrieved as the result. Note that the rule field can be empty (in this case, the entire set of resources is used).

The subject side can contain reference chains with multi-value attributes; for example, subject.dxrContextLink.dxrPrivilegeLink (see also the example below).

Access to specific attribute values is possible with - for example - dxrRPvalues(myParamName). See also the example below.

In the subject part, you can reference a defined role parameter of a specified role assignment.

The following expressions are supported:

- Define the role for the assignment with a DN:

$(subject.roleAssignment[dxrAssignTo=cn=Project Member,cn=Project Specific,cn=Corporate Roles,cn=RoleCatalogue,cn=My-Company].roleParameter_Project)

where:

roleAssignment is the keyword for role assignments.

[..] is an expression that defines the which assignments to be used.

dxrAssignTo= takes the assignment that references the given DN (of the role).

roleParameter_ is the keyword for role parameters.

roleParameter_Project is the value after the _ that specifies the role parameter name.

Use the Roleparameter values of the role parameter Project for role assignments that reference the role "Project Member" of the subject(=user).

- Define the role for the assignment by an unique identifying attribute

$(subject.roleAssignment[dxrAssignTo@roleName=Project Member].roleParameter_Project)

where:

dxrAssignTo@roleName= takes the assignment that references a role with the given attribute value "Project Member" for the attribute roleName

Here is a sample rule as it is presented in the Rule Editor:

($(subject.roleAssignment[dxrAssignTo@cn=Project Member].roleParameter_Project)=$\28resource.dxrproject\29)

You can define any number of rules (use the + and - buttons). The result of each rule is OR-combined to one result.

Limitations:

You must not use placeholder expressions **$(subject...)** or **$(resource...)** within AND- or OR-filters. You may specify OR-filters by adding another rule using the **+** button.

Examples:

1) Simple rule:

$(subject.dn)="$(resource.manager)"

This rule compares the manager attribute of the resource (a user) with the dn of the subject. It retrieves the manager(s) of the user.

2) Example for a reference chain:

($(subject.dxrContextLink.dxrPrivilegeLink)=$(resource.dn))

In this example, the subject is linked with a context object via the dxrContextLink. The attribute dxrPrivilegeLink is compared with the dn of the resource. The references are resolved recursively: if the dxrContextLink contains multiple values, then all references are resolved.

3) Access to specific attributes

($(subject.dxrPrivilegesGrantedLink)=$(resource.dxrRPvalues(name))

Allows a comparison between the approved assigned privileges at the user object (the

subject) with the corresponding specific value at the group.

Related Topics

General
Subjects
Resources
Attributes Modify
Attributes Read

A.5.1.2. Attribute Policies

Attribute policies define when approval of object attribute changes is needed.

You can structure attribute policies in folders and define attribute policy objects.

Related Topics

Attribute Policies Folder
Attribute Policy

A.5.1.2.1. Attribute Policies Folder

This tab shows the properties for an attribute policies folder. It can contain other attribute policy folders or attribute policy objects.

The property items shown here include:

Name - the displayed name of the folder.

Description - the description for the folder.

Related Topics

- Attribute Policy - General
- Attribute Policy - Configuration

A.5.1.2.2. Attribute Policy - General

In DirX Identity Provisioning, an attribute policy represents an object that defines the attributes of an object whose change automatically triggers an approval workflow. To enable this feature, set the Attribute Modification Approval flag of the domain object.

Use this tab to view and manage the general properties of an access policy. These properties include:

Name - the name of the object.

Description - a description for the object.

Is active - whether (checked) or not (unchecked) the attribute policy is in use (activated).

Operation - the operations for which this policy is valid.

Approval Workflow - the associated approval workflow to be started. If this link is not set, the standard evaluation mechanism of the workflow engine applies.

Related Topics

- Attribute Policies Folder
- Attribute Policy - Configuration

Section "Understanding Modification Workflows" in chapter "Using Request Workflows" in the *DirX Identity Application Development Guide*.

A.5.1.2.3. Attribute Policy - Configuration

Use this tab of an attribute policy to select the attributes for which an approval workflow should be started. The available controls are:

Object type - the object description name for which this policy is valid.

Object class - The object class to identify entries of this type. The meta controller uses this field to locate the appropriate attribute policy.

Audit object type - The object type to be inserted into audit messages.

Available - the available properties of this object. This list comes from the corresponding object descriptions (if several object descriptions for this objects exist, the attribute list is merged).

Selected - the selected attributes for which the policy is valid. Use the arrow buttons to move attributes from the upper pane to the lower or vice-versa. If one of these attributes is changed, an approval workflow must be started. Note that this attribute list is not evaluated when deciding whether to send a change event.

Related Topics

- Attribute Policies Folder
- Attribute Policy - General

Section "Understanding Modification Workflows" in chapter "Using Request Workflows" in the *DirX Identity Application Development Guide*

A.5.1.3. Delete Policies

Delete policies define whether a request workflow should be started if an object is to be deleted.

You usually only need to define one delete policy for your domain, so using folders to structure this area is not necessary.

Related Topics

- Delete Policy - General
- Delete Policy - Configuration
- Delete Policies

A.5.1.3.1. Delete Policies Folder

This tab displays the properties for the top-level delete policies folder. It can only contain delete policy objects.

The property items shown here include:

Name - the displayed name of the folder.

Description - the description for the folder.

Related Topics

- Delete Policy - General
- Delete Policy - Configuration
- Delete Policies

A.5.1.3.2. Delete Policy - General

A delete policy configures, for a list of object types, whether a request workflow must be started.

The list of object types is configured in the Configuration tab. The **When Applicable** tab of the deletion request workflow must be set exactly to this object type (which is the name of the object description). The operation must be set to **delete**.

We recommend configuring all object types in one policy. However, the system searches all active delete policies and collects their 'delete' flags. If an object type occurs in more than one policy with different flags (one is set to true, the other to false), the result cannot be predicted: it depends on the sequence of the search result.

Use this tab to view and manage general properties of a delete policy. These properties include:

Name - the name of the object.

Description - a description for the object.

Is active - whether (checked) or not (unchecked) the delete policy is in use (activated).

Related Topics

- Delete Policies Folder
- Delete Policy - Configuration
- Delete Policies

A.5.1.3.3. Delete Policy - Configuration

Use this tab of a delete policy to select the object types for which a deletion request workflow should be started. The available controls are:

Available - displays all available object description names. Click an entry and then use the corresponding button to move it down to the selected area.

Selected - shows the selected object descriptions. Click an entry and then use the corresponding button to remove it from this list. For all entries in the list, you can set two additional attributes:

Object Class - the object class for this object description type. This field allows the meta controller to send events.

Send - whether (checked) or not (unchecked) event creation is enabled.

Related Topics

- Delete Policies Folder
- Delete Policy - General
- Delete Policies

A.5.1.4. Event Policies

Event policies define event-based handling of object attributes.

You usually need to define only one event policy for your domain, so folders for structuring this area are not necessary.

Related Topics

- Event Policy - General
- Event Policy - Configuration
- Event Policy - General

A.5.1.4.1. Event Policies Folder

This tab displays the properties for the top-level event policies folder. It can only contain event policy objects.

The property items shown here include:

Name - the displayed name of the folder.

Description - the description for the folder.

Related Topics

- Event Policy - General
- Event Policy - Configuration

- Event Policy - General

A.5.1.4.2. Event Policy - General

An event policy configures, for a list of object types, whether an event is sent when an entry of this type is changed. Change in this context means either an attribute change or creation of the entry.

When the IdS-J Server receives this event, it starts the event-based workflow configured for this object type.

The list of object types is configured in the Configuration tab. The event workflow requires exactly this object type (which is the name of the object description) in the field **Type** of its **Is applicable for** configuration.

We recommend configuring all object types in one policy. However, the system searches all active event policies and collects their 'send' flags. If an object type occurs in more than one policy with different flags (one is set to true, the other to false), the result cannot be predicted: it depends on the sequence of the search result.

Use this tab to view and manage general properties of an event policy. These properties include:

Name - the name of the object.

Description - a description for the object.

Is active - whether (checked) or not (unchecked) the event policy is in use (activated).

Related Topics

- Event Policies Folder
- Event Policy - Configuration
- Event Policy - General

A.5.1.4.3. Event Policy - Configuration

This tab of an event policy allows you to select the object types for event generation. The available controls are:

Available - displays all available object description names. Click an entry and then use the corresponding button to move it down to the selected area.

Selected - shows the selected object descriptions. Click an entry and then use the corresponding button to remove it from this list. For all entries in the list, you can set two additional attributes:

Object Class - the object class for this object description type. This value allows the meta controller to send events.

Object Class2 - a second object class for this object description type. This value allows the

meta controller to distinguish between users, personas and functional users. Since personas and functional users have the object class dxrUser (as the normal users), the meta controller matches the definition with the best match (a two object class match is preferred over a single object class match).

Type - another match criterion for the meta controller. Evaluates the object's dxrType attribute. Matching is performed in the order
1) two object classes match
2) object class and type match
3) object class matches

Send - whether (checked) or not (unchecked) event creation is enabled.

Related Topics

- Event Policies Folder
- Event Policy - General
- Event Policy - General

A.5.1.5. Risk Policies

Risk policies define the risk factors that are to be used for risk calculation and their weights and specify upper and lower risk limits for risk classification. They also store the compound score deviation and the date of the last run of the Risk Calculation workflow. Only one risk policy can be active for a domain at a time.

Related Topics

- Risk Policy - General
- Managing Risk

A.5.1.5.1. Risk Policy - General

Use this tab to view and manage the general properties of a risk policy object.

General properties of a risk policy include:

Name - the name of the risk policy.

Is active - whether (checked) or not (unchecked) the risk policy is in use (activated). Only one risk policy can be active for a domain at a time.

Last Risk Calculation - the date of the Risk Calculation workflow's last run. The Risk Calculation workflow populates this field.

Compound Score Deviation – the standard deviation of the compound score over all users. The Risk Calculation workflow populates this field.

Upper Risk Limit - the upper risk limit. A user compound score that exceeds this limit classifies the risk as high.

Lower Risk Limit - the lower risk limit. A user compound score that falls below this limit classifies the risk as low.

For every risk factor that you want to use, define the risk factor and its weight. Empty risk factors are ignored. Do not define a risk factor twice. Risk factor properties include:

Risk Factor - the risk factor to be selected. Leave this field blank to ignore the selected risk factor. Don't select a risk factor twice.

Risk Weight - the weight of the selected risk factor.

Standard Deviation - the standard deviation of the risk scores (= raw value) of this factor over all users.

mean - the arithmetic mean of the risk scores (= raw value) of this factor over all users.

count - the sum of the occurrences of the appropriate risk factor: count SoD violations, count memberships, and so on.

Related Topics

Managing Risk

A.5.1.6. Rules

Use this tab to display the properties for a rule folder. The rule folder contains either rule entries (consistency, provisioning or validation rules) or other rule folders for improving the ordering and grouping of rules.

The property items shown here include:

Name - the displayed name of the rules folder.

Description - the description for the rules folder.

Related Topics

- Rules
- Operations
- Consistency Rule - General Properties
- Provisioning Rule - General Properties Validation Rule - General Properties

A.5.1.6.1. Consistency Rule - General Properties

Use this tab to view and manage the general properties of a consistency rule object. These properties include:

Name - the name of the consistency rule as it is used for display and identification purposes throughout the DirX Identity Provisioning system.

Description - a description for the consistency rule.

Type - a fixed type value: ConsistencyRule (read-only).

Operation - a link to the operation that is performed on the subjects when the rule is executed. After setting this link, the name, type and description fields of the parameter list below are filled automatically.
Note: Be sure that the operation is set to active. Otherwise execution of the rule will fail.

Parameter Values - each line represents one of the parameters the operation requires.

- **Name** - name of the parameter (read only)
- **Value** - value of the parameter
- **Type** - type of the parameter (read only)
- **Description** - description of the parameter (read only)

Use the name, type and description information to enter the correct parameter values.

Is Active - whether (checked) or not (unchecked) the rule can be executed.
Note that this flag is automatically reset if the rule is deleted.

System Default - whether (checked) or not (unchecked) the object is a default object that cannot be changed (the **Edit** action is not available). Copy the object and then modify it according to your needs.

Related Topics

- Consistency Rule - Filter
- Rules
- Operations
- Provisioning Rule - General Properties Validation Rule - Genera Properties

A.5.1.6.2. Consistency Rule - Filter

Use this tab to view and manage the filter properties that define the subjects on which the consistency rule is to operate. These properties include:

Search Base - the base at which to start the search.

Search Filter - the LDAP filter condition.

Search Scope - the scope to be searched. Possible selections are:

- **Subtree** - the entire subtree that belongs to the search base.
- **One Level** - only the objects at the level directly under the search base.
- **Base Object** - only the search base object itself.

You can use the **+** and **-** buttons to define multiple combined LDAP filters.

The following expression types can be used in the filter for time attributes:

- **$*_base_ or *$(*_base_)*** - represents the current time, depending on *base*. *base* can be:

  NOW or **gmtime** or **time** - current time in GMT.
  localtime - current time in local time zone.
  date - the time of this day start in GMT.
  localdate - the time of this day start in local time zone.

  Examples:

  dxrExpirationDate>=$NOW - retrieves all entries that will expire in future.

  &(dxrStartDate>=$(date))(dxrStartDate<=$(time)) - retrieves all entries that were activated today up to now.

- **$*_base operation constant_ or *$(*_base operation constant_)*** - the time plus or minus a constant. The format of *constant* is:

  *n*y*n*M*n*d*n*h*n*m*n*s*

  where *n* is the number of time units. The time units are:

  y years
  M months
  d days
  h hours
  m minutes
  s seconds.

  The order of time units is fixed, but each unit is optional. For example:

  (dxrStartDate>=$(NOW-3h)) - retrieves all entries that were created within the last three hours.

  (dxrExpirationDate<=$(gmtime+1y6M)) - retrieves all entries that expire in one and a half year.

  A number without a time unit indicates days.

Related Topics

- Consistency Rule - General Properties
- Rules
- Operations
- Provisioning Rule - General Properties Validation Rule - General Properties

A.5.1.6.3. Provisioning Rule - General Properties

Use this tab to view and manage the general properties of a provisioning rule object. These properties include:

Name - the name of the provisioning rule as it is used for display and identification purposes throughout the DirX Identity Provisioning system.

Description - a description for the provisioning rule.

Type - a fixed type value: ProvisioningRule (read-only).

Operation - the provisioning operation type. Possible selections include:

- **Grant** - grants the defined privileges (the subjects get these privileges).
- **Deny** - denies the defined privileges (the subjects do not get these privileges).

Priority - the priority of the provisioning rule. Zero is lowest priority, 100 is highest priority. Use this parameter to define precedence during rule processing.

Is Active - whether (checked) or not (unchecked) the rule can be executed.
Note that this flag is automatically reset if the rule is deleted.

System Default - whether (checked) or not (unchecked) the object is a default object that cannot be changed (the **Edit** action is not available). Copy the object and then modify it according to your needs.

Related Topics

- Consistency Rule - General Properties
- Rules
- Operations
- Provisioning Rule - Filter
- Provisioning Rule - Privileges Validation Rule - General Properties

A.5.1.6.4. Provisioning Rule - Filter

Use this tab to view and manage the filter properties that define the subjects on which the provisioning rule operates. These properties include:

Search Base - the base from which to start the search. The search base must be all or a part of the user tree.

Search Filter - the LDAP filter condition.

Search Scope - the scope to be searched. Possible selections are:

- **Subtree** - the entire subtree that belongs to the search base.
- **One Level** - only the objects at the level directly under the search base.
- **Base Object** - only the search base object itself.

You can use the **+** and **-** buttons to define multiple combined LDAP filters.

The following expression types can be used in the filter for time attributes:

- **$*_base_ or *$(*_base_)*** - represents the current time, depending on *base*. *base* can be:

  NOW or **gmtime** or **time** - current time in GMT.
  localtime - current time in local time zone.
  date - the time of this day start in GMT.
  localdate - the time of this day start in local time zone.

  Examples:

  dxrExpirationDate>=$NOW - retrieves all entries that will expire in future.

  &(dxrStartDate>=$(date))(dxrStartDate<=$(time)) - retrieves all entries that were activated today up to now.

- **$*_base operation constant_ or *$(*_base operation constant_)*** - the time plus or minus a constant. The format of *constant* is:

  *n*y*n*M*n*d*n*h*n*m*n*s*

  where *n* is the number of time units. The time units are:

  y years
  M months
  d days
  h hours
  m minutes
  s seconds.

  The order of time units is fixed, but each unit is optional. For example:

  (dxrStartDate>=$(NOW-3h)) - retrieves all entries that were created within the last three hours.

  (dxrExpirationDate<=$(gmtime+1y6M)) - retrieves all entries that expire in one and a half years.

  A number without a time unit indicates days.

Related Topics

- Consistency Rule - General Properties
- Rules
- Operations
- Provisioning Rule - General Properties
- Provisioning Rule - Privileges Validation Rule - General Properties

A.5.1.6.5. Provisioning Rule - Privileges

Use this tab to set a list of privileges that are used by the provisioning rule for the subjects

on which the provisioning rule operates. These properties include:

Privileges to Assign - the list of privileges the rule is to provision. You can define any number of privileges (roles, permissions, groups) in any combination.

Privilege Filter - this option is only available when **Operation** is set to **Deny**. It allows you to select all privileges based on the defined filter condition.

Search Base - the base from which to start the search.

Search Filter - the LDAP filter condition.

Search Scope - the scope to be searched. Possible selections are:

- **Subtree** - the entire subtree belonging to the search base.
- **One Level** - only the objects at the level directly under the search base.
- **Base Object** - only the search base object itself.

You can use the **+** and **-** buttons to define multiple combined LDAP filters.

The following expression types can be used in the filter for time attributes:

- **$*_base_ or *$(*_base_)*** - represents the current time, depending on *base*. *base* can be:

  NOW or **gmtime** or **time** - current time in GMT.
  localtime - current time in local time zone.
  date - the time of this day start in GMT.
  localdate - the time of this day start in local time zone.

  Examples:

  dxrExpirationDate>=$NOW - retrieves all entries that will expire in future.

  &(dxrStartDate>=$(date))(dxrStartDate<=$(time)) - retrieves all entries that were activated today up to now.
- **$*_base operation constant_ or *$(*_base operation constant_)*** - the time plus or minus a constant. The format of *constant* is:

  *n*y*n*M*n*d*n*h*n*m*n*s*

  where *n* is the number of time units. The time units are:

  y years
  M months
  d days
  h hours
  m minutes
  s seconds.

  The order of time units is fixed, but each unit is optional. For example:

(dxrStartDate>=$(NOW-3h)) - retrieves all entries that were created within the last three hours.

(dxrExpirationDate<=$(gmtime+1y6M)) - retrieves all entries that expire in one and a half year.

A number without a time unit indicates days.

Related Topics

- Consistency Rule - General Properties
- Rules
- Operations
- Provisioning Rule - Filter
- Provisioning Rule - General Properties Validation Rule - General Properties

A.5.1.6.6. Validation Rule - General Properties

Use this tab to view and manage the general properties of a validation rule object. These properties include:

Name - the name of the validation rule as it is used for display and identification purposes throughout the DirX Identity Provisioning system.

Description - a description for the validation rule.

Type - the fixed type value ValidationRule (read-only).

Operation - the validation operation type. Possible selections include:

- **Accept** - accepts imported members and changes the member state to "IGNORED"
- **Cleanup** - performs a cleanup operation on imported members by setting the member state to "DELETED"
- **Validate** - analyze attribute-based privileges in a target system and add the accounts found into appropriate groups in the state "IMPORTED"

Is Active - whether (checked) or not (unchecked) the rule can be executed.
Note that this flag is automatically reset if the rule is deleted.

System Default - whether (checked) or not (unchecked) the object is a default object that cannot be changed (the **Edit** action is not available). Copy the object and then modify it according to your needs.

Related Topics

- Consistency Rule - General Properties
- Rules

- Operations
- Provisioning Rule -General Properties
- Validation Rule - Accounts
- Validation Rule - Groups

A.5.1.6.7. Validation Rule - Accounts

Use this tab to view and manage the filter properties that restrict the number of accounts on which the validation rule operates. These properties include:

Search Base - the base from which to start the search. The search base must be all or a part of an accounts tree in a target system.

Search Filter - the LDAP filter condition.

Search Scope - the scope to be searched. Possible values include:

- **Subtree** - the entire subtree belonging to the search base.
- **One Level** - only the objects at the level directly under the search base.
- **Base Object** - only the search base object itself.

You can use the **+** and **-** buttons to define multiple combined LDAP filters.

Related Topics

- Consistency Rule - General Properties
- Rules
- Operations
- Provisioning Rule -General Properties
- Validation Rule - General Properties
- Validation Rule - Groups

A.5.1.6.8. Validation Rule - Groups

Use this tab to view and manage the filter properties that restrict the number of groups on which the validation rule operates. These properties include:

Search Base - the base from which to start the search. The search base must be all or a part of a group tree in a target system.

Search Filter - the LDAP filter condition.

Search Scope - the scope to be searched. Possible values include:

- **Subtree** - the entire subtree belonging to the search base.
- **One Level** - only the objects at the level directly under the search base.
- **Base Object** - only the search base object itself.

You can use the **+** and **-** buttons to define multiple combined LDAP filters.

Related Topics

- Consistency Rule - General Properties
- Rules
- Operations
- Provisioning Rule -General Properties
- Validation Rule - General Properties
- Validation Rule - Accounts

A.5.1.7. Operations

Use this tab to display the properties for an operations folder. The operations folder contains either operation entries (Executable Operation, Java Class, JavaScript Operation) or other operations folders for improving the ordering and grouping of operations.

The property items shown here include:

Name - the displayed name of the operations folder.

Description - the description for the operations folder.

Related Topics

Managing Operations
Executables
Java Class
JavaScript Action

A.5.1.7.1. Executables - General Properties

Use this tab to view and manage the properties of an executable operation object. These properties include:

Name - the name of the executable operation as it is used for display and identification purposes throughout the DirX Identity Provisioning system.

Description - a description for the executable operation.

Type - the fixed type value Executable (read-only).

Is Active - whether (checked) or not (unchecked) the operation can be executed.

System Default - whether (checked) or not (unchecked) the object is a default object that cannot be changed (the **Edit** action is not available). Copy the object and then modify it according to your needs.

Parameters - Each line represents one of the parameters the operation requires:

- **Name** - name of the parameter
- **Type** - type of the parameter (currently, only string is available)
- **Context** - type of the parameter. Select one of the values from the drop down list
- **Description** - description of the parameter

Execute asynchronously - whether the executable is executed synchronously (unchecked) or asynchronously (checked).

Is script - whether (checked) or not (unchecked) the target to execute is a script and therefore needs to be executed with a script interpreter.

Script Interpreter - the script interpreter to be used, if **Is script** is checked. Define your special script interpreter, otherwise the platform-specific default interpreter is used.
Windows default: cmd.exe start /C
Linux default: /bin/sh -c

Path - the path to the executable to be called. The content of this field can be operating system-dependent.

Related Topics

- Java Class - General Properties
- JavaScript - General Properties

A.5.1.7.2. Java Class - General Properties

Use this tab to view and manage shows the properties of a Java class object, including:

Java Class Name - the name of the Java class as it is used for display and identification purposes throughout the DirX Identity Provisioning system.

Description - the description for the Java class.

Is Active - whether (checked) or not (unchecked) the operation can be executed.

System Default - whether (checked) or not (unchecked) this object is a default object that cannot be changed (the **Edit** action is not available). Copy the object and then modify it according to your needs.

Related Topics

- Executables - General Properties
- Java Method - General Properties
- JavaScript - General Properties

A.5.1.7.3. Java Method - General Properties

Use this tab to view and manage the properties of a Java method object, including:

Java Method - the name of the Java method as it is used for display and identification

purposes throughout the DirX Identity Provisioning system.

Description - a description for the Java method.

Type - the fixed type value Java (read-only).

Is Active - whether (checked) or not (unchecked) the operation can be executed.

System Default - whether (checked) or not (unchecked) the object is a default object that cannot be changed (the **Edit** action is not available). Copy the object and then modify it according to your needs.

Parameters - each line represents one of the parameters the operation requires.

- **Name** - the name of the parameter.
- **Type** - the type of the parameter (currently, only string is available).
- **Context** - the type of the parameter. Select a value from the drop-down list.
- **Description** - the description of the parameter.

Related Topics

- Executables - General Properties
- Java Class - General
- JavaScript - General Properties

A.5.1.7.4. JavaScript - General Properties

Use this tab to view and manage the properties of a JavaScript object. These properties include:

Name - the name of the JavaScript object as it is used for display and identification purposes throughout the DirX Identity Provisioning system.

Description - a description for the JavaScript object.

Type - the fixed type value JavaScript (read-only).

Is Active - whether (checked) or not (unchecked) the operation can be executed.

System Default - whether (checked) or not (unchecked) the object is a default object that cannot be changed (the **Edit** action is not available). Copy the object and then modify it according to your needs.

Parameters - each line represents one of the parameters the operation requires.

- **Name** - the name of the parameter.
- **Type** - the type of the parameter (currently, only string is available).
- **Context** - the type of the parameter. Select a value from the drop-down list.
- **Description** - the description of the parameter.

Related Topics

- Executables - General Properties
- Java Class - General Properties
- JavaScript - Implementation

A.5.1.7.5. JavaScript - Implementation

Use this tab to display and edit the content of the JavaScript implementation. Click the **Edit** button to make the text editable. For more information on the editor features, see the help topic for the text editor.

You can export the content of the object into a file by clicking **Export...** and then entering a file name in the resulting file dialog. To replace the current content with another text in a file, click **Import...** and then enter the name of the file in the resulting file dialog.

Related Topics

- Executables - General Properties
- Java Class - General Properties
- JavaScript - General Properties

A.5.1.8. Password Policies

DirX Identity provides a set of password policies that restrict password changes. The password policies are handled in the DirX Identity Services layer and not in the LDAP directory itself.

The properties shown here are:

Name - the name of the password policy.

Description - the description for the password policy.

Customization class - the names of classes that implement customized password checks or password generations for this policy; see the chapter "Customizing Password Management" in the *DirX Identity Customization Guide* for details.

Active - whether (checked) or not (unchecked) the password policy is active.

Default policy - whether (checked) or not (unchecked) the password policy is to be used for all users that do not have an explicit password policy assigned.

The properties related to Windows compatibility checks are:

Must meet Windows password complexity requirements - whether (checked) or not (unchecked) the password policy requires passwords to comply with the Microsoft Windows (Active Directory) password complexity requirements. You can set additional password policies if required. The Windows password complexity requirements are:

- The password must be at least eight characters long.
- The password must contain characters from at least three of the following five categories:
- Uppercase letters of European languages
- Lowercase letters of European languages
- Digits
- Any Unicode character that is categorized as an alphabetic character but is not uppercase or lowercase. This includes Unicode characters from Asian languages.
- Special characters from the list: ~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/. Currency symbols such as the Euro or British pound are not counted as special characters for this policy setting.
- The password must not contain one of the user's Windows account names or three or more consecutive characters from one of his Windows display names.

Attributes for Windows account name checks - the user or account attributes in the DirX Identity database that contain the Windows account name; the default value is **account:dxrName**.

Attributes for Windows display name checks - the user or account attributes in the DirX Identity database that contain the Windows display name or parts thereof; default is **account:cn**.

When a user sets his password or an administrator resets the password of another user, the name checks are performed against all of the user's Windows accounts. This doesn't capture accounts that are assigned only later on after the password change. To make sure that the password complies with future accounts, you can define that the checks should also be performed against some user attributes that hold the Windows account or display names of the future accounts. For example, if the accounts' display names are comprised of the user's first name and last name, enter **account:cn; user:givenName; user:sn** for the display name checks.

You can also perform the checks against one or more portions of an attribute value instead of the entire value. Just add a regular expression (java.util.regex syntax) to the attribute name that captures the desired parts. For example, **user:cn:([a-z]+).*** extracts the beginning of the common name up to the first non-ASCII letter.

History check-related properties are:

Number of passwords in history - the number of passwords that are stored in the history record. The user must define a password that is not in the history.

