EVIDEN

Identity and Access Management

# DirX Identity

## Tutorial

Version 9.0.0, Edition July 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. Getting Started 6
1.1. Prerequisites 6
1.2. Logging In 7
1.3. Preparing to Use the Quick Start 7
1.3.1. Loading the Connectivity Scenario 8
1.3.2. Creating a Sample Connected System Node 8
1.3.2.1. Creating a New Context Prefix 8
1.3.2.2. Checking the New Context Prefix 8
1.3.3. Adding a Certificate for Encryption 9
1.3.4. Adding a Certificate for Signed Auditing (Optional) 10
1.3.5. Enabling Auditing (Optional) 10
1.3.6. Setting up Email Notification (Optional) 11
1.3.7. Setting up Menu Policies (Optional) 12
1.3.8. Restarting the Servers 12
1.3.9. Setting up the Portal Target Systems 13
1.4. Working with the Quick Start 13
1.4.1. Backing up the Database 13
1.4.2. Checking the Database 14
1.5. User Self-Registration 14
1.5.1. Requesting the Services 14
1.5.1.1. Starting the Self-Registration Process 15
1.5.1.2. Providing User Data 15
1.5.1.3. Entering a Password 16
1.5.1.4. Selecting the Services 16
1.5.1.5. Confirming the User Data 16
1.5.2. Monitoring the Self-Registration Request 17
1.5.3. Approving the Self-Registration Request 17
1.5.3.1. Approving without Email Notification 18
1.5.3.2. Approving with Email Notification 18
1.5.3.3. Approving the User Creation Request 18
1.5.3.4. Checking the Result of User Creation Approval 19
1.5.4. Approving the Hardware Beta Programs Request 20
1.5.4.1. Monitoring the Hardware Beta Programs Approval Workflow 20
1.5.4.2. Approving the Hardware Beta Programs Request 20
1.5.5. Using the New Account and Password 21

1.5.5.1. Logging In without Email Notification . . . 21
1.5.5.2. Logging In with Email Notification . . . 21
1.5.5.3. Viewing the New User . . . 22
1.6. Adding a New User . . . 22
1.6.1. Viewing Users . . . 23
1.6.1.1. Checking Auffret Jean-Marc . . . 23
1.6.1.2. Checking Bellanger Lionel . . . 24
1.6.1.3. Checking Blander Dyan . . . 24
1.6.1.4. Checking Blander's Privileges . . . 24
1.6.1.5. Checking Blander's Accounts . . . 25
1.6.2. Adding the User . . . 25
1.6.3. Assigning Privileges by Hand . . . 26
1.6.4. Adding a User Password . . . 27
1.6.5. Approving the User's Privileges . . . 27
1.6.5.1. Checking the Approval Workflow . . . 27
1.6.5.2. Approving the Request . . . 28
1.6.5.3. Checking Approval with Web Center . . . 29
1.6.5.4. Checking Approval with DirX Identity Manager . . . 29
1.6.5.5. Checking Approval with Web Admin . . . 30
1.6.5.6. Continuing the Approval Process . . . 30
1.6.5.7. Checking the Results of the Approval Process . . . 30
1.6.5.7.1. Re-Checking the Workflow's Structure . . . 30
1.6.5.7.2. Re-Checking User Teacher's Privileges . . . 31
1.6.6. Adding the User to a Project . . . 32
1.6.6.1. Assigning the Role with Role Parameters . . . 32
1.6.6.2. Checking the Result of the Role Assignment . . . 33
1.6.7. Assigning Privileges Automatically . . . 33
1.6.7.1. Rule-based Provisioning . . . 34
1.6.7.1.1. Checking the Contractor Provisioning Rules . . . 34
1.6.7.1.2. How Event-based Provisioning Works . . . 34
1.6.7.1.3. Configuring Event-based Provisioning . . . 35
1.6.7.1.4. Using Rule-based Provisioning . . . 36
1.6.7.2. Business Object Inheritance . . . 37
1.6.7.2.1. Viewing Business Objects . . . 37
1.6.7.2.2. Assigning an Additional Role . . . 37
1.6.7.3. Using Permission and Role Parameters . . . 38
1.6.7.3.1. Checking the Sales Tasks Privilege Hierarchy . . . 38
1.6.7.3.2. Checking the Sales Tasks Permission . . . 39
1.6.7.3.3. Checking the FS Sales Group . . . 39
1.7. Importing Identities . . . 40
1.7.1. Viewing the Default Connectivity Scenario . . . 40
1.7.1.1. Viewing Connected Directories and Workflows . . . 41