Don't store current password if check disabled - disable storage of the current password. Whenever a user changes his password or his password is reset by an administrator, the new password is usually stored in the user's password history even if the password history is disabled (that is, if **Number of passwords in history** is **0**). This operation is performed to avoid issues in systems with one or more Windows Password Listeners. You can disable storing the password here, but do it only if you have not deployed any Windows Password Listeners. The flag is ignored if the history is enabled.

Aging check-related properties are:

Maximum age - the age after which the user's password expires. The user must change the password after this period.

Expiration warning time - the amount of time before **Maximum age** is reached at which the user should be notified of the impending expiration. The password expiration notification workflow uses this value to calculate when the user needs to be warned about a password that is about to expire.

Character check-related properties are:

Minimum number of characters - the minimum length of the password.
Note: if the Windows compatibility flag is set, a number smaller than 6 is ignored.

Maximum number of characters - the maximum length of the password.

Minimum number of non-alphanumeric characters - the number of non-alphanumeric characters required for the password. Non-alphanumeric characters comprise all characters that are not letters and numbers.

Minimum number of numeric characters - the number of numeric characters required for the password.

Minimum number of special characters - the number of special characters required for the password. Special characters comprise all characters besides letters.

Minimum number of upper case characters - the number of uppercase characters required for the password.

Minimum number of lower case characters - the number of lowercase characters required for the password.

Not allowed characters - the list of characters that are not allowed in passwords.

Not allowed substrings - the list of substrings that are not allowed in passwords. You can define one or more substrings per line; separate substrings with space characters.

A.5.1.9. SoD Policies

SoD policies specify user-privilege assignments that constitute conflicts of interest. The SoD checking algorithm checks the user-privilege assignments for compliance with one or more SoD policies and reports any discovered violations.

You can structure SoD policies in folders, define SoD policies and view the resulting SoD exceptions.

Related Topics

- SoD Exception
- SoD Policy - General

- SoD Policies Folder

A.5.1.9.1. SoD Policies Folder

This tab shows the properties for a segregation of duties (SoD) policies folder. It can contain other SoD policy folders or SoD policy objects.

The property items shown here include:

Name - the displayed name of the folder.

Description - the description for the folder.

Related Topics

- SoD Exception
- SoD Policies Folder

A.5.1.9.2. SoD Policy - General

An SoD policy in DirX Identity Provisioning represents an object that defines conflicting privileges of users. An example of a conflict of interest is a user who has the right to order a piece of equipment in parallel with the right to approve the order.

For each pair or group of privileges, define a separate SoD policy object. Activate the Segregation of Duty Check flag at the domain objects to enable segregation of duty checks for this domain.

The available attributes are:

Name - the name of the object.

Description - a description for the object.

Is active - whether (checked) or not (unchecked) or not the policy is used. If you activate an SoD policy, the flag **Has conflicting privileges** for the corresponding privileges is set. All users are also checked as to whether there are already existing SoD violations. If SoD violations are detected, the corresponding approval workflows are started. If the privilege is no longer part of an SoD policy, the flag is reset automatically.*
Note:* You must re-start DirX Identity Manager for a change to this field to take effect.

Conflicting Privileges - the list of privileges that are in conflict, which means that any of the privileges in the list is in conflict with any other privilege in this list (for example, if you define privilege p1, p2 and p3, then these conflicts are assumed: p1 <-> p2, p1 <-> p3, p2 <-> p3). If you want to define pairs of conflicting privileges, define individual SoD policies.

Related Topics

- SoD Exception
- SoD Policy - General

A.5.1.9.3. SoD Exception

An SoD policy in DirX Identity Provisioning can result in approved exceptions to the policy. These objects are located under the relevant SoD Policy object.

The available attributes are:

Name - the name of the object.

Description - a description for the object.

User - the relevant user.

Accepted Violations - the privilege combination that caused the SoD violation.

Approval Details - the corresponding approval details.

Related Topics

- SoD Policy - General
- SoD Policies Folder

A.5.2. Delegations Folder

This tab displays the properties for a delegation folder. The delegation folder contains complete delegations (that is, a definition of which access rights are delegated from a user to another user) as well as the corresponding access rights in the folder **Access Rights** and its subfolders for **approve**, **grant**, **modify** and **read** access rights.

The property items shown here are the following:

Name - the displayed name of the delegations folder.

Description - the description for the * Delegations folder.

Related Topics

- Delegations
- Access Rights

A.5.2.1. Delegations

In DirX Identity Provisioning, a delegation represents an object that collects the access rights inherited from a user.

The available attributes are:

Name - the name of the delegation object.

Description - a description for the delegation object. The description should denote the main reason for this delegation.

Delegator - the person that delegates access rights to the substitute.

Substitute - the person that gets access rights from the delegator.

Start Date - the start date if the delegation is time restricted.

End Date - the end date, if the delegation is time restricted.

Delete Date - the date on which this delegation object will be deleted from the Identity Store (Provisioning Configuration).

Access Right Link - links to all access rights that belong to this delegation object.

Prevent delegation - whether (checked) or not (unchecked) further delegation is not allowed.

Originated from - the object from which this object was derived. It is either a copy or a subset of it.

Related Topics

- Delegations Folder
- Access Rights

A.5.2.2. Access Rights

In DirX Identity Provisioning, an access right represents an object that defines the operations that can be performed on the list of resources.

The available attributes are:

Name - the name of the access right object.

Description - a description for the access right object. The description should denote the main reason for this access right.

Originator - the access right from which this object was derived. It is either a copy or a subset of it.

Operation - the operations that can be performed on the resources. Possible operations are:

- **read** - controls read access to the resources
- **modify** - controls whether the subject is allowed to modify the resources
- **grant** - allows assignment of resources (privileges = roles, permissions, groups) to subjects (users)
- **approve** - controls approval of user-to-privilege assignments

Resource link - the list of resources (users, roles, permissions or groups).

Related Topics

- Delegations Folder
- Delegations

A.6. Certification Campaign View

A.6.1. Campaign

A.6.1.1. Campaign Folder

A campaign folder represents a campaign.It provides the following containers:

- **Notifications** - this folder contains all notifications that are used for this campaign.
- **Certifications** - this folder contains all certification tasks.The name depends on the campaign type (see **General Properties**, property **Type**).If users are to be certified, the folder is named **User Certification**.If privileges are to be certified, the container is named **Privilege Certification**.

Related Topics

- Certification Campaign View
- Campaign- Status
- Campaign - User Filter
- Campaign - Privilege Filter
- Campaign - User Hooks
- Notification - General
- Notification - Mail Body
- Certification - General
- Certification - Status
- Certification - Approvers
- Certification - Attributed Assignments Certification - Simple Assignments

A.6.1.2. Campaign - General Properties

Use this tab to view and manage the general properties of a campaign. These properties include:

Name - the campaign name. Define it carefully: it is displayed wherever the campaign is identified, especially in Web Center.

Description - a more detailed description for the campaign. It is also shown in DirX Identity Web Center and in reports.

Type - the campaign type. Select one of the following values:

- **User Certification** - users are certified. One certification task is created for each user

that matches the campaign's subject filter (see the User Filter tab). The task requires all manually-assigned privileges of this user to be certified.

- **Privilege Certification** - privileges are certified. One certification task is created for each privilege that matches the campaign's privilege filter. The task requires the manual assignments of all users to this privilege to be certified.

Owner - the user who is considered to be the owner of this campaign. **Owner** receives notifications on certain events; for example, when the campaign was started, finished or when some error occurs. He or she also has the access rights to change campaign entries.

Days before due date - the number of days before the certification campaign due date at which reminder notifications are to be sent, if templates for reminders are available in the certification campaign.

Interval between reminders (hours) - the number of hours between reminder notifications. This interval is suggested to the Certification Campaign Controller workflow. The real reminder notification interval depends on how often the Certification Campaign Controller workflow is executed. For example, if the workflow is scheduled to run every 24 hours, and the interval between notifications is set to 1 hour, reminder notifications will be sent every 24 hours.

Apply Changes - the action to be performed at the end of the campaign (after the due date). Select one of the following options:

- **Do not revoke any rejected privileges** - changes are not applied to user or privileges. They are only visible in the campaign entries and in reports.
- **Revoke only rejected privileges that were manually assigned** - changes are applied only for assignments that are explicitly rejected. Uncertified entries are left untouched.
- **Revoke all manual assigned privileges that are rejected or left uncertified** - changes are applied for all assignments that are not explicitly accepted. All assignments that have been explicitly rejected or left untouched are removed.
- **Review the revocation of all manually assigned privileges that are rejected or left uncertified** - changes are not applied. An approval workflow is started for each rejected, changed or untouched privilege.

Related Topics

Campaign- Status
Campaign - User Filter
Campaign - Privilege Filter
Campaign - User Hooks

A.6.1.3. Campaign - Status

Use this tab to display the status of a campaign. Status properties include:

State - the certification campaign's state. Possible states are:

Campaign is in preparation - PREPARING - the default state of a campaign that is ready to

start. This state changes after the start date is reached and the Certification Campaign Controller workflow has set up all of the certification entries.

Campaign is running - RUNNING - the campaign is running (the start date is reached) and the certification entries have been set up successfully.

Campaign failed to start - FAILED.PREPARE - the campaign failed to start. The reason for the failure is available in **Logs**.

Campaign finished successfully - SUCCEEDED - the campaign successfully finished.

Campaign start date and end date are expired - FAILED.EXPIRED - the campaign's due date has passed and not all certifications are finished.

Campaign is marked for deletion - DELETED - the campaign's **Status Expiration Date** has passed and the campaign is marked for deletion.

Start Date - the date at which the certification must start. When this date is reached, the Certification Campaign Controller creates a certification task for each subject (user to be certified) of the campaign.

Approval Period - the duration of the certification. This duration is added to **Start Date** to determine **Due Date** value at which the approval period will end.

Due Date - the date at which the certification campaign is planned to end. This date is calculated with **Start Date** and **Approval Period** and can be overwritten later on by the administrator with another date value. After this date, changes are applied and certifications are finished with the state **FINISHED** or **FAILED.EXPIRED** depending on the settings in **Apply Changes** (see the General tab). If a value is already set by the administrator, this value will not be overwritten by the Certification Campaign controller, but if **Recurring Certification Campaign** is enabled, this value will be overwritten with **Start Date** + **Approval Period**.

End Date - the date at which the certification campaign actually ended. This field is populated when the campaign is finished (when changes are applied).

Status Expiration Date - the date at which certification LDAP entries should be physically deleted. By default, this field is not set at campaign startup. The administrator can set it manually. When campaign is finished and the field is empty, the campaign workflow sets a default status expiration date of the current date plus 30 days. The administrator can still change this value later on.

Recurring Certification Campaign - the interval between recurring certification campaigns. If no value is set, the campaign will run only once. If values are set, the Certification Campaign controller will move the current successful campaign to the _Archive folder and will start the campaign again. Note that the **Due Date** will be set to the default value: **Start Date** + **Approval Period**.

Logs - important informational messages generated from the Certification Campaign Controller workflow, such as warnings and errors.

Related Topics

- Certification Campaign View
- Campaign - User Filter
- Campaign - Privilege Filter
- Campaign - User Hooks

A.6.1.4. Campaign - User Filter

Use this tab to define the users that are to be certified in this campaign. The Certification Campaign Controller evaluates them at the campaign start. Later changes are silently ignored. Fields in this tab include:

Filter Base - the LDAP search base for users to be certified in this certification campaign. This field is mandatory for a user certification campaign.

User Filter - the LDAP search filter for users to be certified in this certification campaign. This field is mandatory for a user certification campaign.

You can use this filter to create certification campaigns for users with high risk. Use the **dxrRskLevel** attribute for values 0 (normal risk) and 3 (high risk).

Related Topics

- Certification Campaign View
- Campaign- Status
- Campaign - Privilege Filter
- Campaign - User Hooks

A.6.1.5. Campaign - Privilege Filter

Use this tab to restrict the set of privilege assignments that are to be certified in a campaign. The Certification Campaign Controller evaluates them at the campaign start. Later changes are silently ignored.

Filter Base - the LDAP search base for privileges that are to be certified in this campaign. This field is mandatory for a privilege certification campaign and optional for a user certification campaign.

Privilege Filter - the LDAP search filter for privileges that are to be certified in this campaign. This field is mandatory for a privilege certification campaign and for a user certification campaign.

These fields are optional for user certification campaigns. If they are empty, all assignments for the users that match the user filter are certified.

Related Topics

- Campaign - General Properties
- Certification - Status

- Campaign - User Filter
- Campaign - User Hooks

A.6.1.6. Campaign - User Hooks

Use this tab to customize a campaign with user hooks. You can create campaign user hooks that change standard Certification Campaign Controller operations like finding approvers or subjects (users or privileges to be certified) or sending emails. A campaign user hook must implement specific interfaces. For details, see the DirX Identity Use Case Document *Access Certification*.

Fields in this tab include:

Find Approvers - a Java package and class name of a user hook for selecting the approver for users (or privileges) to be certified. The specified user hook overrides the default behavior of the Certification Campaign Controller. This user hook is particularly useful when there is more than one approver for a user or privilege to be certified.

Approval Sequence - the execution order for the approval tasks. This field is evaluated when there is more than one approver for a user or privilege to be certified (called sub-certifications). The user hook needs to return a tree of approvers.

Bottom-up (Top Manager approves last) - approval starts with the leaves and ends with the top of the tree (the default).

Top-down (Top Manager approves first) - approval starts with the top and continues to the leaves. Note: With this setting, multiple parallel approval tasks can occur at the end.

Find Subjects - the Java package and class name of a user hook for locating the subjects of the campaign: the users or privileges to be certified. The specified user hook overrides the default behavior of the Certification Campaign Controller.

Limit Resources - the Java package and class name of a user hook for restricting the resources used in the campaign. For a user campaign, the resources are the assigned privileges. This specified user hook overrides the default operation of the Certification Campaign Controller.

Send Email - the Java package and class name of a user hook for changing the content of a notification or canceling the send email action.

Campaign Creator - the Java package and class name of a user hook for completely overriding the create campaign action of the Certification Campaign Controller.

Related Topics

- Campaign - General Properties
- Certification - Status
- Campaign - User Filter
- Campaign - Privilege Filter

A.6.2. Notifications

A.6.2.1. Notification - General

Use this tab to view and manage the general properties of a notification. These properties include:

Name - the notification name.

Type - the notification type, which is one of the following values:

- **Campaign Start** - the notification is sent to the campaign owner when the campaign start date is reached. It should contain details about the campaign.
- **Approval Start** - the notification is sent to the certification approvers when the start date of their certification task is reached.
- **Approval Remind** - the notification is sent during the campaign to the certifications approvers to remind them about their tasks.
- **Approval Timeout** - the notification is sent at the end of a certification task (when the due date is reached). The notification should contain details about not certified subjects (users or privileges).
- **No Approver** - the notification is sent to the campaign owner when no approver was found for a user or privilege to be certified. The notification should contain details about the entry to be certified.
- **Prepare Error** - the notification is sent to the campaign owner if errors occurred at the start of the campaign. It should contain details about the campaign and especially on the errors.
- **Assignment Rejected** - the notification is sent to the certified user when at least one of the privilege assignments has been rejected. It should contain details about the campaign and the rejected assignments.
- **Campaign End** - the notification is sent to the campaign owner when the campaign end date is set. It should contain details about the campaign.

From - the email address to be used as the notification sender; for example, the address of the campaign owner **${campaign.owner.mail}**.

To - one or more email addresses for the notification recipients; for example, the address of the campaign owner (**${campaign.owner.mail}**) or the address of an approver if the email is about a certification task: **${approver.mail}**.

CC - carbon copy. See **To**.

BCC - blind carbon copy. See **To**.

Subject - the email subject heading. Extract the necessary information from the campaign object; for example, the name: **Campaign ${campaign.name} has been started!**

Language - the language to be used for the email. The value can be taken from an attribute name of the user; for example, **${to.preferredLanguage}**.

Related Topics

- Campaign - General Properties
- Certification - Status
- Notification - Mail Body

A.6.2.2. Notification - Mail Body

Use this tab to define the body text of a notification. The content can contain plain text merged with placeholders taken from LDAP entries. Templates for general text and certification campaign-specific text are defined in the Provisioning View group at the following location: **Domain Configuration** -> *Domain Name* -> **Nationalization**.

The following list gives examples of how to include attributes of the campaign entries campaign, certification, approvers and subject:

```
#{Common Text.ParticipantSalutationTo}
${campaign.name} certification campaign is running.
Type: ${campaign.type.label}
Start time: ${campaign.startDate}
Due time: ${campaign.expirationDate}
User base: ${campaign.userFilterBase}
User filter: ${campaign.userFilter}
Privilege base: ${campaign.privilegeFilterBase}
Privilege filter: ${campaign.privilegeFilter}
<? for certification in ${certifications} ?>
Certification: '${certification.name}':
Attributed assignments:
<? for assignment in ${certification.assignmentLinks} ?>
    '${assignment.dn}'
<? endfor ?>
Simple assignments:
<? for resource in ${certification.resourceLinks} ?>
    '${resource.dn}'
<? endfor ?>
Subject information:
     '${certification.subjectLink.dn}'
<? endfor ?>
```

Related Topics

- Campaign - General Properties
- Certification - Status

- Notification - General

A.6.3. Certifications

A.6.3.1. Certification - General

Use this tab to display and edit the general properties of a user or privilege certification. The items shown here include:

Name - the name of the certification; typically the name of the user in a user certification or the name of the privilege in a privilege certification.

Subject - the subject of the certification (a link to the user to be certified in a user certification).

Type - the type of the certification (**dxrUser** for a User Certification).

Campaign - the campaign name; taken from the campaign entry (folder).

A certification task can contain child entries. Each child entry represents requested changes for existing assignments; for example, a changed end date or role parameter.

Related Topics

- Campaign - General Properties
- Certification - Status
- Certification - Approvers
- Certification - Attributed Assignments
- Certification - Simple Assignments

A.6.3.2. Certification - Status

Use this tab to display the status of a user or privilege certification. The items shown here include:

State - the state of the certification task. Possible values are:

- **PREPARED** - certification is ready to start.
- **RUNNING** - certification has started and is visible to approvers.
- **APPROVAL.FINISHED** - certification is finished and all assignments are either accepted or rejected.
- **AWAITING.FOR.APPLY.CHANGES** - changes defined in the certification are being applied; this is an intermediate state at the end of the campaign.
- **FINISHED** - certification has finished successfully. All changes are applied.
- **RETRY.PREPARE** - starting the certification failed and will be repeated.
- **FAILED.PREPARE** - starting the certification failed. At least one mandatory attribute is not correct.

- **FAILED.EXPIRED** - certification has expired. One or more assignments are not certified. Assignments neither accepted nor rejected are handled according the **Apply Changes** settings at the campaign entry (see the Campaign - General Properties tab).
- **FAILED.APPLY.CHANGES** - certification is finished but one or more errors occurred when the apply changes action was executed.

Ignore rejected - rejected assignments were not removed, changes were not applied. This is a read-only flag that is populated at the end of the campaign.

Ignore uncertified - uncertified assignments were ignored and remain active. This is a read-only flag that is populated at the end of the campaign.

Start Date - the date at which the certification was started. Typically the same as the campaign start date.

Due Date - the date at which the certification task needs to be finished.. Typically the same as the campaign due date.

End Date - the date at which the certification was finished with state **FINISHED** or **FAILED.EXPIRED**.

Related Topics

- Campaign - General Properties
- Certification - Status
- Campaign - User Hooks
- Notification - General
- Certification - General
- Certification - Approvers
- Certification - Attributed Assignments
- Certification - Simple Assignments

A.6.3.3. Certification - Approvers

Use this tab to display and edit the approvers for a user or privilege certification. Items shown here include:

Approvers - a list of all approvers for this user. By default, this is the user's manager. For other values, you must provide a Find Approver user hook.

Potential Approvers - a list of approvers that can act as a backup for the primary approvers. This field is empty by default.

Acting Approvers - the approver who actually accepted or rejected the assignments.

Related Topics

- Campaign - General Properties

- Certification - Status
- Certification - Attributed Assignments
- Certification - Simple Assignments

A.6.3.4. Certification - Attributed Assignments

Use this tab to display the assignments with attributes; that is, assignments that have a start and end date or role parameters. The assignments shown in this tab are read-only. To approve or reject them, use an appropriate certification campaign client such as DirX Identity Web Center.

Items shown here include:

Not Certified - a list with all attributed assignments that still need to be approved.

Accepted - a list with all assignments that are accepted.

Rejected - a list with all assignments that are rejected.

Related Topics

- Campaign - General Properties
- Certification - General
- Certification - Status
- Certification - Approvers
- Certification - Simple Assignments

A.6.3.5. Certification - Simple Assignments

Use this tab to display the assignments without attributes; that is, the assignments that do not have a start or end date or role parameters. The assignments shown here are read-only. To approve or reject them, use an appropriate certification campaign client such as DirX Identity Web Center.

Items shown here include:

Not Certified - a list with all assignments that still need to be approved.

Accepted - a list with all ass assignments that are accepted.

Rejected - a list with all assignments that are rejected.

Related Topics

- Campaign - General Properties
- Certification - General
- Certification - Status
- Certification - Approvers

- Certification - Attributed Assignments

A.7. Request Workflows View

The **Request Workflows** view shows all of the items necessary for handling and maintaining request workflows:

- The Configuration section lets you handle global objects and parameters.
- The Definitions section lets you define flexible request workflows.
- The Monitor section keeps a record of all workflow instances, either running or completed.

The DirX Identity Web Admin allows you to monitor the request workflow engine running in the Java-based Identity Server.

For each selected item in the tree, the respective property pages are shown on the right-hand side.Click **Edit** to modify property values.Note that saving your changes to the properties of an entry changes the way the request workflows operate.

Related Topics

- Configuration
- Monitor
- Workflows

A.7.1. Configuration

A.7.1.1. Activity Types

The Activity Types subtree contains the component descriptions for activities.

The subfolders collect, for example, the descriptions for automatic, error and people activities.

Related Topics

- Common Activities
- Service
- SMTP Service
- Certificate Service

A.7.1.2. Common Activities

The Common Activities subtree contains common activities that can be used by all workflows.

Currently only error activities are supported.

Related Topics

- Activity Types
- Service
- SMTP Service
- Certificate Service

A.7.1.3. Services Folder

The Services subtree comprises configuration objects that define global settings for all (request) workflows; for example, the mail server parameters.

Related Topics

- Activity Types
- Common Activities
- Service
- Certificate Service
- SMTP Service

A.7.1.4. Service

A service defines a resource used by the request workflow service. The property items shown here include:

Name - the displayed name of this object.

Description - the description of this object.

Location - the location of the service (a URL).
Example:
${services:[cn=Web Server]@dxrOptions(location)}/privilegeAssignment
In this case, the variable is built by searching under the services tree for an object with cn=Web Server. The attribute location is read and suffixed with the string '/privilegeAssignment' to build the correct URL.

Note: The location of **Workflows** -> **Confguration** -> **Services** -> **Web Server** is pre-configured to **http://localhost:8080**. As a result, the link presented by an approval request e-mail will only be usable for the machine on which the Web Center is installed, which is not desirable in production environments.

Change the location to **http://*host**:*port*", where *host* is the name or IP address of the machine on which the Web Center is installed and *port* is the port for the Web server (for Tomcat: **8080** if not configured otherwise).

Related Topics

- Activity Types

- Common Activities
- Certificate Service
- SMTP Service

A.7.1.5. Certificate Service

This service needs to be configured if you want to send encrypted e-mail notifications. It allows you to define the external LDAP directory server from which to retrieve the user certificates that are used by the SendMail workflow for sending encrypted e-mails. Retrieval of the certificate is done by a user-defined LDAP search operation that looks (apart from the user-defined search filter) for objects with a matching e-mail address in the LDAP attribute “mail”.

The property items shown here include:

Name - the displayed name of this object.

Description - the description of this object.

Owner - the owner of this object.

IP Address - the IP address (or host name) of the external LDAP server to search for user certificates. If **IP Address** is not specified, the LDAP server accessing the Provisioning tree is used. In this case, no additional bind operation is performed and therefore none of the parameters **Port**, **SSL**, **Authenticate**, **Username**, **Password** need to be defined.

Port - the port of the LDAP server to search for user certificates (default: **389**). This information only needs to be defined if you want to connect to an external LDAP directory.

SSL - whether (checked) or not (unchecked) to establish an SSL connection to the external LDAP directory.

Authenticate - whether (checked) or not (unchecked) to use simple authentication to the external LDAP server. If you check this flag, enter the information for simple authentication in the following fields:

Username - the user DN to authenticate to the LDAP server.

Password - the password of that user to authenticate.

Certificate Search Base - the base object to use for searching for user certificates

Certificate Search Filter - the filter to use for searching for user certificates. Note that this filter is internally combined with the following AND filter: “&(mail=*the user’s mail address*))”.

Certificate Attribute Name - the LDAP attribute name of the attribute that holds the certificate; for example, “userCertificate”

Refresh Interval [in h] - the time interval in hours (the default is **24**) after which the internal map that holds the e-mail address and the corresponding certificate is cleared. For performance reasons, the SendMail workflow uses an internal map to avoid having to

search for the user certificate with every incoming request. It searches for the certificate in LDAP only if it is not present in the map. After the given Refresh Interval, the content of the map is discarded to get up-to-date certificates in case they change in the external LDAP directory over time. Also keep in mind that when the certificates are changed in LDAP, you can use **Load IdS-J Configuration** to enforce a reset of the internal map.

Related Topics

- Activity Types
- Common Activities
- Services Folder
- Service
- SMTP Service

A.7.1.6. SMTP Service

SMTP Services let you define an SMTP server for e-mail notification.

The property items shown here include:

Name - the displayed name of this object.

Description - the description of this object.

Owner - the owner of this object.

Transport - the transport protocol to use for sending e-mails. The default value is **SMTP**. Another option is **GRAPH**, which allows you to use Microsoft Exchange Online service to send e-mails via Graph-API.

SMTP host - the host name of the SMTP server to which the mail is to be sent.

Port - the port of the SMTP server where the mail is to be sent (default: 25).

SSL - whether (checked) or not (unchecked) to establish an SSL connection to the external LDAP directory.

Protocols - comma-separated list of protocols supported for SSL encryption; for example, SSLv3, TLSv1.2.

Authenticate - whether (checked) or not (unchecked) to use simple authentication to the SMTP server. If you check this flag, enter the information for simple authentication in the following fields:

Username - an administrative account to authenticate to the mail server.

Password - the password for this administrative account to authenticate.

Encrypt Email - whether (checked) or not (unchecked) to send encrypted mail. If this field is enabled, be sure to set up the Certificate Service.

Send On Encryption failure - whether (checked) or not (unchecked) to send an unencrypted mail if there are problems when encrypting it. If this field and **Encrypt Email** are both enabled, be sure to set up the Certificate Service.

Map mail address - if you enter **dummy** into this field, then no mail is sent at all (use this feature if the SMTP server is not available, for example, when demonstrating the approval workflows on a notebook that is not connected to a mail server). If you enter another mail address (for example, your mail address), all mail messages are sent to this address, no matter what the calculated mail address is. This mode is useful when demonstrating the approval workflows. You cannot use this feature in production!



DirX supports mail servers that are either configured for anonymous access (the **Authenticate** flag is clear) or that have an administrative account whose credentials can be configured (the **Authenticate** flag is set). We do not support user-specific authentication (authentication as the user in the **To** field).

Graph API Tenant ID - the tenant ID to use when sending emails via Graph API.

Graph API Client ID - the client ID to use when sending emails via Graph API.

Graph API Client Secret - the client secret to use when sending emails via Graph API.

Graph API Sender UPN - the sender UPN to use when sending emails via Graph API.



When using Graph API for sending emails, the SMTP host, port, SSL and authentication settings are not used. Instead, the Graph API settings must be provided and the transport protocol must be set to **GRAPH**. It is required to register an application in Azure AD and grant it the necessary permissions to send emails on behalf of users (Mail.Send, Admin Consent). Also create a client secret for the registered application and provide the tenant ID, client ID and client secret in the respective fields. The sender UPN must also be provided and should correspond to a valid user in the tenant with a mailbox.



When using Graph API for sending emails, be aware that the email sending limits and throttling policies of Microsoft Graph API apply. With this release throttling responses and retry logic are not yet implemented. Therefore, this configuration is considered to be experimental.



Changing attributes in this section requires restarting the IdS-J Server. Running **Load IdS-J Configuration** (from the context menu at the workflow entry) is not sufficient because it only loads the workflow definitions.

Related Topics

- Activity Types
- Common Activities

- Service
- Certificate Service

A.7.1.7. SMS Gateway

Use this tab to define parameters for connection to an SMS Gateway service/server to send text messages. DirX Identity provides a sample implementation that connects to an HTTP service (**com.siemens.idm.jobs.util.SMSGatewaySimpleHttpPluginImpl**). If your SMS Gateway uses a different connection mechanism, you need to implement your own Java class. The class must implement the interface **com.siemens.idm.api.notification.SMSGatewayPlugin**.

The property items shown here include:

Name - the displayed name of this object.

Description - the description of this object.

Owner - the owner of this object.

Plug-in Class

Plug-in Classname - the full-qualified classname.

Destination URL

URL - the service URL

HTTP Method- the HTTP method which should be used: get or post.

Constant Parameters- constant parameters that should be part of the URL (*get* method) or in the body (*post* method).

Authentication

Authenticate - whether (checked) or not (unchecked) to use simple authentication to the SMS Gateway server. If you check this flag, enter the information for simple authentication in the following fields:

Username - an administrative account to authenticate to the Gateway server.

Password - the password for this administrative account to authenticate.

Parametername for Username - the parameter name in the URL (*get*) or body (*post*) for the **Username**.

Parametername for Password - the parameter name in the URL (*get*) or body (*post*) for the **Password**.

Text Message

Parametername for Text Message - the parameter name in the URL (*get*) or body (*post*) for

the text message.

Mobile Number

Parametername for Mobile Number - the parameter name in the URL (*get*) or body (*post*) for the mobile number.

Delete any signs in number - whether specific characters such as a hypehn (-) or a plus sign (+) should be deleted in the mobile phone number.

Proxy Server

Host Name - the host name or IP address if a proxy server has to be used

Port - the port number, if a proxy server must be used.

Note: changing attributes in this section requires restarting the IdS-J Server. Running Load IdS-J Configuration (from the context menu at the workflow entry) is not sufficient because it only loads the workflow definitions.

Note: DirX Identity supports SMS gateway services that are either configured for anonymous access (**Authenticate** is clear) or that have an administrative account whose credentials can be configured (**Authenticate** is checked). DirX Identity does not support user-specific authentication (authentication as the user in the **To** field).

Related Topics

- Activity Types
- Common Activities
- Service

A.7.2. Definitions

A.7.2.1. Workflow - General

This element allows you to configure and view the workflow structure. It is displayed in an intuitive graphical view. You can see all activities and their relationships.

Use the **Zoom** commands from the context menu to adjust the view. Use **Open** to view details.

Click **Edit** to add, modify or delete elements. After editing the structure, click **Save** to store the results or **Reset** to discard all changes.

For details, see the Workflow Editor description.

Note that changes of the workflow definition are only updated in the Java-based Identity Server if the server is restarted or a **Load IdS-J Configuration** is performed (from the context menu at the workflow entry).

Related Topics

- Activity - General
- Activity - Parameters
- Activity - Participants
- Activity - Participants Filter and Constraints
- Activity - Notifications
- Activity - Escalations
- Workflow - Workflow
- Workflow - When Applicable
- Notifications
- Workflow Editor

A.7.2.2. Workflow - Workflow

The top-level object that defines a workflow. It contains all activities.

Note that changes to the workflow definition are only updated in the Java-based Identity Server if the server is restarted or a **Load IdS-J Configuration** is performed (from the context menu at the workflow entry).

The property items shown here include:

Name - the displayed name of this object.

Description - descriptive text to be used below the title of the Web page. Use this field to describe what this page is about and give hints on how to use it. This field supports nationalization. You can use nationalized items and/or strings with expressions (containing workflow variables), but no expressions inside a nationalized item.

Workflow Type - the type of workflow (either **Request** or **Certification**). Use this field to separate workflow types for processing in multiple Java-based Servers.

Monitor Display - the display name of instances in the Monitor subtree. Use variables to get individual instance names. See the section "Request Workflows" in the *DirX Identity Application Development Guide* for information on how to use variables.

Examples:
${workflow.subject.sn} ${workflow.subject.givenName}
results in "Smith Joe" as the display name.

<? if ${workflow.subject.sn} == null ?> Under construction (${UID}) <? else ?>${workflow.subject.sn},${workflow.subject.givenName} <? endif ?>
results if sn is empty to "Under construction 12323432" or if sn is filled to "Smith Joe".

Owner - the responsible administrator for this workflow. Mails from this workflow can be sent on behalf of this person (**From** field) and this person can be notified if instances of this workflow fail.

Error activity - a link to the relevant error activity for this workflow if something goes wrong

at the workflow level. By default, this link points to the configured General Error activity (Workflows -> Configuration -> Common Activities -> Default). Alternatively, you can create and link your own general error activity (for example under Workflows -> Configuration -> Common Activities -> *my-project*) or you can create an error activity local to the workflow.

Active - whether (checked) or not (unchecked) the workflow engine and the Web Center can use this workflow.

Timeout - the time at which the workflow times out. If you enter 0, the workflow does not time out.

Related Topics

Activity - General
Activity - Activity
Activity - Parameters
Activity - Participants
Activity - Participants Filter and Constraints
Activity - Notifications
Activity - Escalations
Workflow - General
Workflow - When Applicable

Notifications

A.7.2.3. Workflow - When Applicable

Use this tab to define the applicability of this workflow, if the workflow is not directly defined via a link.

Note that changes to the workflow definition are only updated in the Java-based Identity Server if the server is restarted or a **Load IdS-J Configuration** is performed (from the context menu at the workflow entry).

The property items shown here include:

Operation - the type of operation. Available types are:

- Create - workflow to create an object or an assignment
- Modify - workflow to modify an object or an assignment
- Delete - workflow to delete an object or an assignment
- Re-approve - workflow for re-approval of an assignment
- SoD - mitigation workflow for segregation of duties approvals

Subject - the DirX Identity object type on which this workflow operates.

Resource Type - the type(s) of resource(s) this workflow definition is for.

Note: if no resource is defined, a "not present" condition is automatically entered into the resulting XML definition.

Priority - the priority of this workflow definition. Use this field to define a sequence of workflows if the selection is not unique.

Condition - an optional filter definition. Select "none" if no additional filter is necessary. Use the LDAP filter editor to define a more specific applicability of this workflow. You can define any combination of expressions.

Some examples of conditions are:

initiator.c equals DE
which is converted to the internal expression
${requestcontext.initiator.c}="DE"
It compares whether the c attribute of the initiator is equal to "DE".

subject.c equals DE
which is converted to the internal expression
${requestcontext.subject.c}="DE"
It compares whether the c attribute of the subject is equal to "DE".

${requestcontext.resource.dxrassignto@description} equals MyRoleDescription
is an example showing that you can define internal expressions of any kind. Just enter it into the corresponding text field.
It compares the description attribute of the resource with the value "MyRoleDescription".

Related Topics

Activity - General
Activity - Activity
Activity - Parameters
Activity - Participants
Activity - Participants Filter and Constraints
Activity - Notifications
Activity - Escalations
Workflow - General
Workflow - Workflow

Notifications

A.7.2.4. Activity - General

Use this element to control the activity's structure. It is displayed in an intuitive graphical view. You can see all sub-activities and their relationships.

Use the **Zoom** commands to adjust the view.

Click **Edit** to activate or deactivate elements. After editing the structure, click **Save** to store the results or **Reset** to discard all changes.

For details, see the Activity Editor description.

Note that changes to the workflow definition are only updated in the Java-based Identity Server if the server is restarted or a **Load IdS-J Configuration** is performed (from the

context menu at the workflow entry).

Related Topics

Activity - Activity
Activity - Parameters
Activity - Participants
Activity - Participants Filter and Constraints
Activity - Notifications
Activity - Escalations
Workflow - General
Workflow - Workflow
Workflow - When Applicable

Activity Editor
Notifications

A.7.2.5. Activity - Activity

Use this tab to define an activity's general properties.

Note that changes of the workflow definition are only updated in the Java-based Identity Server if the server is restarted or a **Load IdS-J Configuration** is performed (from the context menu at the workflow entry).

The property items shown here include (note: not all activities contain all fields):

Name - the displayed name of this object.

Title - the text to be used as the Web page title. This field is only available for people activities.
Note: This field supports nationalization. You can use nationalized items and/or strings with expressions (containing workflow and activity variables), but no expressions inside a nationalized item.

Description - the description of this object. This field supports nationalization..You can use nationalized items and/or strings with expressions (containing workflow and activity variables), but no expressions inside a nationalized item.

Start Condition - the condition that is necessary to start this activity.

Name - the name that is displayed in the graphical view at the connecting line.
Activity - the predecessor activity (select it from the list)
Trigger - define a variable (State or applicationState), a condition and a value.

Is final - whether (checked) or not (unchecked) the activity is handled as the final (end) activity for this workflow. Set this flag for all activities that are connected to the end node.

Client Signature - whether (checked) or not (unchecked) the activity is protected with client signature (only visible for people activities). In this case, the actor for this activity must provide a valid signature from his card (works only on Windows platforms and with Internet Explorer).

Note: you must also set the **Enable Auditing for** -> **Request Workflows** flag and the **Enable Client Signature for** -> **Request Workflows** flag at the domain object to enable this feature.

Content read only - whether (checked) or not (unchecked) the order data can be changed. When checked, the approver can accept or reject the data but he cannot change it. When clear, the approver is allowed to modify some or all of the data.
Note that setting the **Approval content read only** flag at the domain object overrides this flag.

Resource family - the type of resource this activity needs to run. It runs only on servers or worker containers that are associated with the same resource family(ies).

Timeout - the timeout for the activity. If you enter 0, then the workflow does not time out.

Retry Limit - the number of retries the server will perform.

Wait Before Retry - the time between retries.

Related Topics

Activity - General
Activity - Parameters
Activity - Participants
Activity - Participants Filter and Constraints
Activity - Notifications
Activity - Escalations
Workflow - General
Workflow - Workflow
Workflow - When Applicable

Notifications

A.7.2.6. Activity - Parameters

This tab contains a list of parameters for configuring the specific activity.

Note that changes of the workflow definition are only updated in the Java-based Identity Server if the server is restarted or a **Load IdS-J Configuration** is performed (from the context menu at the workflow entry).

The parameters displayed in this tab depend on the activity type. For some activities (the Apply activities), common parameters are available. These parameters include:

Apply subject order - whether (checked) or not (unchecked) the activity performs the operation defined in the subject order. This is either a creation request or a request to modify attributes (created by previous activity steps or workflows):

- It creates a new object if requested.
- It adds new attributes.
- It modifies or deletes existing attributes (if the object did already exist).

- If the target object is a user: it checks whether one or more attributes are Permission Parameters. If this is true, it performs a privilege resolution.

Evaluate assignments - whether (checked) or not (unchecked) the activity evaluates all assignments as defined in the according order (created by previous activity steps or workflows):

- The activity performs a privilege resolution and initiates (only if **Enable Real-time Provisioning** is set at the target system) the necessary updates by creating update events.

Start approval workflows - whether (checked) or not (unchecked) the activity performs for all assignments defined in the according order (created by previous activity steps or workflows) a privilege resolution and starts (only if **Allow Real-Time Provisioning** is set at the target system) the necessary updates by creating update events for the target systems.

Class name - the Java that can intercept the preprocessing and postprocessing of the apply activity. The class must implement the interface **siemens.dxr.service.order.api.ApplyChangesUserHook** from **dxrServices.jar** and must be deployed in **${DIRXIDENTITY_INST_PATH}/ids-j-*/confdb/common/lib**.

See the section "Implementing a User Hook for an applyChange Activity" in the *DirX Identity Application Development Guide* for more information.

Track changes - whether (checked) or not (unchecked) provisioning actions for accounts and groups caused by the order are noted at the workflow instance (workflow context). A **Check Provisioning** activity can then check whether these provisioning actions have completed. Note that this activity is named **Wait for completed provisioning** in the sample workflows.

Track changes in child workflows - whether (checked) or not (unchecked) the child workflow propagates all provisioning actions for accounts and groups to the parent workflow. A **Check Provisioning** activity in the parent workflow can then check whether these provisioning actions have completed. Note that this activity is named **Wait for completed provisioning** in the sample workflows.

The following table shows the standard activity types and corresponding flag settings:

Activity Name	Apply subject order	Evaluate assignments	Start approval workflows	Track changes	Track changes in child workflows
Apply object	X	-	-	-	N/A
Apply user with assignments	X	X	X	-	N/A
Apply approved privileges	-	X	-	-	N/A.
Apply order	X	(X)	(X)	X	X

Legend: **X**=set, **-**=not set, **(X)**=implicitly set, N**/A**=not available

Apply approved privileges

Implements a privilege assignment after approval is performed.

This type is based on a common implementation that is controlled via flags. The flags are correctly set for this activity type, but you can change them if required. These settings are:

- **Apply subject order** - not set
- **Evaluate assignments** - set
- **Start approval workflows** - not set
- **Class name** - not set
- **Track changes** - not set

For a detailed description of these flags, see the common parameters description.

Apply object

Applies the subject order, which can be either a creation or a modification request. Use this type for creation of objects other than users (roles, etc.). You can also use it for modification of any object type (including users).

This type is based on a common implementation that is controlled via flags. The flags are correctly set for this activity type, but you can change them if required. The settings are:

- **Apply subject order** - set
- **Evaluate assignments** - not set
- **Start approval workflows** - not set
- **Class name** - not set
- **Track changes** - not set

For a detailed description of these flags, see the common parameters description.

Apply orders from ticket

Applies the subject order, which can be either a creation or a modification request for a user. It also evaluates assignments and starts (if required) approval workflows.

This type is based on a common implementation that is controlled via flags. The flags are correctly set for this activity type, but you can change them if required. The settings are:

- **Apply subject order** - set
- (**Evaluate assignments** - implicitly set (not changeable))
- (**Start approval workflows** - implicitly set (not changeable))
- **Class name** - not set
- **Track changes** - set
- **Track changes in child workflows** - set

For a detailed description of these flags, see the common parameters description.

Apply user with assignments

Applies the subject order, which can be either a creation or a modification request for a user. It also evaluates assignments and starts (if required) approval workflows.

This type is based on a common implementation that is controlled via flags. The flags are correctly set for this activity type, but you can change them if required. The settings are:

- **Apply subject order** - set
- **Evaluate assignments** - set
- **Start approval workflows** - set
- **Class name** - not set
- **Track changes** - not set

For a detailed description of these flags, see the common parameters description.

Approve Creation:

Attributes - the list of attributes to approve. Enter the LDAP names of the attributes. Note: the display style of each attribute is taken from the object description (type and editor).

Acknowledge Update:

Updates the TS State account or group states as defined by the State attribute and the membership states as defined by DirX Identity.

Calculate VMID GUID:

GUID Attribute - the name of the attribute in which the generated GUID value is to be stored.

Calculate Siemens GUID:

Note: this service is only accessible within the Siemens internal network. See the corresponding service description for details.

- **GUID Attribute** - the name of the attribute in which the calculated GUID is stored.

Connection attributes:

- **URL** - the network address of the Web service
- **User ID** - the user identifier who calls the Web service; more precisely the application. The GUID module allows access only for registered applications.
- **Password** - the password for the user ID.
- **Requestor** - the user who is using the application to request the Web service.
- **Initial Role** - a number that describes the required data and validation base.

Attributes for error handling:

- **Retry Count** - the number of retries to be performed in the event of network error.
- **Interval** (in sec) - the interval before the next retry is performed.

The LDAP attributes **sn**, **givenname**, **mail** and **c** are used to calculate a new GUID. For other attributes, the corresponding LDAP attribute is not fixed. These attributes are configured in the **Additional Parameters** tab (the names are case-sensitive, enter the following name into the name column and the LDAP attribute name from which the value is to be taken into the value column):

- **birthname** - the attribute used for the value of the **birthname** parameter.
- **birthdate** - the attribute used for the value of the **birthdate** parameter. If the value ends with **Z**, the value is automatically converted from Zulu time to the required date format *dd**.***MM***.**yyyy*. Otherwise, the value is passed on as it is.
- **birthplace** - the attribute used for the value of the **birthplace** parameter.
- **nameprefix** - the attribute used for the value of the **nameprefix** parameter.
- **namesuffix** - the attribute used for the value of the **namesuffix** parameter.

An additional configuration parameter is:

- **dosearch** - whether (default) or not a search should be performed. Set this field to 0 if you do not want a search to be performed after the method getNewGID() fails.

Check SoD:

(has no parameters)

Enter Attributes:

User Base - the location at which to create the object.

Attributes - the list of attributes required to create the object. Available columns are:

Name - the LDAP name of the attribute.

Description - the display name of the attribute for the Web Center.

Mandatory - the attribute must be populated by the participant. Mandatory attributes are highlighted in red.

Map Attributes:

Defines additional attributes or recalculates existing attributes (in this workflow order).

Attributes - a list of name / value pairs. Available columns are:

Name - the LDAP name of the attribute.

Value - the expression that defines the value. You can use it to set constants or to combine

attributes to populate another attribute or to recalculate an attribute.
Examples:
employeeType = Customer
cn = ${sn} + " " + ${givenName}

Notification:

From - the e-mail address of the sender.

To - the e-mail address of the receiver.

CC - the e-mail addresses to copy this e-mail.

BCC - the blind copy emails addresses to copy this email.

Subject - the title (subject) of the email.

Language - the e-mail is sent in the defined language. You can either set a fixed value like **en** (choose from the drop-down list) or a variable value, for example ${workflow.subject.preferredLanguage} which defines the language value dynamically based on the **preferredLanguage** attribute of the subject.

Body - the body of the e-mail.

All fields can contain variables that are substituted at runtime. See the section "Request Workflows" in the *DirX Identity Application Development Guide* for more information.

Request Privileges:

Request Roles - lets the participant assign roles.

Request Permissions - lets the participant assign permissions.

Request Groups - lets the participant assign groups.

SubjectFromEntry:

This activity type is used for the first activity in the persona and functional user create workflows, **Persona from User** / **Functional User from User**. It creates default values from the persona's owner or the functional user's sponsor and the new object's parent DN.

The activity is configured with the following parameters:

Name of Object Description - the name of the object description to be used to create the object template from the responsible person; for example, PersonaFromUser or FunctionalUserFromUser. If left blank, the name of this object description is derived from the related user and the destination type (see the example below).

Parent folder for subject - the default parent folder for the subject. If left empty, the responsible user's parent folder is used.

Context attribute for associated entry - the name of the worklow context attribute that contains the DN of the associated entry (sponsor or owner) from which to create the

default values. Web Center populates the attribute associatedEntry with this information, so this parameter normally should not be changed.

Context attribute for destination OD name - the name of the worklow context attribute that contains the object description name of the entry to be created. Web Center populates the attribute destinationType with this information, so this parameter normally should not be changed.

Example:

If a new persona is created in Web Center, destinationType contains dxrPersona. The SubjectFromEntry activity uses the createFrom tag in the template's object description to determine the object description to be used to create the default values from the template object. This action results in PersonaFromUser, since the createFrom tags in the UserCommon.xml object description are

<createFrom destinationType="dxrPersona" objectDescriptionName="PersonaFromUser"/>

<createFrom destinationType="dxrFunctionalUser"
objectDescriptionName="FunctionalUserFromUser"/>

With this mechanism, the same workflows can be used to create personas from users or personas and to create functional users from users or personas.

Related Topics

Activity - General
Activity - Activity
Activity - Participants
Activity - Participants Filter and Constraints
Activity - Notifications
Activity - Escalations
Workflow - General
Workflow - Workflow
Workflow - When Applicable

Notifications

A.7.2.7. Activity - Participants

Use this tab to define participants and the procedure for how they interact.

Note that changes to the workflow definition are only updated in the Java-based Identity Server if either the server is restarted or a **Load IdS-J Configuration** is performed (from the context menu at the workflow entry).

First, select the calculation method:

static - a static list of participants or a static list of dynamic participant definitions.
A static list is useful when a fixed number of persons are responsible for approval (for example, specific and well-defined persons from the IT organization).
You can also define the initiator of the workflow as a participant (select from the drop-

down menu).

+ You can also set dynamic participant definitions here. This feature uses the same definitions as the ones used for dynamic mail text creation. See the chapter "Using Variable Substitution" in the *DirX Identity Application Development Guide* for more information. Here are some samples:
${workflow.subject.manager} - retrieves the manager(s) of the subject (uses the manager link to look up the manager DNs)
${workflow.subject.owner} - retrieves all owners of the subject (uses the owner link to look up the owner DNs)
${workflow.subject.manager.manager} - retrieves the manager of the subject's manager (useful for escalation)
${workflow.subject.dxrLocationLink.manager} - retrieves the manager of the location to which the subject is assigned
${workflow.previousParticipants.manager} - retrieves the manager(s) of the previous activity during an escalation
Use multiple lines to retrieve users via different links (for example, ${workflow.subject.manager} and ${workflow.subject.secretary}).
Make sure that the expressions represent objects of type User. Note also that this feature works for single-value and multi-value properties.

+ Additional parameters for the **static** definition are:

Size limit - the maximum number of participants. Use this field to restrict the number of participants. The value **0** indicates that there is no limit.

groups - the groups whose members define the participants. If the group holds members of type SvcTSAccount, the users identified by the dxrUserLink attribute are returned; otherwise, the members of type SvcUser are returned. Use this method, for example, for a hotline with a varying number of persons that are defined via roles. Additional parameters for the **groups** definition include:

Size limit - the maximum number of participants. Use this field to restrict the number of participants. The value **0** indicates that there is no limit.

filter - the filter to be used to calculate the participants. In this case, the list of participants is defined by all persons that match the filter criteria. Use this method, for example, for all persons from an organization. The filter parameters are:

Search base - the node at which to start the search.

Search scope - the scope of the search. Valid parameters are:

- Base Object - the search is done on the search base only.
- One Level - the search is done on the first level below the search base.
- Subtree - the search is done on the entire subtree below the search base.

Search filter - the LDAP filter condition. Use the filter editor to define it.

Size limit - the maximum number of retrieved results. Use this field to restrict the number

of participants. The value **0** indicates that there is no limit.

access Policy - the access policy to be used to calculate the participants.

Type - the view from which to calculate participants for a user-to-privilege assignment

- User - calculate participants from the user view.
- Privilege - calculate participants from the privilege view.

Size Limit - the maximum number of retrieved results. Use this field to restrict the number of participants. The value **0** indicates that there is no limit.

Note: for details about configuration with access policies, see the section "Policies for Approvals" in the chapter "Managing Policies -> Delegated Administration". For a sample workflow that uses this technology, see also the subsection "4-Eye Approval" in the section "Understanding Assignment Workflows" in the chapter "Using Request Workflows" in the *DirX Identity Application Development Guide*.

class - the customer-specific participant retrieval method. For information on writing your own Java-based participant retrieval method, see the chapter "Implementing a Java Class for Finding Participants" in the *DirX Identity Application Development Guide* or the use case document about "Java Programming". Additional parameters for **class** definition include:

Class - name of the user defined Java class

Parameters - parameters that are passed as a Java "Map" to the user-defined Java class. The parameters are defined as type / value pairs.

Size limit - restricts the number of participants. This value is important because it allows you to restrict the maximum number of participants. The value **0** implies that there is no limit.

Next, define the operational parameters:

Condition - the condition under which the approval process is finished:

- **All-must-succeed** - all participants must accept: for a successful approval with status "accepted", all participants must accept. If one participant rejects, the result is "rejected' and the entire approval process is aborted (the other participants are no longer asked to decide about approval).
- **Only-one-may-decide** - only one participant must decide (either accepts or rejects), after which the approval process is aborted (the other participants are no longer asked to decide about approval).

Execution - the type of execution:

- Sequential - the approvals are executed in sequential order.
- Parallel - the approvals are executed in parallel order.

Reduce Runtime Activities - whether (checked) or not (unchecked) the workflow engine

uses a single activity for all participants. By default, the workflow engine creates one activity for each participant in addition to the master activity. Setting this flag forces the workflow engine to use a single activity (in addition to the master activity) for all participants. This configuration can significantly reduce the amount of activities in the Java-based Identity Server and thus enhance performance.

Related Topics

Activity - General
Activity - Activity
Activity - Parameters
Activity - Participants Filter and Constraints
Activity - Notifications
Activity - Escalations
Workflow - General
Workflow - Workflow
Workflow - When Applicable

Notifications

A.7.2.8. Activity - Participants Filter and Constraints

Use this tab to define filters and constraints. Filters are evaluated before the constraint operation is performed.

Note that changes to the workflow definition are only updated in the Java-based Identity Server if either the server is restarted or a **Load IdS-J Configuration** operation is performed (from the context menu at the workflow entry).

After the filter operation, you can define a constraint operation in Java. For example, you can define a minimum number of participants. If the check fails, manual repair of the workflow is necessary.

Filter Class - the filter to be used on the calculated participants list. For example, you can remove specific persons or restrict the number of participants. Predefined Constraints classes are:

com.siemens.idm.requestworkflow.participant.ParticipantFilterInitiator - filters the initiator of the workflow from the list. You must ensure that there is more than one participant in the list if you use the Constraints Class to check for zero participants.

Constraints Class - the (Java) constraint operation to be used; for example, you can check a minimum number of participants. If this fails, manual repair of the workflow is necessary. Predefined Constraints classes are:

com.siemens.idm.requestworkflow.participant.ParticipantConstraintMinOne - checks that the number of participants is not zero.

See the section "Understanding Request Workflow States" in chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Reference* for more information.

Related Topics

Activity - General
Activity - Activity
Activity - Parameters
Activity - Participants
Activity - Notifications
Activity - Escalations
Workflow - General
Workflow - Workflow
Workflow - When Applicable

Notifications

A.7.2.9. Activity - Notifications

Use this tab to define the notifications to be used for an activity. The following types are available:

Notify before - sends a notification before the main step of the activity is performed. This is useful to inform participants before the approval step.

Notify on error - sends a notification if an error occurred while the main step of the activity is performed.

Notify after - sends a notification after the main step execution.

Click the button behind the text field to create a new notification (the notification wizard opens) or view or edit an existing notification.

Click the delete button behind the text field to remove a notification.

Note that changes to the workflow definition are only updated in the Java-based Identity Server if the server is restarted or a **Load IdS-J Configuration** operation is performed (from the context menu at the workflow entry).

Related Topics

Activity - General
Activity - Activity
Activity - Parameters
Activity - Participants
Activity - Participants Filter and Constraints
Activity - Escalations
Workflow - General
Workflow - Workflow
Workflow - When Applicable

Notifications

A.7.2.10. Activity - Escalations

Use this tab to define a ordered list of escalation steps. For example, you can define the manager of a user as the first step and the secretary of the manager as the second step.

You can edit the escalation table as follows:

- Use  to create a new escalation line. A wizard opens. Define a name, a description, the timeout, the participants and the operation parameters.
- Display an existing one with .
- Move lines up and down with  and  to define the sequence of escalations.
- Delete a line with .

Note that changes to the workflow definition are only updated in the Java-based Identity Server if the server is restarted or a **Load IDS-J Configuration** operation is performed (from the context menu at the workflow entry).

Related Topics

Activity - General
Activity - Activity
Activity - Parameters
Activity - Participants
Activity - Participants Filter and Constraints
Activity - Notifications
Workflow - General
Workflow - Workflow
Workflow - When Applicable

Notifications

A.7.2.11. Notifications

Use this tab to view and manage notification objects. Notifications properties include:

General

Name - the internal name of the notification object.

JobType - the job type (currently, only mailNotification is available).

Separate mails - whether (checked) or not separate mail messages are sent to each of the mail addresses in the **To** field. If you set this flag, make sure that the mail text is configured accordingly.

Userhook - the user hook that permits modifying e-mail properties like to, cc, subject, and body before sending the e-mail. The userhook class must implement the interface **com.siemens.idm.api.notification.EmailUserhook**. The method parameter **e-mailContext** allows accessing the workflow-related data like subject, resources, workflow name, activity name, and so on. It must be deployed to the folder **${DIRXIDENTITY_INST_PATH}/ids-j-*/confdb/common/lib**.