1.7.1.2. Viewing a Workflow's Structure with the Global View . . . 41
1.7.1.3. Viewing a Workflow's Structure with a Report . . . 42
1.7.2. Creating a New Scenario . . . 42
1.7.2.1. The Main Scenario . . . 43
1.7.2.2. Creating the NewCompany Scenario . . . 43
1.7.3. Adding a New Identity Source . . . 43
1.7.3.1. Setting up the ODBC Database . . . 43
1.7.3.1.1. Copying the PreDefined ODBC Database . . . 44
1.7.3.1.2. Creating the ODBC Data Source . . . 44
1.7.3.2. Assigning the Identity Store . . . 44
1.7.3.3. Creating the New Connected Directory . . . 45
1.7.3.4. Using the Configuration Wizards . . . 45
1.7.3.5. Configuring the New Connected Directory . . . 46
1.7.3.5.1. Selecting a Connected Directory Template . . . 46
1.7.3.5.2. Supplying General Information . . . 46
1.7.3.5.3. Updating the Attribute Configuration . . . 47
1.7.3.5.4. Checking the Bind Parameters . . . 47
1.7.3.5.5. Supplying Operational Attributes . . . 48
1.7.3.5.6. Naming the Connected Directory . . . 48
1.7.3.6. Checking the Database Viewer . . . 48
1.7.4. Creating a New Workflow . . . 49
1.7.5. Configuring the New Workflow . . . 49
1.7.5.1. Selecting the Workflow Template . . . 50
1.7.5.2. Supplying Workflow General Information . . . 50
1.7.5.3. Selecting the Attributes to be Exported . . . 50
1.7.5.4. Selecting the Mapping Attributes . . . 51
1.7.5.5. Configuring the Attribute Mapping . . . 51
1.7.5.6. Setting the Export Properties . . . 53
1.7.5.7. Setting Delta Handling . . . 53
1.7.5.8. Setting the Export Trace Parameters . . . 53
1.7.5.9. Setting Operational Parameters . . . 53
1.7.5.10. Setting Import Properties . . . 54
1.7.5.11. Setting Entry-Handling Properties . . . 54
1.7.5.12. Setting Import Tracing Parameters . . . 55
1.7.5.13. Naming the Workflow . . . 55
1.7.6. Running the New Workflow . . . 55
1.7.6.1. Checking Product Testing Users . . . 55
1.7.6.2. Running the Workflow . . . 56
1.7.7. Monitoring the Workflow Run . . . 57
1.7.7.1. Monitoring from the Structure Tab . . . 57
1.7.7.1.1. Viewing Activity Status Data . . . 58
1.7.7.1.2. Viewing Input/Output Data . . . 58