Operations after notification

(currently not used)

Mail parameters

From - the e-mail address of the sender.

To - the e-mail address of the receiver.

CC - the e-mail addresses to copy this e-mail.

BCC - the blind copy e-mails addresses to copy this e-mail.

Subject - the title (subject) of the e-mail.

Language - the e-mail is sent in the defined language. You can either set a fixed value like **en** (choose from the drop-down list) or a variable value; for example, ${workflow.subject.preferredLanguage} which defines the language value dynamically based on the **preferredLanguage** attribute of the subject.

Body - the body of the e-mail. To send a notification in HTML format, the resolved body must start with a **<!DOCTYPE HTML** string, for example <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">. The resolved body must form a valid HTML document. HTML tags can be then used directly in the mail body as well as in nationalization texts.

All fields can contain variables that are substituted at runtime. See the section "Request Workflows" in the *DirX Identity Application Development Guide* for more information.

Related Topics

Activity - Activity
Activity - Parameters
Activity - Participants
Activity - Participants Filter and Constraints
Activity - Notifications
Activity - Escalations
Workflow - General
Workflow - Workflow
Workflow - When Applicable

Activity Editor

A.7.2.12. SendMail

Use this tab to define the system send mail workflow. Properties include:

General

Name - the name of the workflow object.

Description - the description of the object.

Control

Active - whether (checked) or not (unchecked) the workflow is active.

Timeout - the time in seconds at which the workflow times out.

Number of retries - the number of retries the server will perform.

Wait before retry - the time in seconds between retries.

SMTP-service - the link to the SMTP service object.

A.7.2.13. Send Text Message

Use this tab to define the system Send Text Message workflow. Properties include:

General

Name - the name of the workflow object.

Description - the description of the object.

Control

Active - whether (checked) or not (unchecked) the workflow is active.

Timeout - the time in seconds at which the workflow times out.

Number of retries - the number of retries the server will perform.

Wait before retry - the time in seconds between retries.

SMS-Gateway - the link to the SMS Gateway service object.

Note: The Send Text Message workflow can only resolve nationalization definitions from the central Nationalization folder in the Domain Configuration.

A.7.3. Monitor

A.7.3.1. Request Workflow Instance

The instance of a running or completed workflow. It contains all activities as sub-entries. The workflow instance is preceded by an icon. A clock icon shows running workflows, a red cross shows failed workflows, the workflow icon shows successful workflows.

The property items displayed include:

Structure

Structure - the graphical representation of the workflow instance. It shows all activities (large objects) and the structural items of the workflow (small objects).

Workflow

Name - the displayed name of the workflow instance.

Description - the description of this workflow instance.

Workflow Type - the workflow type.

Operation Type - the type of operation.

Subject Type - the type of object this workflow is for.

Initiator - the user that initiated this workflow.

Subject - the link to the subject (for example, a user).

Resources - the links to the related resources (if any).

Status Information

State - the state of the workflow. See the status handling section for details.

Start date - the time at which this workflow instance was started.

Expiration date - the time at which this workflow instance expires. This value is calculated by start time plus workflow timeout.

End date - the time at which this workflow instance finished. This field is empty for running workflows.

Status Expiration - the lifetime of this workflow instance. Set up a cleanup consistency workflow to delete these instances from time to time.

Error messages - a list of problems that occurred during request workflow processing.

Object

This tab displays the set of attributes to be modified for this object (the subject). It provides the following columns:

Due date - the date on which this change will be performed. If this field is empty, the change is performed immediately after approval.

Attribute - the name of the attribute.

Old value - the old value.

New value - the new value.

Note that the operation is indirectly defined by the value combination:

Add - the old value is empty, new value is present.

Modify - both values are present.

Delete - the old value is present, new value is empty.

Assignments

This tab displays the assignments of resources to this subject that are to be modified. It provides the following columns:

Op - the operation: '+' stands for add, '-' for remove and 'o' for modify.

Due date - the date on which this change will be performed. If empty the change is performed immediately after approval.

Type - the type of resource (privilege).

Name - the name of the resource.

Attribute - the attribute name.

Old values - the old value of a resource attribute.

New values - the new value of a resource attribute.

Assigned by - the type of assignment (manual or by rule).

Child Workflows

This tab lists all started child workflows. Click a line and then click the icon on the right side of the table to navigate to this workflow instance. Use the arrow button of the Manager to return to the parent workflow. These columns exist:

Instance ID - the instance ID of the started child workflow (this is the cn).

Name - the display name of the child workflow.

Path - the relative path of the child workflow.

State - the state (only visible after workflow completion).

Application State - the application state (only visible after workflow completion).

Track Changes

A list of all provisioning actions that are necessary to complete the workflow. The following columns are available:

Name - the name of the workflow
This workflow - indicates changes from this workflow instance._
name_ - the display name of the sub workflow.

No - the line number per workflow. For example, if a workflow produces three changes, these are numbered from 1 to 3.

Op - the type of request ('+' for Addition, 'o' for Modification and '-' for Deletion).

Type - the type of object (Group or Account).

State - the target system state of the object.

Object - the DN of the object (account or group).

Member - the member attribute of the account for group membership operations.

Conflict - the conflicts that exist between different lines, for example, a parallel add and delete of a membership in a ticket.

For LDAP searches, most of this information is available as searchable LDAP attributes.

Related Topics

Monitor
Workflows
Activity Instance

A.7.3.2. Request Workflow Activity Instance

An activity instance of a running or completed workflow. It is preceded by an icon. A clock icon shows running activities, a yellow hub shows not yet active activities, a green hub shows successfully completed activities and a red cross shows failed activities.

The property items shown here include:

General

Name - the displayed name of this activity instance.

Description - the description for this activity instance.

Operation type - the operation type. Valid types are:

- **automatic** - an activity to be executed automatically by DirX Identity.
- **notification** - an activity to notify persons.
- **people** - an activity to be performed by one or more participants.

Activity Type - the action that the activity performs (for example, applyChange). Note: Customers can extend this list.

Start Condition - the start condition that triggered the activity.

Status Information

State - the state of the activity. Valid values are:

- FAILED - an error occurred.

- SUCCEEDED - the activity performed successfully.

Application state - the state of the application. The values are individual to each activity. Common values are:

- ACCEPTED - a participant accepted the approval request.
- REJECTED - a participant rejected the approval request.

Reason - the reason that the participant entered into this field.

Participant - the participants of this activity before the approval. Shows the participants that approved after approval.

Potential participant - is empty before approval. Shows all participants that did not approve after approval.

Start date - the time at which this activity instance was started.

Expiration date - time at which this activity expires.

End date - the time at which this activity instance finished. This field is empty for running activities.

Escalation Level - the current level of escalation for this activity.

Retry Limit - the number of retries that have been performed up to now.

Related Topics

- Monitor
- Workflows

Workflow Instance

A.8. Target Systems View

The **Target Systems** view shows a list all currently configured target systems, their configuration data and properties.This information includes the list of all accounts valid for accessing the respective target system, the target system-specific descriptions of objects, the description of specific property pages, proposal lists, obligations, JavaScript files and reports.Additional query folders may exist to filter out subsets of this data.

For each selected item in the tree, the corresponding property pages are shown on the right-hand side.Use **Edit** to modify property values.Note that saving the properties of an entry will result in another resolution of its relationships to other objects.

Related Topics

- Target System
- Account and Group

- JavaScript File Object / Property Page Description
- Proposal List
- Report

A.8.1. Target Systems

This tab shows the general properties for the target system folder. The target system folder contains the entries of all currently configured target systems. It may also contain one or more query folders to filter out a subset of target systems that match the given criteria.

Note: This folder is also displayed as the **Groups** folder in the **Privileges** view.

The property items shown here include:

Name - the displayed name of the target system folder.

Description - the description for the target system folder.

Related Topics

Target System

A.8.1.1. Accounts and Groups - General

This folder contains all accounts and groups that belong to a particular target system. It may also contain query folders to filter subsets of accounts and groups matching the respective criteria.

The property items shown here include:

Name - the displayed name of the accounts and groups folder.

Description - the description for the accounts and groups folder.

Alternatively, you can work with separate Accounts and Groups folders.

Related Topics

- Accounts and Groups - General
- Accounts and Groups - Attributes for the Connected System

A.8.1.2. Accounts and Groups - Attributes for the Connected System

Use this tab to define the object classes and attributes for superior node creation within Java-based workflows.

The property items shown here include:

Object Classes in Connected System - defines the object classes in the corresponding entry of the connected system.

Attributes in Connected System - defines default values for other mandatory attributes of the corresponding entry.

For more information, see "Using the Target System (Provisioning) Workflows" -> "Understanding the Java-based Target System Workflows" -> "General" -> "Creation of Superior Folders" in the *DirX Identity Application Development Guide*.

A.8.1.3. Target System - General Properties

A target system object holds all configuration data for a resource pool like a network system or a host computer running under a particular operating system (UNIX, Windows, MVS, ...) providing several services (mail, file, printing, applications for various tasks).

The data stored for a target system includes all accounts and groups that belong to that target system and the configuration data, which consists of object and property page descriptions, JavaScript files, reports and specific proposal lists.

Use this tab to view and manage the general properties of the target system object. These properties include:

Identification Properties

Name - (mandatory) the name of the target system as it is used for display and identification purposes throughout the DirX Identity Provisioning system. Once this name is set during new target system addition, it cannot be changed after the initial save operation.

Description - (optional) a description of the target system. This information is often useful when target system entries are listed in tables just with their names but also with a description column to better identify them.

Type - (mandatory) the target system type. This value is automatically set during the creation operation and cannot be changed afterwards.

Administration

Local admin - (optional) the distinguished name of the user who is permitted to administer the target system.

Last validation - the date on which the target system was last validated. This field is automatically set by the target system validation workflows.
Note: If this field is empty, the validation workflow runs in **InitialLoad** mode; if it is populated, it runs in **Validation** mode. In Validation mode, differences between the connected system and the target system are evaluated and written to the corresponding fields of an object (for example, the ToDo field).

Relationships

Peer Target System - the peer target system. A peer target system exists when there are two target systems which each contain a part of the master information. For example, there is a NT account that is associated with an Exchange 5.5 account.

Connected Directory - the corresponding connected directory that stores the data on Connectivity side; for example, the data required to run workflows.

Portal External Application - a list of names. Each name is used as a separate external application for the account. This field is mainly used by the Dashboard portal application.

Password Policy

Default Password Policy - the default password policy for this target system. You can select a global password policy (use the **...** button) or a target system specific policy (use the **TS...** button).

Related Topics

- Advanced Properties
- Timing Properties
- Workflow Configuration
- Connector Configuration
- Options
- Account
- Group - General Properties
- JavaScript File
- Object / Property Page Description
- Proposal List
- Report

A.8.1.4. Target System - Advanced Properties

Use this tab to view and manage the advanced properties of a target system. These properties are mainly used for configurations where accounts are not available or are referenced by an attribute that is different from the common name. All properties are optional and include:

Match Properties

- **Type** - (mandatory) the target system type is automatically set during the creation operation and cannot be changed afterwards.
- **Cluster** or **Forest Name** - the cluster or forest name, for clustered target systems. For example, in Windows, several domains belong to a forest (the cluster). The event manager can use this property to identify the account of a password change event correctly. In this case, it uses the cluster and domain attributes of the password change event to identify the target system for the account search.
- **Domain** - the domain name of the target system (for example, the Windows domain name). The domain is used when searching for a suitable Java-based workflow to process events raised by modification of target system objects (accounts and groups). The target system domain can also be computed dynamically: if a special format of

value is recognized, the domain is computed based on the given input. The input value must match the following JRE expression:
\\$\\{([a-z]+)\\(\"([^\"]*)\",\"([^\"]*)\"\\)\\}.
The notation is **${*function*("*arg*","*default*")}***, where *function* can be:

- *property* - the value of the given property name of the modified object
- *class* - a Java class that implements **siemens.dxr.service.core.api.ResourceExtender** interface. See the Java documentation (javadoc) for details.

- Example 1: When an account is modified and the TS domain is set to **${property("myDomain", "default-domain")}**, the computed value will be the value of the myDomain property of the given account or "default-domain" if the account does not have such a property.
- Example 2: When an account is modified and the TS domain is **${class("my.custom.Extender", "default-domain")}**, the computed value will be the value that is returned by the Java method **extendResource(account, "default-domain")** of the class **my.custom.Extender**. The class must be properly deployed in the WebCenter and/or the DirX Identity Manager.

Assignment Properties

- **Assignment states** - whether (checked, the default) or not (unchecked) it is necessary to have multiple account-to-group or user-to-group assignment states. When checked, DirX Identity uses the member attributes of the groups. When unchecked, DirX Identity uses only the attribute type that is specified in **Member property**. *
  Note:* If Assignment states is set to false, the existing states ENABLED and ADD are merged to ENABLED. Assignments in state IMPORTED or IGNORE are reported with warnings. All other states are deleted.
- **Reference group from account** - whether (checked) or not (unchecked) group memberships are stored in the accounts. This flag is set individually for each target system.*
  Notes:*
  - Changing this property results in a warning that a migration procedure will be performed, in which all account-group memberships are moved from groups to accounts or vice versa. If you have a large number of accounts, the migration procedure can run for a long time!
  - Extension groups are not supported when this property is set. When you check this field (change it to true), the migration procedure writes the reference into the master group. The extension groups can be deleted afterwards.
  - When you uncheck this field (change it to false), the migration procedure creates the necessary extension groups up to the defined limit.
  - For nested groups, the references point from root to leaf (as is the case when this property is set to false). During a migration, the assignment states, the member attributes or the referenced property are updated and the reference direction is not changed independent of how this property is set.
  - The Tcl-based workflows can only handle the default configuration for a specific target system. If you switch to the opposite flag position, you must change the Tcl-based workflows accordingly.

- Java-based workflows can handle both flag positions automatically. If you switch from the default to the opposite flag position, you also need to adapt the workflows regarding the channel sequence numbers and the association of the member channel to the appropriate group or account channel. For a detailed description, see the section “Cross-Memberships” in the *DirX Identity Application Development Guide*.

- **Referenced object type** - the name of the object to which group memberships refer (the object description name, sometimes called the "odname"). Since the object must be present in the operation context, only the following values are allowed:
- **SvcTsAccount -** for the account.
- **dxrUser -** for the user.

  The default value is SvcTsAccount.



Changing this property results in a warning that informs you that manual migration is necessary. If **Referenced object type** is set to dxrUser, the options **Assignment States** and **Reference Group from account** are not supported. A warning is displayed that the options are not consistent.

- **Member property** - the attribute type where group members are to be stored. This property is only evaluated when **Assignment states** is unchecked. The default value is uniqueMember.
- **Referenced property** - the attribute type of assigned objects to be used for group memberships. All attribute types (properties) listed in the corresponding object description are allowed. The default value is cn.
- **Source for referenced property** - the attribute of the referenced object which the real target system stores in the members attribute. This property is only of interest in cross-membership scenarios; that is, where DirX Identity stores the account-group memberships in the account, whereas the target system stores them in the group. For example, in LDAP servers, this is the DN; in IBM Notes, it is the full name. If this property is left empty, the system assumes the identifier (for example, the DN). The join engine uses this user attribute value to find the groups to which the user belongs (user is a member) when it synchronizes the target system user-group memberships into the Identity Store.

Synchronization Properties

- **Enable Realtime Provisioning** - whether target system-specific changes on account and group objects are propagated immediately to the target system (checked) or stored in the Java-based Server’s dead letter queue (unchecked). A real-time workflow must be configured correctly in order to set this flag.
- **Disable Password Sync** - whether (unchecked) or not (checked) DirX Identity synchronizes password changes to the target system (checked). When this flag is clear, DirX Identity creates requests for password changes in this target system. Make sure you have a corresponding password change workflow in place, or the password requests accumulate endlessly.

Password Management

- **Master** - whether (checked) or not (unchecked) external authentication is enabled. The target system is considered to be the master for the users. Password synchronization must be enabled (**Disable Password Sync** is unchecked). If there is more than one target system that is marked as a master for users, external authentication succeeds if at least one master target system authenticates the user.
- **Authentication class** - the class used for external authentication. The class must implement the interface **com.siemens.webMgr.identityAPI.authentication.ext.Authentication** and must be deployed in the **WEB-INF/lib** directory of Web Center. DirX Identity provides the following built-in classes for LDAP and Active Directory authentication that can be used out of the box:
  - * **com.siemens.webMgr.identityAPI.authentication.ext.AdAuthentication**
  - * com.siemens.webMgr.identityAPI.authentication.ext.LdapAuthentication
- **Connection string** - the connection string (domain) for external authentication. For the built-in classes LDAP and Active Directory, the syntax is [*protocol**://]***host***[:*port*][/*dn*]. For example, **localhost:389**, **ldap://host.domain.com**, **ad.my-company.com** or **ldaps://ad.my.company.com:636**. The default protocol is **ldap**, the default ports are **389** for unsecured connections and **636** for SSL connections. For other connected systems, the custom authentication class defines the syntax..
- **Login attribute name** - contains the DirX Identity TS account attribute name that holds the login in connected system with respect to the authentication class. This overrides the defaults in the implementation of authentication class. The default for AD and LDAP is **dxrPrimaryKey**.
- **Login form attribute name** - contains the name of the DirX Identity account attribute that holds the user identification entered into the login form. This overrides the defaults in the implementation of the authentication class. The default for AD is **dxrName**. The default for LDAP is **cn**.

Group Handling

- **XXL Groups** - whether (checked) or not (unchecked) the algorithm for privilege resolution and policy execution for large groups (> 100 000 members) is optimized. When checked:
  - The groups of this target system are read only once at the start of the resolution process. For further processing, they are held in a cache.
    Reason: Reading large groups is a very time consuming operation at any directory server.
  - All groups of the target system are read with a single search (simple paged search with pagesize = 1).
    Reason: Because of the simple search filter, the read operation is faster. Main memory consumption at the directory server side is reduced



If **Reference Group from Account** is selected, **XXL groups** is ignored.

- **Ignore nested groups** - whether (checked) or not (unchecked) nested groups are ignored. Check this property if you do not need nested groups. Otherwise performance can be very poor in domains with many users.

Privilege Resolution

- **Smooth Account Creation** - whether or not smooth account creation is activated. Assigning a privilege to a user results in account creation if an account in the respective target system is not yet created. Depending on the algorithm defined for the account, name errors can occur. If an account with the same name already exists, and the algorithm is not able to calculate an alternative name, this is reported and account creation is aborted. You can check this flag to avoid this situation. The software then takes the existing account.

Related Topics

General Properties
Timing Properties
Workflow Configuration
Connector Configuration
Options

A.8.1.5. Target System - Timing Properties

Use this tab to view and manage the timing properties of a target system. These properties include:

Maximum time

to delete an object - the lifetime (number of days) after which DirX Identity Provisioning will automatically delete a user, account, or group object in the TBDEL state from the Identity Store (Provisioning Configuration). Specifying 0 for this parameter means the object is deleted today, 1 means tomorrow and so on. Use a very high value (maximum is 999,999) to simulate an infinite lifetime.
Note: If this field is left blank, the corresponding value from the Timing tab of the Domain object applies.
For more information on how these parameters control the user and account life-cycle, see the section "User and Account Life-Cycle" in the "Managing States" chapter.

to disable an object - the lifetime (number of days) after which DirX Identity Provisioning will automatically delete a disabled account (set its state to TBDEL and set EndDate to the current day plus the value of MaxTime2Delete). Specifying 0 for this parameter means the object is set to TBDEL state today, 1 means tomorrow and so on. Use a very high value (maximum is 999,999) to simulate an infinite lifetime.
Note: If this field is left blank, the corresponding value from the Timing tab of the Domain object applies.
For more information on how these parameters control the user and account life-cycle, see the section "User and Account Life-Cycle" in the "Managing States" chapter.

Related Topics

General Properties
Advanced Properties
Workflow Configuration
Connector Configuration
Options

A.8.1.6. Target System - Workflow Configuration

Use this tab to view and manage the parameters used by Java-based workflows to connect to a target system. These properties allow you to set up one workflow that can synchronize several target systems in parallel. Properties include:

Host Parameters

Address - the IP Address or URL to access the target system.

Data Port - the port to use when **SSL Connection** is unchecked.

Secure Port - the port to use when **SSL Connection** is checked.

Bind Parameters

Bind Account - the account that is used by the workflow to access the target system.

Authentication Mode - the authentication mode. Supported values are:
user/password - user and password are sufficient to authenticate.
PKI - a certificate is required at the bind account to authenticate.

SSL Connection - whether (checked) or not (unchecked) to use a secure connection. Set **Secure Port** above consistently.

Path to Key Store File - the path to the key store file.

Key Store Alias - the alias that marks the certificate to use.

Path to Trust Store File - the path to the trust store file.

Related Topics

General Properties
Advanced Properties
Timing Properties
Connector Configuration
Options

A.8.1.7. Target System - Connector Configuration

Use this tab to set special parameters for the connector if they are individual for each target system. These parameters allow you to set up one workflow that can synchronize several target systems in parallel.

Use the multi-value string editor to set up name/value pairs for these parameters. They are

stored in the **dxmSpecificAttributes** LDAP attribute of the target system object. Use the space character as a separator.

Example:

parameter1 value1

Related Topics

General Properties
Advanced Properties
Timing Properties
Workflow Configuration
Options

A.8.1.8. Target System - Environment Properties

Use this tab to set special environment parameters for the workflow controller if they are individual for each target system. These parameters allow you to set up one workflow that can synchronize several target systems in parallel.

Use the multi-value string editor to set up name / value pairs for these parameters. They are stored in the **dxrEnvironmentProperties** LDAP attribute of the target system object. Use the space character as a separator.

Example:

parameter1 value1

Related Topics

General Properties
Advanced Properties
Timing Properties
Workflow Configuration
Options

A.8.1.9. Target System - Options

Use this tab to view and manage optional parameters for a target system. The properties listed here are target system-specific.

For more detailed information on these parameters, see the specific target system's documentation.

For a **Dashboard** target system:

- No specific information is required.

For an **Active Directory** target system (also referred to as Windows 2000 type):

- **Account Root in CS** - the location (distinguished name) where the accounts reside in

the connected system. It is used by the account object description's naming rules to calculate the **dxrPrimaryKey** attribute when the account is created in DirX Identity.

- **Group Root in CS** - the location (distinguished name) where the groups reside in the connected system. It is used by the group object description's naming rules to calculate the **dxrPrimaryKey** attribute when the group is created in DirX Identity.

For an **Exchange (extending Active Directory)** target system:

- **Base Mail Address** - the base part of the email address, which is extended to a valid user-specific email address by the mailbox-enabling obligation rules and then mapped by the workflow to the **mail** connected system attribute.
- **Base Legacy Exchange DN** - the base part of the legacy Exchange DN, which is extended by the mailbox-enabling obligation rules to a valid user-specific value and then mapped by the workflow to the **legacyExchangeDN** connected system attribute.
- **Exchange Home MTA** - the DN of the Message Transfer Agent Server which is mapped to the connected system attribute **homeMTA**. This property is ignored for Exchange version 2013 or higher; in these versions, the Message Transfer Agent is always located on the Message Database server (homeMDB).
- **Exchange Home MDB** - the DN of the Message Store Server which is mapped to the connected system attribute **homeMDB**.
- **Exchange Home Server** - the Exchange Server Name mapped to the connected system attribute **msExchHomeServerName**.

For a **Lync** (**extending Active Directory**) target system:

- **Lync Home Server** - the Skype for Business Server (formerly Lync Server) Name mapped to the target system attribute **msRTCSIP-PrimaryHomeServer**.
- **Lync Base Sip Address** - the base part of the sip address, which is extended by the lync-enabling obligation rules to a valid user-specific sip sign in address and mapped by the workflow to the connected system attribute **msRTCSIP-PrimaryUserAddress**.
- **Lync Base Sip Address** - the base part of the sip address, which is extended to a valid sip sign in address and mapped to the target system attribute msRTCSIP-PrimaryUserAddress.

For an **OpenICF Windows Local Accounts** target system:

- No specific information is required.

For a **Windows NT** target system:

- **Group Member Limit** - maximum number of accounts in one group (1000).

For an **LDAP** target system:

- **Account Root in CS** - the location (distinguished name) where the accounts reside in the connected system.
- **Group Root in CS** - the location (distinguished name) where the groups reside in the connected system.

For a **Notes** target system:

- **Account Root in CS** - the location (distinguished name) where the accounts reside in the connected system. Specify the structure element (like /o=atos) that corresponds to the one contained in the **cert.id** file that you use as one of your bind profiles. IBM Notes creates the full name of each user by appending this structure element to it and uses the full name inside the member attribute of groups. Hint: In DirX Identity, the cn of the account is also built by concatenating gn, sn and this structure element, because the member attributes of the groups in DirX Identity reference the account cn.

For an **ODBC** target system:

- No specific information is required.

For an **Office 365** target system:

- **Tenant Domain** - the name of your domain as configured in Office 365 admin center. This value is used as a suffix for the Office 365 **User Principal Name** when creating a new account.

For a **RACF** target system:

- **Account Owner** - the owner of the user's profile
- **Account Default Group** - the user's default group
- **Account Installation Data** - installation-defined data (comment)
- **Group Owner** - the owner of the group's profile
- **Superior Group** - the name of the group's superior group
- **Group Installation Data** - installation-defined data (comment)

For an **SAP ECC UM** target system:

- No specific information is required.

For an **SAP NetWeaver UM** target system:

- **Account Root in CS** - the location (distinguished name) where the accounts reside in the connected system (for example, "USER.CORP_LDAP.cn="). This value is used during account creation in DirX Identity to build the dxrPrimaryKey attribute of the account. The member attributes of groups in DirX Identity reference the dxrPrimaryKey attribute of accounts.
- **Group Root in CS** - the location (distinguished name) where the groups reside in the connected system (for example, "ROLE.UME_ROLE_PERSISTENCE.un:"). DirX Identity uses this value during group creation to build the dxrPrimaryKey attribute of the group.

For a **SiPass** target system:

- No specific information is required.

For a **UNIX-OpenICF** target system:

- **Group for Disabled Accounts** - the name of an existing UNIX group with limited access rights which can be used for disabled (locked) accounts.

For a **Google Apps** target system:

- **Domain Name** - the name of your company domain. This value is used as a suffix for the GoogleApps **User Email** when creating a new account and for the GoogleApps **Group Email** when creating a new group.

Related Topics

General Properties
Advanced Properties
Timing Properties
Workflow Configuration
Connector Configuration

A.8.1.10. Account - General Properties

The account object represents an account, usually an entry in the user database of an operating system, that is necessary to get access to system resources. An account can be managed completely by the DirX Identity Provisioning system or only locally in the respective target system.

Use this tab to view and manage the general properties of an account. The following properties are always shown:

Identification and Type Properties

Name (mandatory) - the name of the account as it is used for display and identification purposes throughout the DirX Identity Provisioning system. Once set during adding a new account entry, it can no longer be changed after the initial save operation.

Description (optional) - the description of the account. This field is often useful when role entries are listed in tables with just their names but also with a description column to better identify them.

Account type - the type of the account. The following types are available:

- **Personal** - represents a personal account of the user. This account type uses the **dxrUserLink** attribute to indicate the corresponding user.
- **Privileged** - represents a privileged account that does not belong to a specific user. Users can only use this account for some period of time. The **dxrUsedBy** attribute is used to indicate users that use this account.

Primary account - whether (checked) or not (unchecked) the account is the primary account. This flag is important if the user has more than one account in a target system. New access rights (that is, groups) are always assigned to the primary account. If a user owns more than one account in a target system, one of them must be flagged as primary. New access rights (groups) are assigned only to primary accounts.

Account specific properties

User (optional) - the distinguished name of the user to whom this account is assigned. The distinguished name can be entered directly in LDAP syntax or selected from the node chooser dialog by clicking ... Once an entry is assigned, use the properties button to view the properties of the user entry in a separate window. If you do not want to assign this account to a user, check **Manage only in target system**.

Surname - (optional) the last name of the respective user. This field is automatically updated when you exit the **User** field after entering a correct user distinguished name.

Given name - (optional) the first name of the respective user. This field is also automatically updated when you exit the **User** field after entering a correct user distinguished name.

Note that this property section can vary from target system to target system.

Related Topics

Folder
Target System Specific
Operational
Authentication
Member Of
Used By
Group

A.8.1.11. Account - *Target System Specific*

This tab shows target system-specific attributes of the account. Target system-specific account attributes per target system type are:

LDAP

PrimaryKey (DN in TS) -the distinguished name of the account in the target system.

For LDAP, additional properties shown are **Employee number**, **E-Mail**, **Postal address**, **Postal code**, **Location**, **Street** and **Room**. These properties correspond to the related attributes at the user.

Office 365

Primary Key (ID in TS) - the identifier of the account generated by Office 365.

Display Name - the display name attribute for Office 365.

Mail Nickname - the mail nickname attribute for Office 365.

User Principal Name - the user principal name attribute for Office 365.

Sku ID - the license identifier currently assigned to the Office 365 account allowing usage of Office 365 applications (service plans).

RACF

For the RACF attributes, see the RACF LDAP documentation.

Salesforce

PrimaryKey (Id in TS) - the value of “Id” attribute of the Salesforce user (or profile).

Alias - the value of the Alias attribute of the Salesforce user.

Nickname - the value of the CommunityNickname attribute of the Salesforce user.

User License Id - he value of the userLicenseId attribute of the Salesforce profile.

User Type - the value of the UserType attribute of the Salesforce profile.

For Salesforce users, additional attributes are displayed: **Country**, **Department**, **Employee number**, **E-Mail**, **Fax**, **Language**, **Location**, **Mobile**, **Organization**, **Postal code**, **Street**, **Title**. These attributes are in most cases directly derived from the corresponding attributes of the Salesforce user. For details, see the account channel mapping of the Salesforce realtime workflows.

SAP R3 UM (user management)

SAP User Name - the unique user account name in the SAP ECC system.

For SAP ECC, additional properties shown are **Employee number**, **E-Mail**, **Room**, **Street**, **Postal Code**, **Location** and **State**. These properties correspond to the related attributes at the user.

UNIX OpenICF

Account Name - the unique username in a UNIX system.

UID Number - the unique UID in a UNIX system.

Primary Group - the unique primary group name in a UNIX system assigned to the user account.

Home Directory - the path to the user home directory in a UNIX system.

Shell - the path to the user default UNIX shell.

Active Directory

Account Name - the unique user account name in the respective domain.

Primary key (DN in TS) - the account’s distinguished name in the target system.

ADS Path - the location of the Active Directory Service with the server and provider identification, for example, LDAP://MetahubServ2000/DC=de/DC=sni/DC=mch/DC=dirxmetahub2000/OU=TestUsers2.

ADS GUID - the global unique user identifier.

User Account Control - the account profile flags for access control.

User Principal Name - the user login name.

Mail address - the SMTP mail address.

Legacy Exchange DN - a distinguished name of the account used for synchronization purposes on different MS-Exchange domains and versions.

Home MTA - the home message transfer agent.

Home MDB - the home mailbox database.

Show in Addressbook - a list of all address books in which this account appears.

Home Server Name - the home MS-Exchange server.

Mail Nickname - a short display name for the mail account.

SIP User Address - the sip address used to sign in to Skype for Business.

For more detailed information on these properties, see the respective manuals and documentation for Microsoft Exchange.

Related Topics

Folder
General Properties
Operational
Authentication
Member Of
Used By
Group

A.8.1.12. Account - Operational

Use this tab to view and manage the attributes that are relevant for standard DirX Identity operation.

Status Fields

State - the status of the account. This may take one of the following values (for more information, see the section "Managing States"):

- **ENABLED** - the account shall exist, be created in the target system and be enabled.
- **DISABLED** - the account shall exist, be created in the target system and be disabled.
- **DELETED** - the account shall be deleted in the target system.
- **IMPORTED** - the account was created in the target system, but is not yet requested by DirX Identity Provisioning.

End date - the date on which the account will automatically be set to state **TBDEL** (when in state **DISABLED**) or really deleted in DirX Identity Provisioning (when in state **TBDEL**).

State in target system - the status of the account in the respective target system. This value may differ from the status of the account entry in DirX Identity Provisioning as long as the information is not synchronized. The field can take one of the following values (for more information, see the section "Managing States"):

- **ENABLED -** the account exists in the target system and is enabled.
- **DISABLED -** the account exists in the target system and is disabled.
- **DELETED** - the object has been deleted in the target system without DirX Identity Provisioning having requested it! This state is only set by the validation workflow along with a message in the 'dxrToDo' attribute reminding the administrator to inspect this situation.
- **NONE** - replaces the empty field when the object is created by the DirX Identity Manager or agent.

Manage only in target system - whether the account is only managed in the target system (checked) or also in the DirX Identity Provisioning system (unchecked).

Tasks Fields

To do - the list of manual working items for the administrator. DirX Identity Provisioning stores the results of consistency checks it cannot repair automatically. The items are overwritten on the next consistency check.

Error - the list of error messages that show the detailed reason for inconsistent states or **To do** entries. They are cleared automatically when a privilege resolution succeeds.

Related Topics

Folder
General Properties
Target System Specific
Authentication
Member Of
Used By
Group

A.8.1.13. Account - Authentication

Use this tab to view and manage the properties that are relevant for authentication with passwords or certificates.

Password Management

Password Changed Time (read-only) - the last time the password was changed.

Password Display Time (read-only) - the last time the password was displayed.

Password Expiration Notified (read-only) - the time at which the password expiration was notified to the relevant user.

Password Policy - the password policy for this account. You can select a global password policy (use the **...** button) or a target system specific policy (use the **TS...** button).

One-Time Password

This group of fields relates to logins via one-time password.

One-Time Password (read-only) - the usually short-lived one-time password.

End of Life - the time the one-time password expires.

Mobile - the account's mobile number. If left unspecified, one-time passwords will be sent to the mobile number of the user to which the account belongs.

Logins with Password

This group of fields relates to account logins with password.

Failure Count (read-only) - the number of failed login attempts. The number is reset on any successful login attempt.

Last Failure Time (read-only) - the time of the last failed login attempt. The time is cleared on any successful login attempt.

Locked until (read-only) - the time until which login attempts by the account are rejected.

Secondary Authentications

This group of fields relates to account logins via other mechanisms like challenge response or one-time password.

Failure Count (read-only) - the number of failed authentication attempts. The number is reset on any successful login attempt.

Last Failure Time (read-only) - the time of the last failed authentication attempt. The time is cleared on any successful login attempt.

Locked until (read-only) - the time until which authentication attempts by the account are rejected.

Related Topics

Folder
General Properties
Target System Specific
Operational
Member Of
Used By
Group

A.8.1.14. Account - Member Of

Use this tab to view the list of groups in which the account is currently a member.

For each group, the following properties are shown:

Group - the group's name as it is displayed throughout the DirX Identity Provisioning system.

Description - the description of the group.

Assignment State - the status of the account assignment. Possible values are:

- **ADD** - the assignment is requested by DirX Identity Provisioning and shall be created in the target system by the synchronize workflow.
- **DELETED** - DirX Identity Provisioning requests the assignment to be deleted in the target system by the synchronize workflow. When it is deleted in the target system, the workflow removes the assignment physically from the attribute "dxrMemberDel".
- **ENABLED** - the assignment is established both in DirX Identity Provisioning and in the target system: they are "in-sync".
- **IMPORTED** - the assignment was created in the target system, but not yet requested in DirX Identity Provisioning. DirX Identity Provisioning does not delete it (ignores it), but it may be an indicator to the administrator that the user has access rights not granted by his roles. DirX Identity Provisioning switches the state to **ENABLED** when the user is granted this access right. If it is revoked afterwards, the state switches to **DELETED** and not back to **IMPORTED**!
- **IGNORE** - the administrator has accepted the imported assignments. DirX Identity Provisioning switches the state to **ENABLED**, when the user is granted this access right. If it is revoked afterwards, the state switches to **DELETED** and not back to **IGNORE**!

Direct - whether the account is a direct member of this group (checked) or indirect via a nested group membership (unchecked). If an account is member of group B, which is member of group A, then the account is also member of group A. This is flagged by Direct = FALSE (unchecked).

You cannot use this tab to make direct assignment of groups. The list just shows the backward references obtained by adding a user to a group. To inspect the details of a particular group, click ... to the right of the respective table row.

Related Topics

Folder
General Properties
Target System Specific
Operational
Authentication
Used By
Group

A.8.1.15. Account - UsedBy

Use this tab to view the list of users that currently use the account.

Related Topics

Folder
General Properties
Target System Specific
Operational
Authentication
Member Of
Group

A.8.2. Configuration

The configuration folder contains all configuration script files necessary to set up a target system and its representation in the DirX Identity Manager. A configuration can consist of

- **JavaScript files** - They are used to support creation and representation of object attributes and relationships between them which may also vary from target system to target system.
- **Object descriptions** - These are XML documents describing object types which have properties varying from target system to target system. Examples of such objects are the account and the group object.
- **Obligations** - A set of rules that can be linked to any group object.
- **Proposal lists** - A proposal list is necessary for property page controls presenting object properties where the user can select an item out of a list of possible values. These lists are configured here. A popular example is the list of locations used to fill in the location property for user and group entries.
- **Reports** - The report configuration files are also XML documents. Reports are used to simply display an overview or the details of a (sub-)set of RBAM objects, or to store them as files for further processing. The report scripts contain information on filter parameters and representation of the found results.

This tab sheet shows the general properties of a configuration folder. The property items shown here are the following:

Name - The displayed name of the configuration folder.

Description - The description for the configuration folder.

Related Topics

- JavaScript
- Object/Property Page Description
- Obligation
- Proposal List

- Report

A.8.2.1. JavaScripts

Use this tab to view the general properties for the JavaScript folder. The JavaScripts folder contains all JavaScript files used for performing special tasks during operation in a target system. It can contain additional query folders for filtering out subsets of JavaScript files matching the given filter criteria.

The property items shown here include:

Name - the displayed name of the JavaScript folder.

Description - the description for the JavaScript folder.

Related Topics

JavaScript

A.8.2.1.1. JavaScript - General Properties

This tab shows the general properties of a JavaScript file object. A JavaScript file is used to perform special tasks during configuration and presentation of RBAM objects.

The property items shown here include:

Name - the displayed name of the JavaScript file.

Description - the description for the JavaScript file.

Related Topics

Content

A.8.2.1.2. JavaScript - Content

Use this tab to view and edit the content of a JavaScript file. Click **Edit** to make the text editable. For more information on the editor features, see the manual page for the text editor.

To export the content of file to an external file, click **Export...** and then enter a file name in file dialog displayed. To replace the current content with text from an external file, click **Import...** and then enter the name of the file in the file dialog displayed.

Related Topics

- Text Editor
- General Properties

A.8.2.2. Object Description

A.8.2.2.1. Object Description - General Properties

This tab shows the general properties of an object description object. An object description is an XML script file and is used to define an object type with regard to its properties and its function in the system.

The property items shown here include:

Name - the displayed name of the object description.

Description - the description for the object description.

Related Topics

Content

A.8.2.2.2. Object Description - Content

Use this tab to view and edit the content of the object description. Click **Edit** to make the text editable. For more information on the editor features, see the manual page for the text editor.

The object description file is an XML script. The XML tags used in the script differ from context to context. This means that in a pure object description, other tags are used than in a property page description and those differ from the ones used for report descriptions or formats.

To export the content of the object description to an external file, click **Export...** and then enter a file name in the file dialog displayed. To replace the current content with text from an external file, click **Import...** and then enter the name of the file in the file dialog displayed.

For detailed information on how to modify the contents of an object description file, see the *DirX Identity Customization Guide* also contained in the online help.

Related Topics

- Text Editor
- General Properties

A.8.2.3. Obligation

A.8.2.3.1. Obligation - General Properties

Obligations help to reduce administration workload. They hold operations and filters that are common to a set of privileges (usually groups of target systems). Use this tab to display the general properties of an obligation object.

The property items shown here include:

Name - the displayed name of the obligation object.

Description - the description of the obligation object.

Related Topics

Obligations

A.8.2.3.2. Obligation - Obligations

Obligations are a method for setting or resetting account attributes when an account becomes a member of a group or is removed from the group.

In some cases, access rights in a target system are not expressed by a simple group membership, but instead by the appropriate setting of account attributes. One example is the Active Directory account, which is mailbox-enabled by setting some attributes, such as the mail address, the Exchange home server name, and so on. In DirX Identity Provisioning, on the other hand, an access right in a target system is modelled by a group membership. Obligations help to keep group memberships and appropriate attribute settings consistent.

Obligations consist of the following items:

- OnAssignment operations, to be performed whenever an account becomes a member of a group.
- OnRevocation operations, to be performed whenever an account is removed from a group. These obligations serve to reset attribute values of the account to reflect the lost access rights associated with the group membership.
- Validation filters, which identify accounts with attribute-based privileges and enter them into the appropriate groups. The policy agent evaluates these validation filters when it processes validation rules with the operation type "Validate".

Obligations can be stored with the associated group, but different groups can share the same onAssignment and onRevocation operations. As a result, DirX Identity Provisioning supports their storage in a common obligation object. You create obligation objects in the subtree "cn=Obligations,cn=Configuration" of a target system. A DN link from the group to the obligation object directs DirX Identity Provisioning to evaluate the obligation operations in addition to the obligations stored with the group.

When you edit an obligation:

- If you share common obligations for more than one group, enter the link from the group to the obligation in **Obligation Link**.
- Enter the operations for new group members into the rows of the **OnAssignment** table and the operations for deleted group members into the rows of the **OnRevocation** table. The handling is the same:
- To configure an **operation**, you first select the destination object and the attribute type; for example, the mail attribute. Normally, you set the attributes of the account. Only when the target system does not need own accounts, this is the user object itself. This should be consistent with the settings for the target system: the "Referenced object type".

- For the attribute value to store, you may either enter a simple string or configure one or more naming rules. Selecting the string "<clear>" directs the target system synchronization workflow to delete the attribute values in the target system object. If you specify a number of naming rules for one attribute, DirX Identity Provisioning evaluates them beginning with the first one until it is able to generate a non-null value. This feature allows you to define alternatives for situations where you are not sure that a necessary attribute value is available. As an example: take the mail address from the user, if he has one specified. Otherwise generate a default one.
- If the destination attribute is multi-valued, you can use a special syntax (in the value column) to specify to add or delete a given value. The statement **<add>Mail_A@my-company.com** adds the value **Mail_A@my-company.com**. You can use the **<del>** prefix to delete a value; this prefix is available for onAssignment and OnRevocation. However, keep in mind that this technique only works when you have defined the destination attribute (for example, mail) as multi-valued. If you want to specify multiple values, you can repeat the **<add>** or **<del>** statement. The statements **<add>Mail_A1@my-company.com<add>Mail_A2@my-company.com** adds the two values **Mail_A1@my-company.com** and **Mail_A2@my-company.com**. If you want to specify a fixed number of values to be put to a specified attribute, you can prefix the **<add>**s with a **<clearall>** statement. For example, **<clearall><add>Mail_A1@my-company.com <add>Mail_A2@my-company.com** sets the two values **Mail_A1@my-company.com** and **Mail_A2@my-company.com**.
- In a naming rule, you can re-use values from existing attributes of known objects and generate a new value by combining them with other attribute values, fixed strings, counters and random numbers. You can select from the following items:

reference- to re-use an attribute value from the account object itself or another well-known object.

fixed - to use a fixed string.

counter - to generate a value between a minimum and a maximum. DirX Identity Provisioning uses it to generate unique values.

random - to generate a random number between a minimum and a maximum.

If you select **reference**, you must also select:

- The source object that holds the attribute you want to use. This object is often the account itself (select **SvcTSAccount**), the associated user (select **SvcUser**) or the target system object (select **SvcTS**). The default is the account itself.
- The attribute type which holds the reference DN, if you want to get the value from another account that is referenced from the current one. An example: an Exchange 5.5 mailbox usually references an associated Windows account. The reference is stored in the attribute **dxrToPeer**.
- The source attribute itself from where you want to use the value.
- (Optional) A range, to select only the portion of the property value from index0 through index1. An index value of 0 denotes the first character. The "$" character denotes the end of the string: "$-3:$" mean the last 3 characters of the string.

- Case conversion (optional) to upper or lower case. The default is no case conversion.
- The type of the value. Usually this is string. The alternative is taking the value as an integer.

If you select **fixed**, just enter the string value.

If you select **counter**, enter the minimum and the maximum integer number. DirX Identity Provisioning begins calculates the first attribute value beginning with the minimum. If it fails to generate a valid value, it takes the next value in the range and so forth until it reaches the maximum.

Selecting **random** allows you to create a random number within the range minimum to maximum.

To specify an **On Validation** filter, enter a search filter, a search scope and the requested attributes as follows:

- Provide the **Search filter** in standard LDAP notation. It is used to identify accounts whose attribute values match the access rights associated with this group in the target system. Consequently, this filter must include at a minimum the attribute values set in the OnAssignment operations.
- The **Search scope** usually will be **Subtree** to cover all the accounts of the "Accounts" and "Accounts and Groups" tree.
- Make sure that the list of requested **Attributes** contains the important operational attributes like cn, dxrState and dxrTSState. It’s just a matter of performance: if they are absent, DirX Identity Provisioning must re-read the account entries.

Use the following method to call a JavaScript when an obligation is invoked:

- Use a virtual property (starting with a $; for example, $trigger) in the onAssignment/onRevocation action.
- For the property **myProp** to be modified, define dependsOn="$trigger" in **TSAccount.xml**.
- Assign a JavaScript to **myProp**.

Now each time the onAssignment/onRevocation action is called, $trigger is changed, leading to a re-calculation of **myProp** by its assigned JavaScript.

Related Topics

General Properties

A.8.2.4. Proposal List

A.8.2.4.1. Proposal List - General Properties

A proposal list presents a list of possible values for an object. Proposal lists are used in combo boxes and text field elements, where users can select a value from a drop-down list displayed when they click the arrow button to the right of the text field. An example of a

proposal list is the list of locations used in a domain configuration.

Use this tab to view and manage the general properties of a proposal list. These properties include:

Name - the displayed name of the proposal list.

Description - the description for the proposal list.

Type - the type of proposal list. Available types include:

- String - builds the proposal list from a simple list of items.
- DN - builds the proposal list from directory entries.
- Dependent DN - builds the proposal list from a sequence of inter-dependent drop-down lists.

Type String

Proposed values - for type **String**, enter the item list here. To modify this list, use the multi-value editor. You can enter item values in three ways:
storedValue - the displayed value (to the user) and the stored value (in the directory) are the same.

storedValue;displayedValue - the displayed value is shown to the user, the stored value is stored in the directory and can be used by rules and policies.

*storedValue;displayedValue;***si=***n* - the displayed value is shown to the user, the stored value is stored in the directory and can be used by rules and policies and a sort index that defines the position in the list. Per default the values are sorted by the displayedValues. Defining a sort index puts the value to this position in the list.
Example:
AT;Austria
DE;Germany
GB;Great Britain;si=1
US;United States;si=0
Setting the index for US to 0 and for GB to 1 results in this list of selectable values:
United States
Great Britain
Austria
Germany
If your company is mainly located in US and GB, these are the first selectable values in the list and most people can easily select them.

Type DN

Static proposals - additional DN static values. You can also enter an asterisk (*) in this field as an additional value.

Dynamic filter - the search base, the search filter and the search scope for a filter. You can define multiple filters. Use the **+** and **-** buttons to add or remove filter definitions.

Display attribute - the attribute that is displayed to the user as a list (sets the displayedValue).

Storage attribute - the attribute that is stored in the directory and then used by rules and policies (sets the storedValue).

Link attribute - this attribute that holds DN links to other objects that contain proposal list data. The proposal list data are read from the linked objects rather than from the objects returned by the search.

Type Dependent DN

Dynamic filter - the search base, the search filter and the search scope for a filter. You can define multiple filters here. Use the **+** and **-** buttons to add or remove filter definitions.

Display attribute - the attribute that is displayed to the user as a list (sets the displayedValue).

Storage attribute - the attribute that is stored in the directory and then used by rules and policies (sets the storedValue).

Link attribute - this attribute that holds DN links to other objects that contain proposal list data. The proposal list data are read from the linked objects rather than from the objects returned by the search.

Related Topics

- Multi-Value Editor

A.8.2.4.2. Proposal List - Item List

Use this tab to view and manage the proposal list items. To modify this list, use the multi-value editor.

Related Topics

Multi-Value Editor
General Properties

A.8.2.5. Proposal List with Language Support

A.8.2.5.1. Proposal List with Language Support - General Properties

Use this tab to create language-specific proposal lists. For each supported language, define a proposal list subentry and create a proposal list with the name of the corresponding locale. The Java Locale fields language and country are supported.

Name - the displayed name of the proposal list.

Description - the description for the proposal list.

Default Language - the default language to be used if the given language does not match

any of the child proposal lists. Provide one of the existing names of the child proposal lists.

Examples:

- To define an English proposal list, create a proposal list with Name **en**.
- For German with country Austria create a proposal list with Name **de_AT**.

Use **siemens.dxr.service.tags.ProposalNat** as the corresponding tag provider.

A.8.2.6. Report

A.8.2.6.1. Report - General Properties

A report documents the current state of an object or a set of objects. You can display reports or store them in files for further external processing.

Use this tab to view and manage the general properties of a report. These properties include:

Name - the displayed name of the report as it used throughout the DirX Identity Provisioning system.

Description - the short textual description of the report.

Types - a comma-separated list of objects for which this report can be used. This value is evaluated when you select an object in the navigation tree and then select **Run Report** from the context menu. Only the report types specified in the object description of this object are offered. See the XML element <action class="...ActionReport">.

Related Topics

- Report - Description
- Report - Format
- Status Reports

A.8.2.6.2. Report - Description

Use this tab to view and edit the content of the report description file. This file is an XML script that contains information about the report engine (<producer>), the subset filter parameters (<selector>) and the output type to be produced (<HTML>, <XML>).

Click **Edit** to edit the file. For more information on the editor features, see the help topic for the text editor.

For detailed information on how to modify the contents of a report description file, see the *DirX Identity Customization Guide* also contained in the online help.

To export the contents of the file to an external location, click **Export...** and then enter a file name in the displayed file dialog. To replace the current content with text from an external file, click **Import...** and then enter the name of the file in the displayed file dialog.

Related Topics

- Report - General Properties
- Report - Format
- Status Reports
- Text Editor

A.8.2.6.3. Report - Format

Use this tab to view and edit the content of the report format file. This file is an XSL script used for transforming the report from intermediate XML to an HTML format.

To edit the content, click **Edit**. For more information on the editor features, see the manual page for the text editor.

For detailed information on how to modify the contents of a report format file, see the *DirX Identity Customization Guide* also contained in the online help.

To export the content to an external file, click **Export...** and then enter a file name in the resulting file dialog. To replace the current content with text from an external file, click **Import...** and then enter the name of the file in the resulting file dialog.

Related Topics

- Report - General Properties
- Report - Description
- Status Reports
- Text Editor

A.8.3. Groups

Use this tab to view the general properties for a group folder. The group folder first contains a subfolder for each target system which in turn contains either group entries or other subfolders for improving the ordering of groups.

The property items shown here include:

Name - the displayed name of the group folder.

Description - the description for the group folder.

You cannot add or delete groups from this tab. Use the **Target Systems** view instead.

For a description of the Attributes tabs, see the Accounts and Groups - Attributes section.

Related Topics

- Accounts and Groups - General
- Accounts and Groups - Attributes

- Accounts
  +
- Target System
- Multi-Value Editor
- Groups
- Group - General Properties
- Group - Approval
- Group - Re-approval
- Group - *Target System Specific*
- Group - Operational
- Group - Permission Parameters
- Group - Permissions
- Group - Privileged Members
- Group - Members
- Group - Remote Members
- Group - Member of Group
- Group - Obligations
- Group - Provisioning Rules

A.8.3.1. Group - General Properties

The group object represents a user group in the target system. Groups are used to grant access to system resources by giving the group the right to access the resource and adding the user who should also have this access.

Use this tab to view and manage the general properties of a group. The following standard properties are always shown:

Identification Properties

Name (mandatory) - the name of the group as it is displayed throughout the DirX Identity Provisioning system. Once set during the addition of a new group entry, it cannot be changed after the initial save operation.

Description (optional) - a description for the group. This is often useful when group entries are listed in tables with just their names but with a description column to better identify them.

User assignment possible - whether (checked) or not (unchecked) the group can be assigned to a user entry.

Group for Priv. Accounts - whether the group handles privileged accounts (checked) or personal accounts (unchecked). In this case, the **dxrUsedBy** link is set to the user entry. Note: a group can only handle privileged or personal accounts.

Owners - the owners of this privilege. These entities are usually the persons that must approve if this privilege is assigned to a user.

Related Topics

- Multi-Value Editor
- Groups
- Group - General Properties
- Group - Approval
- Group - Re-approval
- Group - *Target System Specific*
- Group - Operational
- Group - Permission Parameters
- Group - Permissions
- Group - Privileged Members
- Group - Members
- Group - Remote Members
- Group - Member of Group
- Group - Obligations
- Group - Provisioning Rules

A.8.3.2. Group - Approval

Use this tab to view and manage the properties of a group that define the approval process to be used when the group is assigned to another object (for more information about this feature, see the section "Managing Request Workflows"). These properties include:

User Assignment - properties that apply to the assignment of a group to a user object. These properties include:

Requires approval - whether (checked) or not (unchecked) assigning the group to a user requires approval via request workflows.

Potential SoD conflict (read-only) - whether (checked) or not (unchecked) the permission is part of one or more SoD policies, which means that a conflicting privilege exists.

Approval Workflows for

Assignment - the request workflow to be used for approval if this privilege is assigned to a user. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Modification - the request workflow to be used for approval if this privilege assignment is changed. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Removal - the request workflow to be used for approval if this privilege assignment is removed. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

SoD - the request workflow to be used for approval if, during assignment of this privilege, an SoD violation is detected. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Permission Assignment - properties that apply to the assignment of a group to a permission object. These properties include:

Requires approval - whether (checked) or not (unchecked) assigning the group to a permission requires approval via request workflows.

Approval Workflows for

Assignment (Default) - the request workflow to be used for approval if this privilege is assigned to a permission. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Removal - the request workflow to be used for approval if this group is removed from a role. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

Related Topics

- Multi-Value Editor
- Groups
- Group - General Properties
- Group - Approval
- Group - Re-approval
- Group - *Target System Specific*
- Group - Operational
- Group - Permission Parameters
- Group - Permissions
- Group - Privileged Members
- Group - Members
- Group - Remote Members
- Group - Member of Group
- Group - Obligations
- Group - Provisioning Rules

A.8.3.3. Group - Re-approval

Use this tab to view and manage the properties of the group that define re-approval scenarios. These properties include:

Requires re-approval - whether (checked) or not (unchecked) all assignments of this group must be regularly re-approved. The DirX Identity re-approval process starts a re-approval workflow at the intervals defined in the other fields in this area.

Re-approval date - the next date for re-approval. Alternatively, you can define a re-approval period.

Re-approval period - the frequency with which re-approval workflows should be started. This value is only used when the **Re-approval date** is not set.

Note: If neither **Re-approval date** nor **Re-approval period** are set, the default values from the domain object are used.

Workflow - the request workflow to be used for re-approval. If this field is left blank, the request workflow engine searches for a suitable workflow template (see the subsection "Selecting Request Workflows" in the section "Request Workflow Architecture" in the chapter "Understanding the Default Application Workflow Technology" in the *DirX Identity Application Development Guide* for details).
Note: you can define any workflow here regardless of the **When applicable** settings in the workflow definition.

R*Related Topics*

- Multi-Value Editor

- Groups
- Group - General Properties
- Group - Approval
- Group - Re-approval
- Group - *Target System Specific*
- Group - Operational
- Group - Permission Parameters
- Group - Permissions
- Group - Privileged Members
- Group - Members
- Group - Remote Members
- Group - Member of Group
- Group - Obligations
- Group - Provisioning Rules

A.8.3.4. Group - *Target System Specific*

Use this tab to display and manage target-specific properties. These properties include:

Default target system-specific group properties

Dashboard

The Dashboard target system has no specific properties.

JDBC

The JDBC target system has the following specific properties:

JDBC Key - the value of the primary key in the JDBC target system.

LDAP

The LDAP target system has the following specific properties:

Primary Key (DN in TS) - the DN in the LDAP target system.