1.7.7.1.3. Viewing Trace Information . . . 59
1.7.7.1.4. Viewing Statistics . . . 59
1.7.7.2. Monitoring from the Monitor View . . . 59
1.7.7.3. Cleaning up Workflow Status Entries . . . 60
1.7.8. Analyzing the Result . . . 61
1.7.8.1. Viewing the Data Entries . . . 61
1.7.8.2. Viewing the Permissions . . . 61
1.7.8.3. How Automatic Assignment Works . . . 62
1.8. Changing the Workflow Configuration . . . 63
1.8.1. Checking User Bader's Attributes . . . 63
1.8.2. Changing the Workflow's Parameters . . . 63
1.8.2.1. Changing the Source Selected Attributes . . . 63
1.8.2.2. Changing the Target Selected Attributes . . . 63
1.8.2.3. Changing the Attribute Mapping . . . 63
1.8.3. Running the Reconfigured Workflow . . . 64
1.8.4. Re-Checking User Bader . . . 64
1.9. Setting up a New Target System . . . 64
1.9.1. Installing the Connected System . . . 65
1.9.1.1. Creating the Connected System . . . 65
1.9.1.2. Checking the New Connected System . . . 65
1.9.2. Adding the Target System . . . 65
1.9.2.1. Configuring the Target System . . . 66
1.9.2.1.1. Selecting a Target System Template . . . 66
1.9.2.1.2. Specifying a Name and an Administrator . . . 66
1.9.2.1.3. Specifying a Cluster and a Domain . . . 66
1.9.2.1.4. Configuring the Account and Group Roots . . . 66
1.9.2.1.5. Selecting the Connectivity Scenario . . . 66
1.9.2.1.6. Selecting the Associated Connected Directory . . . 67
1.9.2.1.7. Specifying Connected Directory Configuration Information . . . 67
1.9.2.1.8. Selecting the Provisioning Workflow Types . . . 67
1.9.2.2. Checking the New Target System . . . 67
1.9.3. Loading the Accounts and Groups . . . 68
1.9.3.1. Running the Validation Workflow . . . 68
1.9.3.2. Checking the New Entries in the Target System . . . 68
1.9.3.3. Checking for Unassigned Accounts . . . 69
1.9.3.4. Checking the Imported Groups . . . 69
1.9.4. Joining Accounts to Users . . . 69
1.9.4.1. Creating a Consistency Rule for Joining Accounts to Users . . . 69
1.9.4.2. Creating a Consistency Workflow that Uses the New Rule . . . 70
1.9.4.3. Assigning the Consistency Workflow to the New-LDAP Target System . . . 70
1.9.4.4. Running the Consistency Workflow . . . 71
1.9.4.5. Checking the Results of the Consistency Workflow . . . 71

1.9.4.6. Resolving the Unassigned Accounts . . . 71
1.9.5. Integrating the Groups into the Privilege Structure . . . 72
1.9.5.1. Assigning the Software Tests Group to the Test Tasks Permission . . . 72
1.9.5.2. Checking the Results of the Group-Permission Assignment . . . 72
1.9.5.3. Creating a New Permission "Firmware Tests" . . . 72
1.9.5.4. Creating a New Role "Firmware Tests" . . . 73
1.9.5.5. Assigning the Firmware Tests Role . . . 73
1.9.5.6. Checking the Group Integration Results . . . 73
1.9.6. Synchronizing the Target System . . . 73
1.9.6.1. Running the Synchronization Workflow . . . 74
1.9.6.2. Checking the Results of the Synchronization Workflow . . . 74
1.9.7. Validating the Target System . . . 74
1.9.7.1. Changing Some Account and Group Data in the Target System . . . 74
1.9.7.2. Running the Validation Workflow . . . 75
1.9.7.3. Checking the Results of the Validation Workflow . . . 75
1.9.7.4. Synchronizing the Changes in the Target System . . . 75
1.10. Using Password Management . . . 76
1.10.1. Preparing for Password Management . . . 76
1.10.1.1. Adding the New-LDAP Resource Family . . . 76
1.10.1.2. Restarting the Java-based Identity Server . . . 77
1.10.1.3. Checking the Java-based Identity Server State . . . 77
1.10.2. Changing Passwords . . . 77
1.10.2.1. Ensuring that Password Management is Enabled . . . 78
1.10.2.2. Changing User Duplan's Password . . . 78
1.10.3. Viewing the Effects of Password Management . . . 78
1.10.3.1. Viewing the Effects with the Monitor View . . . 79
1.10.3.1.1. Checking the Results at the Password Event Manager . . . 79
1.10.3.1.2. Checking the Result at the Intranet Portal Target System . . . 79
1.10.3.2. Viewing the Effects with Web Admin . . . 79
1.10.3.2.1. Checking the Password Change Listener Adaptor . . . 80
1.10.3.2.2. Checking the SetAccountPasswordListener Adaptor . . . 80
1.10.3.2.3. Checking the Dead Letter Queue Adaptor . . . 80
1.10.3.2.4. Viewing the Server Log Files . . . 80
2. Follow-on Tutorials . . . 81
2.1. Reusing the Sample Domain . . . 81
2.1.1. Configuring the Customer Domain . . . 81
2.1.1.1. Creating the New Domain . . . 82
2.1.1.2. Checking the New Domain . . . 82
2.1.2. Exporting Objects from the Sample Domain . . . 83
2.1.2.1. Enabling Full-Text LDIF Output . . . 83
2.1.2.2. Copying the Default Policies and Workflows Collection . . . 83
2.1.2.3. Exporting the Default Collection . . . 83