Mailing List

This target system has the following target system-specific properties:

EMail - the recipients' e-mail addresses.

Notes

This target system has the following specific properties:

List Name - the unique name of the group in the Notes target system.

Type - the type of the group. Valid values are:

- **Multi purpose**
- Mail only
- Access Control List only
- Deny List only
- Servers only

Administrators - the list of administrators in the Notes target system.

ODBC

This target system has the following specific properties:

ODBC Key - the value of the primary key in the ODBC target system.

Office 365

This target system has the following specific properties:

Primary Key (ID in TS) - the identifier of the role (security group or service plan) generated by Office 365.

Display Name - the display name attribute of the role (security group or service plan).

Mail Nickname - the mail nickname attribute for the security group.

Sku ID - the license identifier related to the selected service plan.

Group Type - the type of group (Security, Microsoft 365 - Public, Microsoft 365 - Private, Microsoft 365 - Hiddenmembership).
The value is stored in **dxrType** - SecurityGroup, MS365GroupPublic, MS365GroupPrivate, MS365GroupHiddenMembership.

RACF

This target system has no specific properties.

SAP NetWeaver UM

This target system has the following specific properties:

dxrName - the name of the SAP NetWeaver UM target system in Provisioning.

Primary Key (DN in TS) - the DN in the SAP NetWeaver UM target system.

SAP ECC UM

This target system has no specific properties.

Sipass

This target system has no specific properties.

UNIX-OpenICF

This target system has the following specific properties:

Group Name - the unique group name in a UNIX system.

GID Number - the unique GID in a UNIX system.

Windows 2008/2012

This target system has the following specific properties:

Type - the type of the group. A group can be a local, global or universal security group or distribution list. Possible values are:

- **local, Security**
- local, Distribution list
- global, Security
- global, Distribution list
- universal, Security
- universal, Distribution list

Internal Type - for internal use only. Usually, the groups have an empty value. DirX Identity Provisioning provides one special, pre-installed group named "dxr Mailbox Users", which can be used for granting an Exchange20xx mailbox to its members. This group is identified by its internal type **MAILBOX**. In its **On Assignment** and **On Revocation** tabs, it specifies naming rules that are applied to each member account added to or deleted from this group. These naming rules set / reset the mailbox-relevant attributes of the account.

Windows NT

This target system has the following specific properties:

Type - (optional) the type of the group. A group can be a local or global security group. Possible values are:

- **local**
- global

Related Topics

- Multi-Value Editor
- Groups
- Group - General Properties

- Group - Approval
- Group - Re-approval
- Group - *Target System Specific*
- Group - Operational
- Group - Permission Parameters
- Group - Permissions
- Group - Privileged Members
- Group - Members
- Group - Remote Members
- Group - Member of Group
- Group - Obligations
- Group - Provisioning Rules

A.8.3.5. Group - Operational

Use this tab to view and manage the attributes for the group that are related to standard DirX Identity operation.

Operational Attributes

State - the status of the group entry. Possible values are (for more information, see the section "Managing States" in the chapter "Managing Provisioning"):

- **ENABLED** - the group exists in the target system or shall be created.
- **DELETED** - the group shall be deleted in the target system.

End date - the date on which the group is to be deleted. When this date occurs, the group entry is deleted in DirX Identity Provisioning independent of its state in the target system.

State in target system - the status of the group in the respective target system. This value can be different from the status of the group entry as long as this information is not synchronized. Possible values are (for more information, see the section "Managing States" in the chapter "Managing Provisioning"):

- **ENABLED -** the account exists in the target system and is enabled.
- **DELETED** - the object has been deleted in the target system without DirX Identity Provisioning having requested it! This state is only set by the validation workflow along with a message in the **dxrToDo** attribute reminding the administrator to inspect this situation.
- **NONE** - replaces the empty value when the object is created by the DirX Identity Manager or agent.

Timing Properties

Create time stamp - the date and time at which the object was created in the directory.

Modify time stamp - the date and time of the last modification of this object.

Task Properties

To do - a list of manual working items for the administrator. DirX Identity Provisioning stores the results of consistency checks that it cannot automatically repair in this field. The items here are overwritten on the next consistency check.

Error - a list of error messages that show the detailed reason for inconsistent states or **To do** entries. These messages are cleared automatically when a privilege resolution succeeds.

To be analyzed (read-only) - when checked, indicates to the consistency workflow that the group has been changed and that a privilege resolution must be performed for all affected users. This flag is read-only for the administrator and is set and reset automatically by DirX Identity Provisioning workflows.

Related Topics

- Multi-Value Editor
- Groups
- Group - General Properties
- Group - Approval
- Group - Re-approval
- Group - *Target System Specific*
- Group - Operational
- Group - Permission Parameters
- Group - Permissions
- Group - Privileged Members
- Group - Members
- Group - Remote Members
- Group - Member of Group
- Group - Obligations
- Group - Provisioning Rules

A.8.3.6. Group - Permission Parameters

Use this tab to display the permission parameters for assigning the group. The privilege resolution process evaluates these attribute values for a user when the match rule of an assigned role or permission is applied. The user is granted the group when the attribute value of the group and the user (or user-to-role assignment) match.

For details, see the section "Managing Critical Parameters" in the "Managing Domain" chapter.

Related Topics

- Multi-Value Editor
- Groups
- Group - General Properties
- Group - Approval
- Group - Re-approval
- Group - *Target System Specific*
- Group - Operational
- Group - Permission Parameters
- Group - Permissions
- Group - Privileged Members
- Group - Members
- Group - Remote Members
- Group - Member of Group
- Group - Obligations
- Group - Provisioning Rules

A.8.3.7. Group - Permissions

Use this tab to display the list of permissions currently assigned to this group.

For each permission, the following properties are shown:

Permission - the name of the permission.

Description - the description of the permission.

You cannot use this tab to make direct assignment of permissions. The list shows the backward references to all permissions which have this group assigned. To view the details of a permission, click ... to the right of its row.

Related Topics

- Multi-Value Editor
- Groups
- Group - General Properties
- Group - Approval
- Group - Re-approval
- Group - *Target System Specific*
- Group - Operational
- Group - Permission Parameters
- Group - Permissions

- Group - Privileged Members
- Group - Members
- Group - Remote Members
- Group - Member of Group
- Group - Obligations
- Group - Provisioning Rules

A.8.3.8. Group - Privileged Members

Use this tab to display the list of privileged accounts currently assigned to the group.

Related Topics

- Multi-Value Editor
- Groups
- Group - General Properties
- Group - Approval
- Group - Re-approval
- Group - *Target System Specific*
- Group - Operational
- Group - Permission Parameters
- Group - Permissions
- Group - Privileged Members
- Group - Members
- Group - Remote Members
- Group - Member of Group
- Group - Obligations
- Group - Provisioning Rules

A.8.3.9. Group - Members

Use this tab to display the list of members currently assigned to this group.

For each member, the following properties are shown:

Account or group - the name of the account or group.

State of Assignment - the status of the member's assignment to the group. It can take one of the following values:

- **ENABLED**- the assignment is up to date and there are currently no problems with it.
- **IMPORTED** - the assignment was imported from the target system and was not

granted by any privilege assignment. The state switches automatically to **ENABLED** if the group is granted via a privilege assignment. The administrator must handle these assignments, which are identified in the **ToDo** query folders in each target system. By selecting the assignment in the "Member of" tab of the account, the administrator can either delete this assignment or accept it and then switch it to state **IGNORE**.

- **IGNORE** - the administrator has accepted this imported assignment. The state switches automatically to **ENABLED** if the group is granted via a privilege assignment.
- **ADD** - the assignment is a new one and must still be synchronized with the target system.
- **DELETED** - the assignment is marked for deletion.

You cannot use this tab to make direct assignment of members. The access rights of groups can only be granted by the assigned privileges. To view the details of a member, click ... to the right of its row.

Related Topics

- Multi-Value Editor
- Groups
- Group - General Properties
- Group - Approval
- Group - Re-approval
- Group - *Target System Specific*
- Group - Operational
- Group - Permission Parameters
- Group - Permissions
- Group - Privileged Members
- Group - Members
- Group - Remote Members
- Group - Member of Group
- Group - Obligations
- Group - Provisioning Rules

A.8.3.10. Group - Remote Members

Some target systems are structured into different domains. In this case, groups from one domain can contain accounts or groups from a foreign domain. Use this tab to display the list of members from foreign domains currently assigned to this group.

For each member, the following properties are shown:

Account or group - the name of the account or group.

You cannot use this tab to make direct assignment of members. To examine the details of a

member, click ... to the right of its row.

Related Topics

- Multi-Value Editor
- Groups
- Group - General Properties
- Group - Approval
- Group - Re-approval
- Group - *Target System Specific*
- Group - Operational
- Group - Permission Parameters
- Group - Permissions
- Group - Privileged Members
- Group - Members
- Group - Remote Members
- Group - Member of Group
- Group - Obligations
- Group - Provisioning Rules

A.8.3.11. Group - Member of Group

Use this tab to display the groups of which this group is a member. The tab allows you to view the next higher level of groups in a hierarchical target system.

For each member, the following properties are shown:

Group - the name of the group to which this group belongs.

Description - the description of the group object.

State of Assignment - the status of the member's assignment to the group. Possible values are:

- **ENABLED**- the assignment is up to date and there are currently no problems.
- **IMPORTED** - the assignment was imported from the target system and not granted by any privilege assignment. The state switches automatically to **ENABLED**, if the group is granted via a privilege assignment. The administrator must resolve these assignments, which are shown in the **ToDo** query folder of each target system. By selecting the assignment in the "Member of" tab of the account, the administrator can either delete this assignment or accept it and then switch its state to **IGNORE**.
- **IGNORE** - the administrator has accepted this imported assignment. The state switches automatically to **ENABLED** if the group is granted via a privilege assignment.
- **ADD** - the assignment is a new one and must still be synchronized with the target

system.

- **DELETED** - the assignment is marked for deletion.

You cannot use this tab to make direct assignments of members. The access rights of groups can only be granted by the assigned privileges. To examine the details of a particular member, click ... to the right of its row.

Related Topics

- Multi-Value Editor
- Groups
- Group - General Properties
- Group - Approval
- Group - Re-approval
- Group - *Target System Specific*
- Group - Operational
- Group - Permission Parameters
- Group - Permissions
- Group - Privileged Members
- Group - Members
- Group - Remote Members
- Group - Member of Group
- Group - Obligations
- Group - Provisioning Rules

A.8.3.12. Group - Obligations

Obligations are a method for setting or resetting account attributes when an account becomes a member of a group or is removed from the group. Obligations can be local to a group or common to several groups.

The local obligations page contains these properties:

Obligation link - the link to a common obligation object of this target system (see the Target Systems View and in this view, the folder Configuration -> Obligations).

For the rest of the properties on this page, see the description of the common obligation object.

Related Topics

- Multi-Value Editor
- Groups
- Group - General Properties

- Group - Approval
- Group - Re-approval
- Group - *Target System Specific*
- Group - Operational
- Group - Permission Parameters
- Group - Permissions
- Group - Privileged Members
- Group - Members
- Group - Remote Members
- Group - Member of Group
- Group - Obligations
- Group - Provisioning Rules

A.8.3.13. Group - Provisioning Rules

Use this tab to display the list of provisioning rules that reference this group as a privilege.

For each provisioning rule, the following properties are shown:

Name - the name of the provisioning rule.

Description - the description of the provisioning rule.

The provisioning rules listed here are backward references to all rules which have this group defined as a privilege via a privilege link. You cannot use this tab to make a direct assignment. To examine the details of a particular provisioning rule, click ... to the right of its row.

Related Topics

- Multi-Value Editor
- Groups
- Group - General Properties
- Group - Approval
- Group - Re-approval
- Group - *Target System Specific*
- Group - Operational
- Group - Permission Parameters
- Group - Permissions
- Group - Privileged Members
- Group - Members

- Group - Remote Members
- Group - Member of Group
- Group - Obligations

A.8.4. Services

A.8.4.1. Policy Agent Optimization

The Policy Agent needs two resources: one to read the rules to be processed and another one containing the users and privileges to that the rules are applied.

The parameters for user search optimization are:

- **User LDAP Size** - the page size for paged LDAP searches.Among other searches, it applies to the user searches when provisioning rules are processed.
- **User Cache Size** - the size of the MRU (most recently used) cache of the user storage area.All objects used in role resolution are stored in this cache.As a result, the number of LDAP reads is reduced.If an object is removed from the cache, it must be read from LDAP when it is accessed the next time.Reducing the cache reduces the memory requirements of the Policy Agent but decreases performance due to more frequent LDAP reads.
- **User Accumulator Size** - the cache size used when references from privileges (groups, permissions, roles) to the related users are resolved.The value of 500 should be reduced in case out-of-memory errors occur when determining the referenced users.

A.9. Auditing View

The **Auditing** view shows all configuration objects necessary to get information about the current status of objects (status reports) as well as historical information (audit trail).

For each selected item in the tree, the respective property pages are shown on the right-hand side.Click **Edit** to modify the property values.Note that saving the properties of an entry will result in another resolution of its relation to other objects.

Related Topics

Audit Policies
Audit Policy
Audit Trail

A.9.1. Audit Trail

A folder that contains all audit trail configuration objects. Properties include:

Name - the name of the folder.

Description - the description of the folder.

Related Topics

Audit Policies
Audit Policy

A.9.1.1. Audit Policies

A folder for audit policy objects. Use this tab to assign a name to the audit policy folder.

Name - the name of the folder.

Description - the description of the folder.

Related Topics

Audit Policy
Audit Trail

A.9.1.2. Audit Policy

The audit policy object defines an audit policy for a specific object type. Use this tab to specify the objects and the selected attributes to be audited for these objects.

Name - the name of the object.

Description - a description of the object.

Active - whether (checked) or not (unchecked) the audit trail mechanism is using the audit policy to write the corresponding audit information.

Identifying Attributes

The identifying attributes that are always added to the audit trail regardless of whether or not they have been changed.

Available Attributes - the list of available attributes defined by the corresponding object description for this object type. This list is derived from the corresponding object descriptions (if several object descriptions for this object exist, the attribute lists are merged).

+ You can use **targetsystem.cn** here for accounts or groups to add the target system name as an extra attribute to audit trails. This configuration eases and speeds up searches in DirX Audit.

Selected Attributes - the list of selected attributes that the audit trail mechanism adds as identifying attributes to each audit trail. If attributes specified here are empty, they are completely omitted. The Selected Attributes are displayed in two columns: Name and Display.

Name - the LDAP name of the attribute to be added as an identifying attribute to the audit trail.

Display - the name used in the audit data for the attribute. Use **Name** to define the (main) name of the object, typically the cn or the displayName. This attribute has a special

meaning in DirX Audit.

Audited Attributes

The list of attributes that is audited for the specified object type.

Object Type - the object type to be audited.
Note: this is the name of the object description in the corresponding file (for example <object name="dxrUser">). If you want to map this name to the LDAP object class, add an entry to the file:
Customer Extensions -> Object Descriptions -> ObjectTypeMap.properties

LDAP object class - the LDAP object class that the meta controller should use to identify the audit policies that belong to an object class.

Additional LDAP object class - an object class for this object description type that permits the meta controller to distinguish between users, personas and functional users. Since personas and functional users have the object class dxrUser (as the normal users), the meta controller matches the definition with the best match (a two-object-class match is preferred over a single object class match).

Additional type - the meta controller match criterion that evaluates the object's dxrType attribute. Matching is performed in the order
1) LDAP object class and additional LDAP object class match
2) LDAP object class and additional type match
3) LDAP object class matches

Audit object type - the audit object type to use in the audit trail. This property allows you to specify a convenient string that is different from the original LDAP object class and is more readable for auditors.

Available Attributes - the list of available attributes defined by the corresponding object description for this object type. This list is derived from the corresponding object descriptions (if several object descriptions for this object exist, the attribute lists are merged).

Selected Attributes - the list of selected attributes that are audited by the audit trail mechanism for this object type. The Selected Attributes are displayed in the four columns: Tag, Name, Display, and URL.

Tag - the type of audit processor creating the audit format. Available tags are:

- **attr** - is used for auditing of attribute modifications
- **any** - is used for assignments from the user to a privilege. The assignments are referenced by pseudo-properties $UserToRole, $UserToPermission, and $UserToGroup.
- **privilegeAssignment** - is used to create audit trails in the assignment format for attributes that contain a DN link or for group memberships. A separate audit trail is created for each referenced object, including the identifying attributes of source and referenced objects.

Name - the LDAP name of the attribute to be audited or a pseudo-property (for user-to-

privilege assignments).

Display - the name used in the audit data for the attribute.

URL - the reference to the object to be used to extend the list of audited attributes for this object type. This field is valid only for the **attr** tag.
Example:
storage://DirXmetaRole/$(rootDN)?content=dxrRoleParams
reads the list of permission parameters from the domain object's "dxrRoleParams attribute and then adds it to the audited attributes.

Related Topics

Audit Policies
Audit Trail

A.9.2. Status Reports

Status reports allow you to view and/or print the current status of a set of objects. This folder contains status report configuration objects.

Name - the name of the folder.

Description - the description for the folder.

Related Topics

- Audit Trail
- Report - General Properties
- Report - Description
- Report - Format

A.10. Domain Configuration View

The **Domain Configuration** view shows the configuration of the customer domain.The domain configuration contains all common configuration data like common JavaScript files, common object and property page descriptions, as well as common proposal lists and reports.Furthermore the domain object itself contains domain-wide used settings like the LDAP search size and time limit.

For a more detailed list of the contained subtrees, see the Configuration object description.

Related Topics

- Domain
- Configuration

A.10.1. Domain - Properties

A DirX Identity Provisioning domain represents an administrative area comprising all the users, the privilege and policy structure and the accounts and groups of the target systems which are to be managed with respect to a customer.

For a description of the subtrees contained in a domain, see the Configuration object.

Several tabs show the properties of a domain object. The available tabs are:

Global

This tab displays properties that provide general information about the domain. These properties include:

Name - the name of the domain object as it is displayed throughout the DirX Identity Provisioning system.

Description - a description of the domain. This property is often useful when group entries are listed in tables with only their names, but where a description column is provided to better identify them.

Domain type - the type of the domain; for example, if the domain is just for test purposes ("test domain") or a productive one ("production domain"). The value is customer-specific and is not currently used by DirX Identity Provisioning.

Include domain into topic - whether (checked) or not (unchecked) to include the domain as part of the message topic to allow subscribers (like the Java-based Server adaptors) to subscribe to topics that are only relevant for their domain.

Set this flag if you have configured your domain to run multiple Java-based Servers for load distribution purposes. Set this flag for each domain if you want to set up several Java-based Servers that each work with one domain. Use the Configuration Wizard to set up the Java-based Servers. Note: setting this flag changes the topics that components use to send messages. Changing the subscribers can result in durable subscriptions whose subscribers no longer exist, often resulting in growing message queues. Use Web Admin to check for correct setup of subscribers and delete superfluous subscriber queues.

Monitor Java Workflows - whether (checked) or not (unchecked) Java-based workflows should create Monitor entries.

Enable Functional User Handling - whether (checked) or not (unchecked) evaluation of the object descriptions for the functional users and other features - for example, handling of functional users in Web Center - are enabled. Set this flag for each domain where you want to work with functional users.

Enable for Functional Users - whether (checked) or not (unchecked) the inheritance of privileges from business objects for functional users is enabled.

Enable Persona Handling - whether (checked) or not (unchecked) evaluation of the object descriptions for the personas and other features - for example, handling of personas in Web Center - are enabled. Set this flag for each domain where you want to work with personas.

Enable User Facet Handling - whether (checked) or not (unchecked) evaluation of the object descriptions for the user facets and other features - for example, handling of user facets in Web Center - are enabled. Set this flag for each domain where you want to work with user facets.

Enable for Personas - whether (checked) or not (unchecked) the inheritance of privileges from business objects for personas is enabled.

Enable for User Facets - whether (checked) or not (unchecked) the inheritance of privileges from business objects for user facets is enabled.

Lock Disabled Users - set this flag if you want to block access for disabled users. Locking affects all LDAP clients because the DirX Server performs locking. Note that the DirX Server must be configured to enable this feature. Read the chapter "Configuring the Global Password Policy for Access Locking" in the *DirX Identity Customization Guide* for more information.

Lock Tbdel Users - set this flag if you want to block access for users in state TBDEL. Works in the same way and requires the same configuration as described for disabled users above.

Server - the server name (DNS name or TCP/IP address) of the connectivity configuration that corresponds to the domain.

Port - the port of the corresponding Connectivity configuration domain.

Use SSL - whether (checked) or not (unchecked) SSL is used for connecting to the corresponding connectivity configuration domain.

Policies

This tab displays the following policy parameters:

Disable access policies - whether (checked) or not (unchecked) access policy handling is disabled. Setting this flag disables all access policy handling by turning off the security manager. You should only set this flag during testing because it allows all user to have access to all DirX Identity features.

Enable view policies - whether (checked) or not (unchecked) previously defined Web Center view assignment policies are enabled, if the individual policies themselves have been activated. View assignment policies provide a layer of security for privilege assignments and accounts by controlling their visibility among the user community in the domain. For example, a company may not want its users to view the details of privileges or accounts that grant access to critical or sensitive resources. When this flag is set and view assignment policies have been created, users are only able to view the privilege assignments and accounts that the view assignment policies permit them to see in both Web Center and Identity Manager. If there are no view assignment policies in place and you set this flag, users can see their own privilege assignments and accounts and the privileges they are allowed to grant directly to other users. They cannot see any other privileges or accounts.

Delegation Assignment stores Operation - if checked, the new delegations are enabled. If it is not checked, the old delegations are enabled.

When activating the new delegations, the old delegation assignments and access right entries are not deleted but they will be ignored. The Business User Interface will not display them. The Identity REST Services will also ignore them.

Switching back from the new to the old delegations is not recommended. If you do it anyway, note that the delegators will still see their (now deactivated) new delegations in Web Center but Web Center will not handle the delegations correctly. Therefore, the delegators will have to delete them in Web Center, or you delete all new delegations via the DirX Identity Manager.

Enable menu policies - whether (checked) or not (unchecked) previously defined Web Center menu policies are enabled, if the individual policies themselves have been activated. If you set this flag, users see only the Web Center menus defined by the menu access policies.

Enable attribute policies - whether (checked) or not (unchecked) previously defined Business User Interface attribute policies (Access Policy - Attributes Read and Access Policy - Attributes Modify) are enabled. If enabled, the Business User Interface just presents the readable attributes that the Access Policy - Attributes Read defines. Attributes that the Access Policy - Attributes Modify defines can be changed via the Business User Interface.

Timing

This tab provides domain-wide controls that relate to time limits.

Timing-related parameters related to user, account or group object life-cycles are:

to delete an object - the maximum number of days after which DirX Identity Provisioning automatically deletes a user, account, or group object in the "to be deleted" state (TBDEL for users, DELETED for accounts and groups) from the Identity Store (Provisioning configuration). Specifying 0 for this parameter means the object is deleted today, 1 means tomorrow and so on. The default is 30 days. Use a very high value (maximum is 999,999) to simulate an infinite lifetime.
Note: you can set an individual parameter at each target system.
For more information on how these parameters control the user and account life-cycle, see the section "User and Account Life-Cycle" in the "Managing States" chapter.

to disable an object - the maximum number of days after which DirX Identity Provisioning automatically deletes a disabled account (sets its state to TBDEL and sets EndDate to the current day plus the value of MaxTime2Delete). Specifying 0 for this parameter means the object is set to TBDEL state today, 1 means tomorrow and so on. The default is 30 days. Use a very high value (maximum is 999,999) to simulate an infinite lifetime.
Note: you can set an individual parameter at each target system.
For more information how these parameters control the user and account life-cycle, see the section "User and Account Life-Cycle" in the "Managing States" chapter.

to keep error messages - the maximum number of days that a user object can remain in an inconsistent state. When this time period expires, DirX Identity Provisioning forces a

privilege resolution and sets the user object to a consistent state. This parameter applies particularly to user-privilege resolution with errors; it exists to ensure a consistent state for a user after a waiting period. Specifying 0 for this parameter means that the object will never be automatically returned from an inconsistent state to a consistent state. The default is 30 days.

Timing-related parameters for LDAP searches are:

Search size limit - the maximum number of objects that can be returned by DirX Identity Manager and agent LDAP search operations. Specify a non-negative integer or 0 for an infinite number. If the limit is reached, Identity Manager displays messages at the start of the tree view or of the result lists. If you don't find the desired object in the tree view, use the Search pane to perform a dedicated search. The default is 250 entries.

Search time limit - the maximum duration (in seconds) of DirX Identity Manager or agent LDAP search operation. If the operation does not complete by the specified limit, the operation returns an arbitrary selection of results accumulated before exceeding the time limit. Specify a non-negative integer or 0 for an unlimited time. The default is 300 entries.

Timing-related parameters for re-approval workflows include:

Reapproval date - the default date on which re-approval is started to be used for privileges that are marked for re-approval but do not specify individual re-approval date or period settings.

Reapproval period - the default time period after which re-approval is started to be used for privileges that are marked for re-approval but do not specify individual re-approval date or period settings. The default value is three months. This parameter is only used when **Re-approval date** is not set.

Approval period - the default time period before the default re-approval date or period is reached at which re-approval workflows are to be started. The default is two weeks (14 days) before the re-approval date or period is reached.

Note: re-approval parameters are evaluated in the following sequence:

- The re-approval date at the individual privilege
- The re-approval period at the individual privilege

If none of these fields is configured, the algorithm uses:

- The re-approval date at the domain object
- The re-approval period at the domain object

If none of the above fields is set, the algorithm uses a default re-approval period of 3 months.

Timing-related parameters for tickets include:

Ticket life time - the lifetime of ticket entries if the ticket was processed correctly without error=0.

Ticket life time (error) - the lifetime of ticket entries if errors occurred during ticket processing. This lifetime is usually longer than the lifetime of successfully processed tickets.

LDAP Lock

This tab provides parameters for locking request workflow or user objects and thus preventing these objects from being processed in parallel by more than one application.

The following objects can be processed in parallel:

- Request workflows that are processed in several Java-based Servers or if you have more than one workflow engine thread on one Java-based Server.
- User objects that are updated by different applications; for example, the Policy Agent, the Service Agent, an Apply Change activity in a Request Workflow, DirX Identity EventBasedResolution workflows, DirX Identity Manager, Web Center.

The LDAP lock uses the following two attributes:

- dxrLockLeaseTime - defines the time when the lock becomes invalid.
- dxrLockId - uniquely identifies the application that has set the lock.

These attributes are stored at the user object or the Request Workflow instance. If the lock is released, the two attributes are deleted.

If a Request Workflow is locked, a SchedulerTask is created in the Java-based Server so that the workflow is processed at a later time. This action is done for performance reasons so that the workflow engine thread is not blocked until the lock has been released.

If a User object is locked, you will see an error message in Web Center and in the DirX Identity Manager. Non-interactive applications like the Policy Agent, the Service Agent, or an Apply Change activity of a request workflow will retry to get the lock.

Locking-related parameters for request workflows and users) include:

Lease Time - the time (in seconds) for which the lock is valid. The default is 600 seconds for users and 60 seconds for workflows. When this time is reached, another application can get the lock.

Retry Limit if Locked - the number of retries that are performed to get the lock. The default value is 10 (both for users and request workflows). Interactive applications (for example, Web Center or DirX Identity) will ignore this field and return an error message if the entry is locked; non-interactive application (for example, the Policy Agent, the Service Agent or the Apply Change activity in a request workflow) will retry to get the lock. If the retry limit is reached, an error is generated (because the lock could not be established). a scheduler task is created for a request workflow if the workflow is locked. This action is repeated until the retry limit is reached. In this case, the workflow will not be processed.

Wait Time before Retrying - the time (in seconds) that the object is locked before processing continues. The default values are 30 (for users) and 10 (for request workflows).

Wait Time if Locked - the time (in seconds) that the application sleeps before processing

continues. The default value is 0. Note that this value should never be changed in a production environment. It can be used for demonstration purposes in order to easily generate a collision when locking an entry.

Note that these parameters can be set to different values in the two different sections: User Lock (for users) and Request Workflow Lock (for request workflows).

Permission Parameters

Permission parameters are attributes of user objects whose values control how a permission is resolved into groups. The **Permission params** tab lets you define all of the parameters you want to use for policies or for match rules in permissions. For example, you can specify that the location and/or the organizational membership will be used to control privilege assignment. The list contains one or more user attribute types. Use the multi-value editor to modify this list.

Note that these permission parameters are also used within event-based processing workflows. For more information, see the relevant sections in the *DirX Identity Application Development Guide*.

Privilege Resolution Parameters

This tab provides the following parameters:

Smooth account creation - whether (checked) or not (unchecked) the privilege resolution process checks for existing accounts when a privilege assignment requires the creation of a new account. Use this flag to minimize redundant user account creation during privilege resolution. When this flag is set and a privilege assignment requires the creation of a user account, the privilege resolution process checks the target system and the user-account link data before it creates a new account (or reports an error). In this way, accounts that may be unassigned to the user or in the wrong state (for example, "deleted"), can be re-used. Not that this process can, in some cases, assign an existing account to the wrong user. For more information, see the section about smooth account creation in the chapter "Managing Target Systems" (in the section "Managing Target System Accounts").

Inherit role parameters - whether (checked) or not (unchecked) role parameters are inherited from junior roles to senior roles. When this flag is set and you assign a senior role to a user that has junior roles with role parameters assigned, the sum of all parameters is displayed in Web Center or DirX Identity Manager to be input. Note that using this feature decreases the performance of privilege resolution.

Enable privileged accounts - whether (checked) or not (unchecked) the privilege resolution process handles privileged accounts like UNIX "root" or Windows "Administrator", which are not bound to a specific user entry.

Disable Account When Only Imported Members - whether the privilege resolution process sets the state of accounts that only have imported memberships to disabled (checked) or leaves the state as enabled (unchecked). If an account is disabled because it only has imported memberships, no endDate is set for the disabled account.

User privilege change marker - the marker (attribute name and value) that DirX Identity is

to use to indicate a change in privilege assignment that occurs in conjunction with a change to a user object. Using this kind of marker allows you to distinguish between privilege assignment changes from other changes to a user. If no privilege change occurs, the "to be analyzed" flag is set. If a privilege change occurs, both the marker and the "to be analyzed" flags are set. Note that DirX Identity does not manage this marker. Consequently, if you use this feature, you will need to create a workflow to reset the marker. (DirX Identity services reset the "to be analyzed" flag after analysis of the corresponding user entry.)

Nr of resolution listeners per Server - The number of listeners a Resolution Adapter should start for each Java-based Server. Resolution Adapters resolve user privileges to groups and accounts. They are started on every Java-based Server and obtain their resolution requests by messages from the Message Broker. This number tells the adapter how many listeners to start per Java-based server. The default is 2.

Avoiding sending resolve message of users by privilege related changes - whether (checked) or not (unchecked) to avoid sending resolve messages of users by privilege related changes. If this flag is set, the resolve messages of users are not sent by privilege related changes. Depending on the client by which the change happens:

- WebCenter: the TBA flag of the changed privilege is set and a configured privilege resolution or conistencycheck workflow can resolve the affected users later.
- Other clients such as DXI Manager, REST: the affected users are resolved immediately

This can be used to improve performance by decreasing the number of sent resolve messages.

Request Workflows

This tab provides the following parameters:

Attribute modification approval - this flag, when checked, enables automatic start-up of an approval workflow for approving modifications to attributes that you have defined with attribute policies to require special handling. See the follow-on tutorial "Applying Attribute Modification Approval" in the DirX Identity Tutorial for a demonstration of how to use this feature.

Approval on deassign - this flag, when checked, enables request workflow start-up when a privilege is removed from a user. This means that de-assigning privileges with the "Requires approval" flag set leads to an approval workflow. If unchecked, no approval workflows are started during de-assignment regardless of the "Requires approval" flag at the de-assigned privilege.

Approval content read only - this flag, when checked, makes any data to be approved read-only so that approvers cannot change it during their approval process. If the flag is not set, DirX Identity allows approvers to change the data to be approved.
Note: if this flag is not set, you can control this feature per approval activity (use the flag **Content read only**).

User creation - these flags control the user creation process for Web Services and DirX Identity Manager (Web Center is not affected). When the **Create user by request workflow** flag is checked, Web Services or Manager can use the "create user" request workflow

mechanism (available only in the DirX Identity Professional Suite) to service client requests to create users. This is especially useful if you intend to create additional attributes like a global unique ID (GUID) automatically. When the **Create user directly if no workflow available** flag is checked, Web Services and Manager can fall back to using the direct "create user" mechanism if no request workflows for user creation are found and so no error reporting is done; DirX Identity creates the user entry directly, but without automatically setting attributes like a GUID. If this flag is not checked, Web Services and Manager will report back to the client that they cannot create the user because no user creation workflows are available.

Start Approval Workflow - these flags control the behavior of starting privilege assignment approval workflows for assignments inherited by business objects and rule-assigned privileges that must be approved. You can define whether or not an approval workflow should be started. **Start Approval for Rule Assignments** controls the behavior for assigning privileges by rule; for example, via the policy execution workflow. If the privilege is subject of an approval, setting this flag to false suppresses the approval workflow. **Start Approval for BO Inherited Assignments** controls the behavior for assigning privileges via business object inheritance. If the privilege is the subject of an approval, setting this flag to true starts an approval workflow or this assignment.

Default Language - select the default language that the DirX Identity nationalization wizard is to use to resolve nationalized content in mail messages (for example, mail headers and the mail body) that request workflows send to participants in notification messages. The **Show resolved text** method resolves the message items to the related text in this language. For more information, see the section "Nationalizing Request Workflows" in the *DirX Identity Application Development Guide* (chapter "Understanding Default Application Workflow Technology").

Compliance

This tab provides domain-wide controls that relate to compliance measures, including:

Segregation of Duties (SoD) checks - whether (checked) or not (unchecked) segregation of duty checks during privilege resolution is enabled. Note: Using this feature lowers performance of the privilege resolution.

Risk Check active checks - whether (checked) or not (unchecked) risk assessment (risk-based governance) is active.

Enable Auditing for

Flags for activating auditing for the DirX Identity Services layer, request workflows and authentications against Web Center:

Service Layer - whether (checked) or not (unchecked) auditing is enabled for all additions, modifications and deletions at all configured objects in the domain. The details are controlled by audit trail policies. The location of the resulting audit files can be specified as a parameter of the workflow / connected directory.

Request Workflows - whether (checked) or not (unchecked) auditing is enabled for request workflows. You can define the path where the audit trail information for request

workflows should be written at each Java-based Server object in the Connectivity view group (see the Audit Trail Folder field).

Note: You can enable **auditing for real-time workflows** at each individual real-time workflow (see the Write Audit Log flag of the activities). There is currently no central switch to disable auditing for real-time workflows. You can define the path where the audit trail information for real-time and request workflows should be written at each Java-based Server object in the Connectivity view group (see the Audit Trail Folder field).

Authentications - whether (checked) or not (unchecked) auditing is enabled for all successful or failed attempts to log in to Web Center. You can specify the location of the resulting audit files as a parameter of the workflow/connected directory.

Enable Client Signature for

Flags for enabling secure certificate signing of selected DirX Identity operations in conjunction with smart card/chip card technology. This feature allows businesses to securely identify who performs an operation (the person swipes a smart card with a certificate on it through a card reader and then supplies a PIN in order to perform the operation) and what the operation is; for example, the person who assigns a role to another user, the role that is assigned, and the user who receives the assignment. DirX Identity "signs" the data with the smart card certificate and then stores it in a tamper-proof audit trail whose signature can be verified as necessary with a tool supplied with DirX Identity. You can enable this feature for "people" activities in request workflows, all privilege assignment requests that occur outside of request workflows, and all attribute modification requests that occur outside of request workflows with the following fields:

Request Workflow Activities - enables client signature for people activities in request workflows, but only if these are individually flagged for client signature.
Note: This flag is always audited. You cannot remove it from the audit configuration. It works only on Windows platforms and with Internet Explorer.

+ Privilege Assignment Requests - enables client signature for all privilege assignment requests outside of a request workflow.

+ Attribute Modification - enables client signature for attribute modification requests outside of a request workflow.

Checking the Certificate Owner

Flags for checking whether a logged-in user owns the smart card/chip card in use. You specify one or more secure certificate attributes to be used for comparison to the logged-in user in the field provided in this section of the Compliance tab; DirX Identity provides a built-in checking algorithm that verifies the logged-in user against the specified attribute(s). We recommend that you use this feature if you are using secure certificate signing. Note that you must put procedures in place to protect your selected attribute(s) from unauthorized changes; see the *DirX Identity Customization Guide* for details. You can also write your own certificate ownership-checking algorithm if the one provided with DirX Identity does not meet your requirements, and then specify the class name of your verifier in the field provided in this section of the Compliance tab (described below); see the *DirX Identity Customization Guide* for details.

To enable the check, use the fields in this section as follows:

Attributes to check (comma separated) - enter a comma-separated list of LDAP attributes. If no attribute is entered, the check is disabled. If one or more attribute names are entered, their values are read from the logged-in user and checked against the content of the certificate. DirX Identity's built-in checking algorithm uses this information for the check. If multiple attributes are entered, each attribute value must match (AND condition). Read the chapter "Customizing the Certificate Owner Check" in the *DirX Identity Customization Guide* for more information.

Custom Owner Verifier Classname - enter the class name of the custom certificate ownership-checking algorithm. Read the chapter "Customizing the Certificate Owner Check" in the *DirX Identity Customization Guide* for more information.

Authentication

This tab provides domain-wide controls that apply to user authentications in Web Center and Provisioning Web Services applications.

Logins with Password

This group comprises settings for authentications with user name or user DN and password.

Secondary Authentications

This group comprises settings related to logins via other mechanisms like challenge response or one-time password.

Both groups contain the same parameters:

Response Delays - a list of response delays, separated by semicolons (;). Each delay defines the number of failed authentication attempts to which it applies and a delay time in seconds, separated by a colon (:). The delay **4:2**, for example, applies to 4 attempts and delays each attempt by 2 seconds. The delay time 0 means no delay. The first delay in the list applies to the first number of failed attempts, the second to the next number, and so on. If the total number of failed attempts is exceeded, further authentication attempts will be blocked for some time, which means that any further attempt will simply fail whether or not the questions are correctly answered. To set no limit on failed attempts, leave the value for response delays empty.

Lock Duration (minutes) - the period of time for which a lock is valid. If the lock duration has expired, the response delay handling starts anew. Also, a user's failure count is cleared if his last failure time is older than the lock duration. The default lock duration value is **120**, which indicates two hours.

For example, with response delays **3:0**; **4:2**; **5:3** and lock duration **60:**

- The first three attempts are processed without any delay.
- The next four attempts are delayed by two seconds each.
- The next five attempts are delayed by three seconds each.

- After twelve (= 3+4+5) failed attempts, further attempts will be blocked for an hour.
- Any attempt after the lock period has expired will first clear the lock time and the failure count, and then be processed as a first request.
- Any successful attempt within the first 12 attempts will reset the failure count.

The delay configuration we recommend defines a maximum number of failed attempts without any delay, like **10:0**.

Password Management

This tab provides domain-wide controls that apply to password synchronization, including:

Don't sync secondary account password - whether (checked) or not (unchecked) password synchronization for secondary accounts is disabled. When this flag is set, the user password event manager workflow does not synchronize secondary accounts. This behavior applies to the following situations:

a) When an incoming password event (from the Password Listener) has been sent for a secondary account. The user password event manager does not update the user password for the relevant user and does not start any **setPassword** workflows for synchronization of other accounts. (The incoming event is simply ignored.)

b) After the user password is updated. The user password event manager only synchronizes other accounts (the password is updated in the connected system) if these accounts are not secondary accounts.

Related Topics

- Configuration
- Multi-Value Editor

A.10.2. Collection

A collection consists of a set of objects, subtrees, rule-based and sub-collections objects to be exported to an LDIF file. A collection is a powerful method for exchanging data between different instances of Connectivity databases.Typically, you use it to export object sets into your software configuration system or to transfer them from the development system to an integration or production system.The data is stored in LDIF file format.



The LDIF file format depends on the collection properties in the **dxi.cfg** file.See the *DirX Identity User Interfaces Guide* for more information about this file.

For more information about transporting data with collections, see the chapter "Transporting Data" in the *DirX Identity User Interfaces Guide***.**



The objects covered by a collection can be also deleted from the LDAP store. Use this functionality very carefully.

Use this tab to define a set of objects, subtrees or rule-based object collections to be exported.

Name

the name of the object.

Description

the description of the object.

Version

the version number of this object.

Path

the path to which the LDIF file is to be written. Use the file selector box to define the path.

Objects

the objects to be exported. During the export operation, this list of objects is exported to the LDIF file. Only the defined object is exported, no subtrees or linked objects. You can define the objects with an object selector box.

Subtrees

the subtrees to be exported. After the export of the object list, the listed subtrees are exported to the LDIF file. You can define the subtrees with an object selector box.

Rule-based

the rule specification to use when exporting objects. After the export of the subtree list, the listed objects are exported based on a rule specification. Set the rule link to a rule definition.

Collections

the collections to be exported. After the export of the rule-based list, the defined list of collections is exported into the defined file of this collection (see the **Path** specification).

Related Topics

- Collections
- Collection Rule
- Collection Rules

"Using Collections" in the *DirX Identity User Interfaces Guide*.

A.10.3. Collections

A folder for collection configuration objects. Use this tab to assign a name and a meaningful description to the collection folder.

Name

the name of the folder.

Description
the description of the folder.

Related Topics

- Collection
- Collection Rule
- Collection Rules

"Using Collections" in the *DirX Identity User Interfaces Guide*.

A.10.4. Collection Rule

A collection rule is an XML-based rule that defines the export rule for the rule-based tab in the collection object.

Use this tab to define a set of objects and subtrees to be exported.

Name
the name of the object.

Description
the description of the object.

Version
the version number of this object.

Content
the rule definition in XML format.

A collection rule is to be defined in XML format, for example:

```
<rule>
<entry classes="dxmWorkflow" childLevel="1">
<link attribute="dxmActivity-DN" />
</entry>
<entry classes="dxmActivity" childLevel="1">
<link attribute="dxmRunObject-DN" />
</entry>
...
</rule>
```

This example exports a Tcl-based workflow object, follows the links to the activities, exports these items and then follows the links to the run objects (typically jobs or other workflow definitions).

The full syntax is:

```
<rule>
<entry classes="objectclasses" [filter="filter"]
[childLevel="childLevel"] [parentLevel="parentLevel"]
[action="action"] >
 [<matchFilter ...> ... </matchFilter>]
 [<childFilter ... > ... </childFilter>]
 [<parentFilter ...> ... </parentFilter>]
 [<link attribute="linkAttribute"/>]
 [<link attribute="linkAttribute">
    <entry classes="objectclass" [filter="filter"]
[childLevel="childLevel"] [parentLevel="parentLevel"]
[action="action"] >
      [<matchFilter ...> ... </matchFilter>]
      [<childFilter ... > ... </childFilter>]
      [<parentFilter ...> ... </parentFilter>]
    </entry>
    ...
 </link>]
...
</entry>
...
<!-- use this default entry to process objects that are not yet
matched by previous rules -->
</entry classes="*">
</rule>
```

with these sub elements:

objectclasses

a space or comma separated list of object classes that is used in the LDAP search to retrieve this type of objects.

***filter* (optional)**

if an LDAP filter is defined, only the objects that match the filter condition are exported.



Filter definitions must be enclosed in brackets, for example use "(cn=RoleCatalogue)" instead of "cn=RoleCatalogue".
Alternatively you can specify a **matchFilter** DSML filter clause.

***childLevel* (optional)**

the depth of the sub tree to be exported. Possible values are:*
ignore* - ignore this entry completely
none - ignore this entry but follow the links
all or **0** - the whole subtree
1 - just this entry and no sub objects (default value).
2 - this entry and one level of sub objects
3 - this entry and two levels of sub objects
...

ldapFilter

exports all children down to the level where the LDAP filter condition is valid (this object and lower level objects are not exported).
Note that filter definitions must be enclosed in brackets, for example use "(cn=RoleCatalogue)" instead of "cn=RoleCatalogue".
Alternatively you can specify a **childFilter** DSML filter clause.

***parentLevel* (optional)**

the level of parent objects to be exported. Possible values are:*
none -* no parents (default)
all or **0** - all parents
1 - one level of parents above the given entry
2 - two levels of parents above the given entry

i. +

ldapFilter

exports all parents up to the level where the LDAP filter condition is valid (this object and higher level objects are not exported).
Note that filter definitions must be enclosed in brackets, for example use "(cn=RoleCatalogue)" instead of "cn=RoleCatalogue".
Alternatively you can specify a **parentFilter** DSML filter clause.

***action* (optional)**

an action that defines how the entry is processed:
default - processes the entry (export or delete). This is the default.
skip - this entry is not exported or deleted but its child and parent definitions are processed.

***linkAttribute* (optional)**

the name of the attribute to be followed to other objects.
Use this syntax to define a specific attribute: "dxmSpecificAttributes:channelparent".

When processing an LDAP object (for example a user), then the entry elements are processed sequentially from top to bottom and the first matching element is used to process the object.



Entries can contain link definitions. Link definitions can itself contain entry definitions and so on. This allows defining a different behavior for the same object at different levels. It is also a means to control endless loops

> effectively. The inner elements have higher priority than the root elements of the same type.

A.10.5. Collection Rules

A folder for collection rule objects. Use this tab to assign a name and a meaningful description to the collection folder.

Name - the name of the folder.

Description - the description of the folder.

Related Topics

- Collection
- Collections
- Collection Rule
- Collection Rules

"Using Collections" in the *DirX Identity User Interfaces Guide*.

A.10.6. Collection Rule Syntax

This section describes the collection rule syntax, which is based on the XML Schema Definition (XSD).

Element rule - the root element of the collection rule. A collection rule is XML-based content that defines the export strategy for an LDAP object.

Child elements

- **entry**
  - occurrence: 0-unbounded

Element entry - the object exporting strategy.

Attributes

- **classes** - comma- or space-separated object class names. An entry matches an object if all values in classes are contained in the objectClass attribute of the object. It provides a basic entry filter. The value * matches any object class, which enables you to specify a "default" entry.
  - type: string
  - use: required
- **childLevel** - the scope to use to export from the current object; that is, the depth of the subtree to be exported. Values can be all (all children), none (no children), 1 (the object only), 2..n (object and 1 or n direct children). See levelValues for details.
  - type: levelValues

    - use: optional
    - default: 1
- **parentLevel** - the level to use to export parent objects. Values can be all (all parents), none (no parents), ignore (ignore entry), 1 (direct parent), 2..n (level of parents). See levelValues for details.
    - type: levelValues
    - use: optional
    - default: none
- **action** - actions that are applied to the processed object. See actionValues for details.
    - type: actionValues
    - use: optional
    - default: default
- **filter** - obsolete. Use the matchFilter element instead. The LDAP filter definition must be enclosed in brackets, for example, use "(cn=RoleCatalogue)" instead of "cn=RoleCatalogue".
    - type: ldapFilter
    - use: optional

Child elements

- **link** - the object attribute that must be followed if an object related to this entry is exported.
    - occurrence: 0-unbounded
- **matchFilter** - an entry matches an object (object will be processed by the entry) if the object matches the filter. The filter provides finer object-matching. The filter is specified in DSML format. The dsml namespace declaration is necessary.
    - occurrence: 0-1
- **childFilter** - stop exporting the children subtree if the filter is not matched.
    - occurrence: 0-1
- **parentFilter** - stop exporting the parents subtree if the filter is not matched.
    - occurrence: 0-1

Parent elements

- **rule**
- **link**

Element link - the attr link-following strategy.

Attributes

- **attribute** - the name of the attribute to be followed to other objects. To define a specific attribute, use, for example, "dxmSpecificAttributes:channelparent".
    - type: string

- use: required

Child elements

- **entry** - the export configuration of the object to which this link points. That is, that link definitions can themselves contain entry definitions and so on. This configuration allows defining a different behavior for the same object at different levels and allows for effective control of endless loops. The inner "entry" elements are considered with higher priority compared to root "entry" elements of the same type.
  - occurrence: 0-unbounded

Parent elements

- **entry**

Element **childFilter** - the DSMLv2 syntax filter used to find the matching entry for the processed object.

Parent elements

- **entry**

Element **parentFilter** - the DSMLv2 syntax filter that stops exporting of parents if the filter is matched.

Parent elements

- **entry**

Element **matchFilter** - the DSMLv2 syntax filter used to define the condition for children nodes that are exported.

Parent elements

- **entry**

Simple type **ldapFilter** - obsolete. Use DSML filters instead. Restricts the parents / children nesting level.

Patterns

- [!&|]{0,1}(.*) - the filter is specified in LDAP format, for example, (objectclass=dxrContainer).

Simple type **levelValues** - export parents / children nesting level.

Patterns

- **none** - do not export parents / children, but do export links.
- **all** - export all parents / children of the entry.
- [1-9][0-9]* - export parents / children up to the given level specified by a positive number.
- **ignore** - obsolete. Use the action attribute to skip objects. Do not export any direct parents / children and links.
- **inherit** - For the childLevel attribute, adopt the nesting level from the superior entry. This pattern has no meaning for the parentLevel attribute (it is the same as none).

- [!&|]{0,1}(.*) - obsolete. Use the childFilter / parentFilter elements to define the DSML filter. Specifies the LDAP filter. Stop processing at first parent / child that does not meet the filter. The filter is specified in LDAP format, for example (objectclass=dxrContainer).

Simple type **actionValues** - actions to be applied on the processed object.

Patterns

- **default** - perform the normal processing of the object (export or delete).
- **skip** - skip exporting / deleting of the matched object itself, but continue processing on parents / children / links. Ignoring parents / children is driven by parentLevel and childLevel.

Related Topics

- Collection
- Collections
- Collection Rule
- Collection Rules

"Using Collections" in the DirX Identity User Interfaces Guide.

A.10.7. Role Parameter

Role parameter objects define the role parameters allowed in a role definition.

Use this tab to view and manage the general properties of a role parameter. The items displayed here are:

Name - the displayed name of the role parameter as it used throughout the DirX Identity Provisioning system.

Description - the short textual description for the role parameter

Type - the parameter type. Supported types include:

- String - a list of strings.
- Text - a simple text string.
- Integer - a list of integers.
- DN - a list of distinguished names that is displayed as a flat list during role parameter selection.
- Hierarchical DN - a list of distinguished names that are displayed as a tree during role parameter selection.

Depending on the selected type, you can define:

For type String:

Default Value - the default value for the role parameter if the role is assigned. This value should be one of the values provided by the role parameter value list.

Single value - whether (checked) or not (unchecked) to restrict the role parameter to one value. Setting this flag means that you can assign this role to a specific user only once per time period.
For example, if you assign role X to user A with parameter Y=a and no time restriction (start/end date), you cannot assign the same role with another parameter value (for example Y=b).
If you set time restrictions, you can assign the role multiple times to the same user if the time intervals are distinct, for example: assign role X to user A with parameter Y=a from 1.1.2000 to 31.12.2005 and the same role to the same user with parameter Y=b from 1.1.2006 to now.

You can define a proposal list that describes the allowed role parameter values.

Proposal List - the proposal list.

Attribute Name - the multi value attribute that holds the allowed values.

Reference Expression - for user-specific proposal lists, the attribute that holds the proposed values. Base objects for this expression can be user or role. Examples:

${user.initials} - take the proposed values from the initials attribute of the user. The source for this assignment is provided in a user attribute.

${user.dxrcontextlink.dxrproposedvalues} - follow the dxrcontextlink of the user and take the proposed values from the dxrproposedvalues attribute. dxrProposedValue must be added to the business object's schema; provided by a business object of the user.

The attributes used in the expressions must be defined in the corresponding object descriptions. The attribute that holds the proposal list should be defined as multi-valued. This attribute is treated as a proposal list. So you can use the same expression as for proposal lists for the values.

Values - a discrete list of string values.

Note: the values from the proposal list and the discrete list of values are combined into one (sorted) list.

For type Text:

Default Value - the default value for this role parameter if the role is assigned.

Single value - whether (checked) or not (unchecked) to restrict this role parameter to one value. Setting this flag means that you can assign this role to a specific user only once per time period.
For example, if you assign role X to user A with parameter Y=a and no time restriction (start/end date), you cannot assign the same role with another parameter value (for example Y=b).
If you set time restrictions, you can assign the role multiple times to the same user if the time intervals are distinct, for example: assign role X to user A with parameter Y=a from

1.1.2000 to 31.12.2005 and the same role to the same user with parameter Y=b from 1.1.2006 to now.

Mandatory - whether (checked) or not (unchecked) input to the text field is mandatory.

For type Integer:

Default Value - the default value for this role parameter if the role is assigned. The value you specify should be one of the allowed values.

Single value - whether (checked) or not (unchecked) to restrict this role parameter to one value. Setting this flag means that you can assign this role to a specific user only once per time period.
For example, if you assign role X to user A with parameter Y=1 and no time restriction (start/end date), you cannot assign the same role with another parameter value (for example Y=2).
If you set time restrictions, you can assign the role multiple times to the same user if the time intervals are distinct, for example: assign role X to user A with parameter Y=1 from 1.1.2000 to 31.12.2005 and the same role to the same user with parameter Y=2 from 1.1.2006 to now.

You can define a proposal list that describes the allowed role parameter values.

Proposal List - the proposal list.

Attribute Name - the multi-value attribute that holds the allowed values.

Minimum - a start value for the range of allowed integer values.

Maximum - an end value for the range of allowed integer values.

Values - the discrete list of integer values.

Note: the values from the proposal list and the discrete list of values are combined into one (sorted) list.

For type DN:

Default Value - the default value for this role parameter if the role is assigned. The value you specify should be one of the values provided by the role parameter value list. If the parameter values are of type DN, then you must specify the complete DN.

Single value - whether (checked) or not (unchecked) to restrict this role parameter to one value. Setting this flag means that you can assign this role to a specific user only once per time period.
For example, if you assign role X to user A with parameter Y=cn=abc and no time restriction (start/end date), you cannot assign the same role with another parameter value (for example Y=cn=def).
If you set time restrictions, you can assign the role multiple times to the same user if the time intervals are distinct, for example: assign role X to user A with parameter Y=cn=abc from 1.1.2000 to 31.12.2005 and the same role to the same user with parameter Y=cn=def from 1.1.2006 to now.

You can define a proposal list that describes the allowed role parameter values.

Proposal List - the proposal list.

Attribute Name - the multi-value attribute that holds the allowed values.

Object Type for Access Control - if an object type is selected, you can only assign role parameter values that have a matching grant policy for the defined object type (subject: the user that assigns the role; resource: the object defined by the DN). We recommend selecting RoleParam/dxrRoleParam, which functions similarly to the HDN parameters with “Apply Policy for Selection” checked. If empty, access policies are not evaluated and you can select all of the specified DN values.

You can define a dynamic object search that describes the allowed role parameter values.

Root - the root at which to start the object search (dynamic search).

Object Class - the object class to be retrieved (dynamic search).

Display Attribute - the attribute that is used for display. The DNs of the objects is no longer shown but are stored as values. Be aware that the display attribute values should be unique, or the user will be unable to select the correct value.

Static DNs - a fixed list of additional objects.

Note: the values from the proposal list, from the object search and from the fixed list of additional objects are combined into one (sorted) list.

For type Hierarchical DN:

Default Value - the default value for this role parameter if the role is assigned. The value should be one of the values provided by the role parameter value list. If the parameter values are of type DN, then you must specify the complete DN.

Apply policy for selection - whether (checked) or not (unchecked) only values that have a matching access policy for type dxrRoleParam are allowed. See the *DirX Identity Provisioning Administration Guide* -> "Managing Policies" -> "Delegated Administration" -> "Managing Access Policies" -> "Policies for Hierarchical Role Parameters" for details.

Single value - whether (checked) or not (unchecked) to restrict this role parameter to one value. Setting this flag means that you can assign this role to a specific user only once per time period.
For example, if you assign role X to user A with parameter Y=cn=abc and no time restriction (start/end date), you cannot assign the same role with another parameter value (for example Y=cn=def).
If you set time restrictions, you can assign the role multiple times to the same user if the time intervals are distinct, for example: assign role X to user A with parameter Y=cn=abc from 1.1.2000 to 31.12.2005 and the same role to the same user with parameter Y=cn=def from 1.1.2006 to now.