2.1.3. Importing Objects to the Customer Domain . . . 83
2.1.3.1. Editing the Exported Collection File . . . 83
2.1.3.2. Importing the Modified Collection File . . . 84
2.1.4. Adapting the Imported Objects . . . 84
2.1.4.1. Adapting Access Policies . . . 84
2.1.4.2. Adapting Request and Approval Workflows . . . 84
2.1.4.3. Adapting Provisioning Rules . . . 85
2.2. Integrating a Customer-Specific Agent . . . 85
2.2.1. Understanding the Agent Integration Framework . . . 86
2.2.1.1. Understanding Simple Execution Integration . . . 86
2.2.1.2. Understanding Central Edit Integration . . . 87
2.2.1.3. Understanding Detailed Customization . . . 87
2.2.2. Integrating a Connectivity Scenario . . . 87
2.2.2.1. Understanding the Sample Connectivity Scenario . . . 88
2.2.2.1.1. Prerequisites for Integrating the Sample Connectivity Scenario . . . 88
2.2.2.1.2. Understanding the Sample Connected Directory . . . 88
2.2.2.1.3. Understanding the Sample Agent . . . 88
2.2.2.2. Planning the Integration Tasks . . . 89
2.2.2.2.1. Planning for Simple Execution (Level 1) . . . 89
2.2.2.2.2. Planning for Central Edit . . . 90
2.2.2.2.3. Planning for Detailed Customization . . . 90
2.2.2.2.4. Entering Connectivity Configuration Data . . . 91
2.2.2.3. Integrating to Simple Execution (Level 1) . . . 91
2.2.2.3.1. Setting up the Central Configuration Object . . . 91
2.2.2.3.2. Setting up the Workflow MyWorkflow . . . 92
2.2.2.3.3. Setting up the Agent MyAgent . . . 94
2.2.2.3.4. Setting up the Connected Directory MyDataBase . . . 95
2.2.2.3.5. Setting up a Channel to MyDatabase . . . 96
2.2.2.3.6. Setting up the Job MyJob . . . 96
2.2.2.3.7. Completing the Configuration . . . 97
2.2.2.3.8. Test the Newly Created Workflow . . . 98
2.2.2.4. Integrating to Central Edit (Level 2) . . . 99
2.2.2.4.1. Registering and Importing the INI File for MyJob . . . 100
2.2.2.4.2. Registering the Trace File for MyJob . . . 101
2.2.2.4.3. Modifying the Job MyJob . . . 102
2.2.2.4.4. Editing the INI File and Testing the Modified Workflow . . . 102
2.2.2.5. Integrating to Detailed Customization (Level 3) . . . 103
2.2.2.5.1. Creating the XML Job Description . . . 104
2.2.2.5.2. Defining the References in the INI File . . . 106
2.2.2.5.3. Changing a Property and Testing the Modified Workflow . . . 107
2.2.2.5.4. Creating a Workflow Wizard . . . 108
2.2.2.5.5. Testing the Workflow Wizard . . . 109

2.2.2.5.6. Running the Workflow Again . . . 109
2.2.2.5.7. Exporting Your Scenario . . . 110
2.3. Maintaining the Privilege Structure . . . 111
2.3.1. Checking the Privilege Structure . . . 111
2.3.1.1. Checking My-Company's Department-Specific Roles . . . 111
2.3.1.2. Checking My-Company's Group File Share Permission . . . 112
2.3.1.3. Checking My-Company's Sales Tasks Permission . . . 112
2.3.1.4. Identifying the Maintenance Tasks . . . 113
2.3.2. Correcting the Problems with the Privilege Structure . . . 113
2.3.2.1. Correcting the Sales Tasks Permission . . . 113
2.3.2.2. Creating a File Share Privilege Structure . . . 113
2.3.2.2.1. Creating the Groups . . . 114
2.3.2.2.2. Adding the New Groups to the Group File Share Permission . . . 114
2.3.2.2.3. Copying the Sales Tasks Permission and Role . . . 115
2.3.2.2.4. Creating the Provisioning Rule . . . 115
2.3.2.2.5. Running the Policy Execution Service . . . 116
2.3.2.2.6. Changing the Professional Services Tasks Permission . . . 116
2.4. Changing a Workflow's Structure . . . 117
2.4.1. Understanding a Workflow's Structure . . . 117
2.4.2. Changing a Job's Timeout Value . . . 118
2.5. Applying SoD Policies . . . 118
2.5.1. Activating SoD Checking . . . 118
2.5.2. Activating an SoD Policy . . . 119
2.5.3. Assigning a Conflicting Privilege . . . 119
2.5.4. Checking the SoD Violation . . . 120
2.5.4.1. Checking the SoD Mitigation Workflow . . . 120
2.5.4.2. Checking the User's SoD Information . . . 120
2.5.5. Overriding the SoD Violation . . . 121
2.5.6. Checking the Effect of the SoD Violation Exception . . . 121
2.5.6.1. Checking the Effect on the SoD Mitigation Workflow . . . 121
2.5.6.2. Checking the Effect on the User's Privileges . . . 121
2.5.6.3. Checking the Effect on the SoD Policy . . . 122
2.6. Re-approving a Privilege Assignment . . . 122
2.6.1. Understanding How Re-approval Works . . . 122
2.6.2. Selecting a Privilege for Re-approval . . . 123
2.6.3. Identifying the Privilege's Users . . . 123
2.6.4. Initializing the Re-approval Process . . . 124
2.6.5. Running the Re-approval Workflow . . . 125
2.6.6. Checking the Re-approval Workflow Results . . . 125
2.7. Certifying a Role . . . 126
2.7.1. Understanding How a Privilege Certification Campaign Works . . . 127
2.7.2. Configuring the Certification Campaign . . . 127