The allowed role parameter values are defined via a dynamic search:

Root - the root at which to start the object search (dynamic search).

Object Class - the object class to be retrieved (dynamic search).

Display Attribute - the attribute that is used for display. The DNs of the objects are no longer shown but are stored as values. Be aware that the display attribute values should be unique, or the user will be unable to select the correct value.

For a multi-valued role parameter, you can define:

Unique Values - whether (checked) or not (unchecked) to restrict the multi-valued role parameter to unique values. Setting this flag means that you can assign this role with a specific role parameter value to a specific user only once per time period. For example, if you assign role X to user A with parameter Y=a and no time restriction (start/end date), you cannot assign the same role with the same parameter value (Y=a).

If you set time restrictions, you can assign the role multiple times to the same user if the time intervals are distinct; for example, assign role X to user A with parameter Y=a from 1.1.2000 to 31.12.2005 and the same role to the same user with parameter Y=a from 1.1.2006 to now. The default is unchecked, which means no check is done for multi-valued role parameters. You can assign a given role parameter value multiple times for a given user-role relationship.

Related Topics

Role Parameters
Role - General Properties

A.10.7.1. Role Parameters

This tab shows the general properties for the role parameter folder.

The property items shown here include:

Name - the displayed name of the role parameters folder.

Description - the description of the role parameters folder.

Related Topics

Role - General Properties
Role Parameter

A.11. Miscellaneous

This set of manual pages contains help information on special tools or just property page elements used in DirX Identity Provisioning.

- Activity Editor Assignment Editor
- Multi-Value Editor

- Query Folder - General Properties
- Query Folder - Filter
- Text Editor
- Run Report
- Workflow Editor

A.11.1. Activity Editor

Use the activity editor to view and configure request workflow activities. It shows all sub-activities in a fixed layout. You can activate and deactivate sub-activities.

A.11.1.1. Viewing

If you do not click **Edit**, you are in view mode. Click the background and then select one of the following options from the context menu:

Open - opens a new window for this object's property pages that provides more space for editing and viewing.

Zoom In - shows a part of the view in more detail.

Zoom Out - shows more of the view with less detail.

Reset Zoom - resets the view to the default view.

Click an object and then select an option from the context menu:

Open - opens a new window for this object and shows its property pages. You can click **Edit** to change parameters of this object. Use **OK** to store your changes or **Cancel** to abort the edit operation.

A.11.1.2. Editing

Clicking **Edit** switches to edit mode. The context menu options vary depending on the selected object. There are inactive objects (gray color) and active objects (orange color).

Click the background and then select one of the following options from the context menu:

Configure - opens a wizard that presents all activity tabs in a sequence.

Open - opens the activity objects and shows all tabs in parallel.

Zoom In - shows a part of the view in more detail.

Zoom Out - shows more of the view with less detail.

Reset Zoom - resets the view to the default view.

Select a gray (inactive) object and then use the following options from the context menu:

Insert - allows you to activate and configure this object. The configuration wizard starts.

Select a non-gray object and then use the following options from the context menu:

Configure - opens the configuration wizard for this object. The wizard allows you to select a new activity type and displays the tabs of this object in a sequence for configuration.

Open - opens a new window for this object and shows its property pages. You can click **Edit** to change parameters of this object. Click **OK** to store your changes or **Cancel** to abort the edit operation.

Remove - deactivates this object and its complete configuration from the activity.

A.11.1.3. Other Features

The activity editor also provides the following features:

Selection of single objects - click an object to select it.

Selection of multiple objects - draw a rectangle to select a number of objects. To select objects, they must lie completely within the rectangle boundaries.

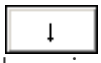
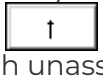
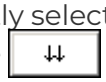
Deselection of objects - click the background to deselect a selected object.

Notes

- The activity editor contains a fixed set of objects. You cannot create or delete objects. Instead, you can only configure (insert) or unconfigure (remove) objects.
- You can only run menu options on a single object. If multiple objects are selected, the selection is reduced to only one object.
- There is currently no Undo / Redo feature implemented.

A.11.2. Assignment Editor

Use the **assignment editor** to change the set of assigned items to a particular object. You can use it to assign or unassign roles to a user, permissions to a role, groups to permissions, and so on. The assignment editor consists of two tables: the upper one shows the list of available items, the lower one the list of already assigned items.

Use the buttons between the two tables to exchange items. The  button moves the item currently selected in the upper table to the lower one, which assigns the selected item. The  button moves the item currently selected in the lower table to the upper one, which unassigns the selected item. Use the  button to assign all available items. Use the  button to unassign all items.

To better read the content of one of the tables, you can enlarge its width: click in the button bar between the two tables with the left mouse button, and then drag it to the desired position.

The list of available items can become very long and thus difficult to view. Use the search panel at the top of the assignment editor to search for a particular subset of available items.

The fields have the following meaning:

Search base - the base node object where the search starts. This value is a distinguished name and can be entered directly into the field or selected from a node chooser dialog which opens when you click [...]. Use the arrow button to the right of the text field to insert already used names again.

Search for - the filter expression that consists of the attribute name, the comparison operator and the filter value. Use the arrow buttons to the right of the text fields to select an attribute name, a comparison operator, or an already used filter value.

The search is started by clicking on the button right to the value field.

If a role uses role parameters, the assignment dialog box is displayed. Switch to the role parameter tab and then define the required parameters. After you click **Save**, some additional checks are performed. If a single-value role parameter was assigned, it is checked that for a specific point in time only one value is specified. Assume that a role X with parameter A is already assigned with no start date but an end date of tomorrow, and the same role is assigned with parameter B starting today, with no end date. In this case, the assignment is refused. Change the start and end date settings of the assignment so that no overlap occurs.

A.11.3. Multi-Value Editor

The **multi-value editor** is a simple tool for managing value lists. It shows the list of current items and provides two buttons to add new items to the list or to remove elements.

To modify the value list, click **Edit**. Click on an entry in the list to change it. If the entry is a simple text value, you can type in the new text directly. For more complex values, an appropriate editor is presented (for example, if only predefined values are allowed, a combo box is displayed).

To add a new item, click **Insert**. This action adds a line and opens the appropriate editor for setting the value.

To remove the currently selected item from the list, click **Delete**.

A.11.4. Query Folder - General Properties

Use this tab sheet to view the general properties of a query folder. The query folder is used to filter out a subset of objects that match the given set of filter criteria. The views of DirX Identity Provisioning contain predefined query folders at various places.

The property items shown here include:

Name - the displayed name of the query folder.

Description - the description of the query folder.

Related Topics

A.11.5. Query Folder - Filter

Use this tab to configure the search criteria for a query folder. The items shown here include:

Search base - the distinguished name of the node in the directory tree where the query starts. Use the ... button to select a node from the tree or type in the distinguished name directly.

Search filter - the match criterion for the query. This value is usually an expression of type *attribute**=**value* or a combination of such expressions in LDAP syntax. Use the LDAP filter editor to build a valid filter condition.

You can use the following expression types in the filter for time attributes:

- **$***base* or **$**(*base*) - the current time, depending on *base*. *base* can be:

  NOW or **gmtime** or **time** - the current time in GMT.
  localtime - the current time in local time zone.
  date - the time of this day start in GMT.
  localdate - the time of this day start in the local time zone.

  Examples:

  `dxrExpirationDate>=$NOW` - retrieves all entries that will expire in the future.

  `&(dxrStartDate>=$(date))(dxrStartDate<=$(time))` - retrieves all entries that were activated today up to now.

- **$***base operation constant* or **$**(*base operation constant*) - the time plus or minus a constant. The format of *constant* is:

  *n***y***n***M***n***d***n***h***n***m***n***s**

  where *n* is the number of time units. The time units are:

  y years
  M months
  d days
  h hours
  m minutes
  s seconds.

  The order of time units is fixed, but each unit is optional. For example:

  `(dxrStartDate>=$(NOW-3h))` - retrieves all entries that were created within the last three hours.

`(dxrExpirationDate<=$(gmtime+1y6M))` - retrieves all entries that expire in one and a half years.

A number without a time unit indicates days.

- **$***base operation* **$***variable* or **$**(*base operation* **$***variable*) - the current time plus or minus a variable. The values of these variables are the values described above for constants, for example:

  `(dxrStartDate>=$(NOW-$Delta))` - each time the filter is evaluated (select it or use the refresh button to start the evaluation), the variable is displayed with the previously entered value. Change the value if necessary and click **OK**.
- **$***variable* - the specified value is used in the filter, for example:

  `cn=$StartsWith*` - selects all objects where **cn** starts with the specified value. Each time the filter is evaluated (select it or use the refresh button to start the evaluation), the variable is displayed with the previously entered value. Change the value if necessary and click **OK**.

Search scope - Defines the scope of the query. One of the following values is allowed:

- **0** - **BASE** - (Base Object) - the search is done on the search base only
- **1** - **ONELEVEL** - (One Level) - the search is done on the first level below the search base.
- **2** - **SUBTREE** - (Subtree) - the search is done on the whole subtree below the search base.

Result limit - the maximum number of entries to be returned by the query. If more results are found, an error "Size limit exceeded" occurs.

Related Topics

General

A.11.6. Text Editor

Use the **text editor** to edit script objects, like XML object descriptions and property page descriptions, and JavaScript files.

To modify the current text content, Click **Edit**, and then place the cursor anywhere in the text area to enter and delete characters.

Right-clicking inside the text area displays a pop-up menu that provides the following actions:

New Window - opens a new separate large window for the editor. This window can be sized independently from the main window but closes when the main window closes. However, it provides the same functionality as the editor window in the property page.

Undo (Edit mode only) - undoes the last action in the current editing session. Note, that when fluid text typing has happend, the whole typed-in string will be removed and not just

the most-recently typed character.

Redo (Edit mode only) - undoes the recent undo action.

Cut (CTRL+X) (Edit mode only) - cuts the currently selected part of text and copies it to the clipboard.

Copy (CTRL+C) - copies the currently selected part of text to the clipboard.

Paste (CTRL+V) (Edit mode only) - pastes the current textual content of the clipboard into the text area at the current cursor position.

Find... (CTRL+F) - opens a dialog for entering search criteria. **Find what** specifies the text to be found. **Match case** should be marked if the find operation should perform a case-sensitive lookup. **Whole word** should be marked if the text to be found should appear as separate words and not as parts of any other words.

Replace... (CTRL+R) - (Edit mode only) opens a dialog for entering replace parameters. **Find what** specifies the text to be found. **Replace by** would be the replacement text. **Match case** should be marked if the find operation should do a case-sensitive lookup. **Whole word** should be marked if the text to be found should appear as separate words and not as parts of any other words. **Confirm** should be marked if the system should always ask to replace a matching string.

Go to insertion point (CTRL+T) - jumps back to the original position, if you have clicked anywhere in the text with the left mouse button and then scrolled to another position.

Find other block end (CTRL+B) - finds the opposite end and marks the respective character or sequence of characters when you have selected a brace or a sequence of characters starting a block, like a comment in a JavaScript.

Select all (CTRL+A) - selects the entire content of the editor window.

Toggle block resolution - disabled in DirX Identity Provisioning because it has no meaning here.

Line Numbers - turns line number display on and off.

A.11.7. Run Report

Use this dialog to modify some pre-configured parameters just before the respective report is run. The table shown at the top of the dialog lists all report templates that can be used at this invocation point.

Once a template is selected, the property values below the table are updated. The following properties can be changed before the report creation is started (by clicking **Run Report**):

Search base - the node of the directory tree where the search for objects to be reported on starts.

Search scope - the scope of the search operation. This item can take one of the following values:

- **BASE OBJECT** - the search operation is only done on the search base itself.
- **ONE LEVEL** - the search operation searches in the search base and all nodes directly below this node.
- **SUBTREE** - the search operation extends to the whole subtree below the search base.

Type - the output type for the report to be created. Select one of the following values:

- **HTML** - the output is written in HTML format. This allows for viewing the report in an internet browser.
- **XML** - the output is written in XML format and can then be processed further by an appropriate application.

Output to viewer - whether (checked) or not (unchecked) to load the results into a HTML or text viewer immediately after creation.

Output file - the name of the output file. This field is enabled only when the **Output to viewer** check box is unchecked.

Related Topics

Report

A.11.8. Workflow Editor

Use the workflow editor to view and configure request workflows. The editor shows all activities and all other objects in an intuitive graphical view.

A.11.8.1. Viewing

If you do not click **Edit**, you are in view mode. Click the background and then select one of the following options from the context menu:

Open - opens a new window for this object's property pages that provides more space for editing and viewing.

Zoom In - shows a part of the view in more detail.

Zoom Out - shows more of the view with less detail.

Reset Zoom - resets the view to the default view.

A.11.8.2. Editing

Click **Edit** to change to edit mode. Click the background and then select one of the following options from the context menu:

Configure - opens the configuration wizard for this object. It displays the tabs of this object in a sequence.

Open - opens a new window for this object's property pages that provides more space for

editing and viewing.

Insert Activity - selects an activity type. After defining the insertion point on the background, a dialog opens where you can select an activity group (automatic or people). Select the appropriate activity type and then click **OK**.

Insert End Event - creates an end point for your workflow (note: after a new workflow is created, a default start and endpoint is automatically provided).

Insert Start Event - creates a start point for your workflow (note: after a new workflow is created, a default start and endpoint is automatically provided).

Insert AND Merge - inserts an AND merge point. The result of several activities (the application state) can be merged to a combined result.

Insert OR Merge - inserts an OR merge point. The result of several activities (the application state) can be merged to a combined result.

Import - imports an existing activity. After defining the insertion point on the background, a tree browser opens for selecting the activity to be imported. Select the activity, provide the new name and then click **OK**.

Zoom In - shows a part of the view in more detail.

Zoom Out - shows more of the view with less detail.

Reset Zoom - resets the view to the default view.

Click **Save** to store your changes. Click **Reset** to discard your changes.

A.11.8.3. Other Features

The workflow editor also includes:

Selection of single objects - click an object to select it.

Selection of multiple objects - draw a rectangle to select a number of objects. To select objects, they must lie completely within the rectangle boundaries.

Selection of all objects - press CTRL -A.

Modification of a selection - hold the CTRL key while clicking additional objects. If an object is not yet selected, it is added to the selection. If an object is already selected, it is removed from the selection.

Deselection of objects - click the background to deselect all objects.

Moving objects - drag a selection of objects to another location.

Connecting objects - move the cursor over the object boundaries. You will notice hot spots that indicate connection points. Drag the cursor from one hot spot to another one. If the connection is allowed:

A line is displayed immediately if no further configuration is necessary.

A wizard is displayed that requests additional parameters (for example, start conditions) before the line is connected.

Descriptive text - lines with conditions can have descriptive text (for example "accepted" or "rejected"). Select a line, select **Configure** and then edit the **Name** field to enter or modify the text. Click **Finish** to store the change. Move the text to the location where it makes sense.

Deletion of objects - select a single object and then use **Delete** from the context menu. You can only delete single objects because you need to confirm each deletion: deletion of an object can destroy lots of configuration information.



- You can only run menu options on a single object. If multiple objects are selected, the selection is reduced to only one object.
- There is currently no Undo / Redo feature implemented.

A.12. Target System Wizard - Overview

The target system wizard allows for easy creation of a target system in the Provisioning and the Connectivity views. After you enter all required information into the wizard, the target system is ready to use, including all of the workflows necessary for initial load, synchronization, validation and password synchronization.

When you use this wizard for the first time in a domain, DirX Identity assists you in creating a brand new Connectivity scenario.

When you subsequently use it for additional target systems, DirX Identity adds only the necessary components to the existing Connectivity scenario.



If the target system requires a target system-specific schema extension, make sure that you extend the schema before you create the new target system.

Related Topics

- Type Selection
- General Properties
- Target System Advanced
- Target System Timing
- Target System Options
- Connectivity Configuration
- Identity Store
- Connectivity Scenario
- Associated Connected Directory

- Connected Directory Configuration
- Provisioning Workflows

A.12.1. Target System Wizard - Type Selection

Select one of the provided target system types to use as a template.

If the target system you want to add is not displayed in the list of available templates, select a template of a target system that most resembles your target system and adapt it to your requirements. See the DirX Identity Customization Guide for more information.

Check **Accounts and groups in common subtree** if you want accounts and groups to be kept in a common **Accounts and Groups** subtree under your target system. Do not check this field if you want the wizard to create separate folders for **Accounts and Groups**. This makes sense especially for LDAP and ADS type target systems.



The template name for a target system of type ADS is still called Windows 2000 for compatibility reasons, but it is the appropriate template for all Windows systems of versions greater than Windows 2000 with an Active Directory installed.

Related Topics

- Overview
- General Properties
- Target System Advanced
- Target System Timing
- Target System Options
- Connectivity Configuration
- Identity Store
- Connectivity Scenario
- Associated Connected Directory
- Connected Directory Configuration
- Provisioning Workflows

A.12.2. Target System Wizard - General

Set the general properties of the target system:

- **Name** - (mandatory) - the name of the target system as it is used for display and identification purposes throughout the DirX Identity system. Once set during adding a new target system, it can no longer be changed after the initial save operation.
- **Description** - (optional) - the description for the target system. This is often useful, when target system entries are listed in tables just with their names but also with a description column to better identify them.

- **Local admin** - (optional) - the distinguished name of the user who is permitted to administer the target system or who is responsible for this target system.

Related Topics

- Overview
- Type Selection
- Target System Advanced
- Target System Timing
- Target System Options
- Connectivity Configuration
- Identity Store
- Connectivity Scenario
- Associated Connected Directory
- Connected Directory Configuration
- Provisioning Workflows

A.12.3. Target System Wizard - Advanced

Set the advanced parameters for the target system. The properties listed here are target system-specific.

See Target Systems - Advanced for more information.

Related Topics

- Overview
- Type Selection
- General Properties
- Target System Timing
- Target System Options
- Connectivity Configuration
- Identity Store
- Connectivity Scenario
- Associated Connected Directory
- Connected Directory Configuration
- Provisioning Workflows

A.12.4. Target System Wizard - Timing

Set the timing parameters for the target system. You can set the time to delete and disable an object. If these parameters are not set, the wizard uses the corresponding parameters

from the domain object as the default.

Related Topics

- Overview
- Type Selection
- General Properties
- Target System Advanced
- Target System Options
- Connectivity Configuration
- Identity Store
- Connectivity Scenario
- Associated Connected Directory
- Connected Directory Configuration
- Provisioning Workflows

A.12.5. Target System Wizard - Options

Set optional parameters for a target system. The properties listed here are target system-specific.

See Target Systems - Options for more information.

Related Topics

- Overview
- Type Selection
- General Properties
- Target System Advanced
- Target System Timing
- Connectivity Configuration
- Identity Store
- Connectivity Scenario
- Associated Connected Directory
- Connected Directory Configuration
- Provisioning Workflows

A.12.6. Target System Wizard - Connectivity Configuration

Each Provisioning domain requires a corresponding Connectivity domain. The wizard has detected that the location of the Connectivity domain for this Provisioning domain is not yet known.

Supply the properties for the location of the Connectivity domain.

Related Topics

- Overview
- Type Selection
- General Properties
- Target System Advanced
- Target System Timing
- Target System Options
- Identity Store
- Connectivity Scenario
- Associated Connected Directory
- Connected Directory Configuration
- Provisioning Workflows

A.12.7. Target System Wizard - Identity Store

The wizard has detected that the corresponding scenario in the Connectivity domain does not correctly identify which connected directory acts as the Identity Store. Supply the information to identify the Identity Store in your scenario.

Related Topics

- Overview
- Type Selection
- General Properties
- Target System Advanced
- Target System Timing
- Target System Options
- Connectivity Configuration
- Connectivity Scenario
- Associated Connected Directory
- Connected Directory Configuration
- Provisioning Workflows

A.12.8. Target System Wizard - Connectivity Scenario

This dialog is displayed when you use the wizard for the first time in a new domain (or if your Connectivity scenario is not named correctly - its name must be identical to the domain). Supply the information for the Identity Store and all of the necessary maintenance workflows.

Select the workflows that you want to include into your scenario. The wizard will create copies of all selected workflows.

The wizard creates a scenario with a default layout and then creates the objects. After the scenario is created, you can move the objects to your preferred location.

You can subsequently add more workflows to your scenario using the Connectivity or the Provisioning view group.

Related Topics

- Overview
- Type Selection
- General Properties
- Target System Advanced
- Target System Timing
- Target System Options
- Connectivity Configuration
- Identity Store
- Associated Connected Directory
- Connected Directory Configuration
- Provisioning Workflows

A.12.9. Target System Wizard - Associated Connected Directory

Select the corresponding connected directory to be used as the template for your target system.

There are two options:

- **Show preferred** - displays the standard template for the previously selected target system type.
- **Show all** - displays the list of all available connected directories. Use, for example, a previously customized connected directory as template for your new one.

Select one of the templates from the list.

Related Topics

- Overview
- Type Selection
- General Properties
- Target System Advanced
- Target System Timing

- Target System Options
- Connectivity Configuration
- Identity Store
- Connectivity Scenario
- Connected Directory Configuration
- Provisioning Workflows

A.12.10. Target System Wizard - Connected Directory Configuration

Set the data required for accessing the new target system's connected directory:

Identity Store - information for the target system's representation in the Identity Store.

- **Account Base** (read-only) - displays the path in the Identity Store where accounts are stored for the target system.
- **Group Base** (read-only) - displays the path in the Identity Store where groups are stored for the target system.

Service - information for accessing the target system (not all target systems provide all options).

- **IP Address** - the server address to access the target system.
- **Data Port** - the data port to access the target system.

Bind Profile - information on how to bind to the target system (not all target systems provide all options). **User - the account to be used to bind. Password** - the password to be used to bind.

Connected Directory Type Specific - target system-specific information.

For a **Dashboard** target system:

- No specific information is required.

For an **Exchange 5.5** target system:

- **User Base Relative** - the domain-relative part of the user directory tree in the target system. * Domain user base-relative is the complete user directory tree.
- **Group Base Relative** - the domain-relative part of the group directory tree in the target system. * Domain group base-relative is the complete group directory tree.
- **Domain** - the domain name of the target system. The name usually consists of the company name and the exchange organization, like ou=exchorg,o=atos.

For an **LDAP** target system:

- **User Base** - the location (distinguished name) at which the accounts reside in the target system.

- **Group Base** - the location (distinguished name) at which the groups reside in the target system.

For a **Notes** target system:

- **User Base** - the location (distinguished name) at which the accounts reside in the target system.
- **Group Base** - the location (distinguished name) at which the groups reside in the target system.

For a **RACF** target system:

- **User Base** - the location (distinguished name) at which the accounts reside in the target system.
- **Group Base** - the location (distinguished name) at which the groups reside in the target system.

For a **SAP ECC UM** target system:

- No specific information is required.

For a **SAP NetWeaver UM** target system:

- **User Base** - the search base used in the SAP NetWeaver UM agent search request files for exporting users.
- **Synchronize Service Users** - whether (checked) or not (unchecked) to synchronize service users.
- **Group Base** - the search base used in the SAP NetWeaver UM agent search request files for exporting roles.

For a **SiPass** target system:

- No specific information is required.

For a **Windows 2008 / 2012** target system:

- **User Base Relative** - the domain-relative part of the user directory tree in the target system. * Domain user base relative is the complete user directory tree.
- **Group Base Relative** - the domain-relative part of the group directory tree in the target system. * Domain group base relative is the complete group directory tree.
- **Domain** - the domain name of the target system. The name consists of all domain components like dc=domain1,dc=munich,dc=atos,dc=de.
- **UPN Extension** - the extension of the user principle name. The attribute userPrincipalName, which is used as the login name, is built by the users samAccountName and this UPN extension.

Related Topics

- Overview

- Type Selection
- General Properties
- Target System Advanced
- Target System Timing
- Target System Options
- Connectivity Configuration
- Identity Store
- Connectivity Scenario
- Associated Connected Directory
- Provisioning Workflows

A.12.11. Target System Wizard - Provisioning Workflows

Select the types of provisioning workflows you'd like to configure for this target system.

The first two entries **Synchronization Workflow** and **Validation Workflow** are only present if Tcl-based workflow templates are available for this type of target system.

The rest of the list is dynamic and represents all available Java-based workflow templates that exist for this target system type. Select all workflows you need for your provisioning scenario.

You can subsequently add more workflows to your scenario using the Connectivity or the Provisioning view group.

Related Topics

- Overview
- Type Selection
- General Properties
- Target System Advanced
- Target System Timing
- Target System Options
- Connectivity Configuration
- Identity Store
- Connectivity Scenario
- Associated Connected Directory
- Connected Directory Configuration

A.13. Message Item

A message item keeps a set of messages for a specific language. Properties include:

General

- **Name** - the name of the object; in this case, the language this item represents.
- **Description** - a description of the object.
- **Locale** - the full locale definition (for example, en_US_Ca).
- **Language** - the language of the locale (for example en).
- **County** - the country of the locale (for example, US).
- **Variant** - the variant of the locale (for example, Ca).

Messages

- **Messages** - the list of messages for this locale.

Related Topics

- Message Item Folder

Subsection "Nationalization" in section "Understanding Request Workflows" in chapter "Understanding the Default Application Workflow Technology" in the DirX Identity Application Development Guide.

A.13.1. Message Item Folder

A folder that can keep several message items that each reflect a different language.

- **Name** - the name of the object.
- **Description** - a description of the object.

Related Topics

- Message Item

Subsection "Nationalization" in section "Understanding Request Workflows" in chapter "Understanding the Default Application Workflow Technology" in the DirX Identity Application Development Guide.

DirX Product Suite

The DirX product suite provides the basis for fully integrated identity and access management; it includes the following products, which can be ordered separately.



DirX Identity

DirX Identity provides a comprehensive, process-driven, customizable, cloud-enabled, scalable, and highly available identity management solution for businesses and organizations. It provides overarching, risk-based identity and access governance functionality seamlessly integrated with automated provisioning. Functionality includes lifecycle management for users and roles, cross-platform and rule-based real-time provisioning, web-based self-service functions for users, delegated administration, request workflows, access certification, password management, metadirectory as well as auditing and reporting functionality.



DirX Directory

DirX Directory provides a standards-compliant, high-performance, highly available, highly reliable, highly scalable, and secure LDAP and X.500 Directory Server and LDAP Proxy with very high linear scalability. DirX Directory can serve as an identity store for employees, customers, partners, subscribers, and other IoT entities. It can also serve as a provisioning, access management and metadirectory repository, to provide a single point of access to the information within disparate and heterogeneous directories available in an enterprise network or cloud environment for user management and provisioning.



DirX Access

DirX Access is a comprehensive, cloud-ready, scalable, and highly available access management solution providing policy- and risk-based authentication, authorization based on XACML and federation for Web applications and services. DirX Access delivers single sign-on, versatile authentication including FIDO, identity federation based on SAML, OAuth and OpenID Connect, just-in-time provisioning, entitlement management and policy enforcement for applications and services in the cloud or on-premises.



DirX Audit

DirX Audit provides auditors, security compliance officers and audit administrators with analytical insight and transparency for identity and access. Based on historical identity data and recorded events from the identity and access management processes, DirX Audit allows answering the “what, when, where, who and why” questions of user access and entitlements. DirX Audit features historical views and reports on identity data, a graphical dashboard with drill-down into individual events, an analysis view for filtering, evaluating, correlating, and reviewing of identity-related events and job management for report generation.

For more information: support.dirx.solutions/about

Eviden is a registered trademark © Copyright 2026, Atos Group – All rights reserved.

Legal remarks

On the account of certain regional limitations of sales rights and service availability, we cannot guarantee that all products included in this document are available through the Eviden sales organization worldwide. Availability and packaging may vary by country and is subject to change without prior notice. Some/All of the features and products described herein may not be available locally. The information in this document contains general technical descriptions of specifications and options as well as standard and optional features which do not always have to be present in individual cases. Eviden reserves the right to modify the design, packaging, specifications and options described herein without prior notice. Please contact your local Eviden sales representative for the most current information. Note: Any technical data contained in this document may vary within defined tolerances. Original images always lose a certain amount of detail when reproduced.