2.7.2.1. Configuring the SMTP Service . . . 127
2.7.2.2. Scheduling the Certification Campaign Controller . . . 127
2.7.2.3. Configuring the Java-based Server for Certification Campaigns . . . 128
2.7.2.4. Configuring the Certification Campaign Controller Workflow . . . 128
2.7.3. Creating the Privilege Certification Campaign . . . 128
2.7.4. Starting the Certification Campaign . . . 130
2.7.5. Certifying the Privilege . . . 130
2.7.5.1. Certifying the Privilege with DirX Identity Web Center . . . 130
2.7.5.2. Certifying the Privileges with DirX Identity Business User Interface . . . 130
2.7.6. Monitoring the Campaign . . . 131
2.7.7. Finishing the Campaign . . . 131
2.8. Certifying a User . . . 131
2.8.1. Understanding How a Certification Campaign Works . . . 132
2.8.2. Configuring the Certification Campaign . . . 132
2.8.3. Creating the User Certification Campaign . . . 133
2.8.4. Starting the Certification Campaign . . . 135
2.8.5. Monitoring the Certification Campaign . . . 135
2.8.6. Certifying the Users . . . 137
2.8.6.1. Certifying the Users with DirX Identity Web Center . . . 137
2.8.6.2. Certifying the Users with DirX Identity Business User Interface . . . 138
2.8.7. Finishing the Campaign . . . 140
2.8.8. Generating a Certification Campaign Report . . . 141
2.8.9. Deleting a Certification Campaign . . . 142
2.9. Applying Attribute Modification Approval . . . 142
2.9.1. Activating Attribute Modification Approval Checking . . . 143
2.9.2. Modifying an Attribute Modification Approval Policy . . . 143
2.9.3. Modifying the User Attributes . . . 143
2.9.4. Monitoring the Attribute Approval Process . . . 144
2.9.5. Approving the Attribute Modification Request . . . 145
2.9.6. Checking the Result of Attribute Modification Approval . . . 145
2.10. Scheduled Privilege Assignment . . . 146
2.10.1. Disabling Event-based Privilege Resolution . . . 146
2.10.2. Performing a Pure LDAP Change . . . 147
2.10.3. Configuring the Policy Execution Service . . . 147
2.10.4. Running the Policy Execution Service . . . 148
2.10.5. Using the Structure Tab to Check the Results . . . 148
2.10.6. Checking Ruben Briner's Privileges . . . 148
2.10.7. Running the Privilege Resolution Service . . . 148
2.10.8. Re-Checking Ruben Briner's Privileges . . . 149
2.11. Creating a Nested Workflow . . . 149
2.11.1. Copying and Moving a Default Workflow . . . 149
2.11.2. Reconfiguring the Default Workflow's Activities . . . 150

2.11.3. Running and Monitoring the Nested Workflow . . . 150
2.11.4. Making the Nested Workflow Available in the Global View . . . 150
2.12. Using Manual Provisioning . . . 150
2.12.1. Understanding Manual Provisioning . . . 151
2.12.2. Viewing the Scenario . . . 151
2.12.2.1. Viewing the Target System . . . 151
2.12.2.2. Viewing the Privilege Structure . . . 152
2.12.2.3. Viewing the Rules . . . 152
2.12.3. Requesting Physical Access . . . 153
2.12.3.1. Requesting Access to the Munich Archive . . . 153
2.12.3.2. Approving the Access Request . . . 153
2.12.4. Checking the Approval Result . . . 153
2.12.4.1. Checking the Request Workflow Result . . . 154
2.12.4.2. Viewing the New Account and Group Membership . . . 154
2.12.4.3. Viewing the Real Time Workflow . . . 154
2.12.4.4. Viewing the Request Workflow Instance . . . 154
2.12.5. Performing Manual Provisioning . . . 155
2.12.6. Viewing the Result . . . 156
2.12.6.1. Viewing the Provisioned User . . . 156
2.12.6.2. Viewing as the Administrator . . . 156
2.12.6.3. Checking Statistics on Groups . . . 156
2.13. Working with Internal Tickets . . . 157
2.13.1. Understanding How to Work with Internal Tickets . . . 157
2.13.2. Creating a Modification Ticket . . . 157
2.13.3. Viewing the Modification Ticket . . . 158
2.13.4. Processing the Modification Ticket . . . 159
2.13.5. Viewing the Modification Ticket Result . . . 159
2.14. Working with Source Tickets . . . 160
2.14.1. Understanding How to Use Source Tickets . . . 161
2.14.2. Preparing the Web Service Environment . . . 161
2.14.2.1. Starting the Web Service as Stand-alone Application . . . 162
2.14.2.2. Deploying the Web Service into Tomcat . . . 162
2.14.3. Preparing the Sample Web Service Client . . . 162
2.14.4. Viewing the Ticket Request . . . 162
2.14.5. Sending the Ticket . . . 163
2.14.6. Watching the Ticket Workflow . . . 164
2.14.7. Checking the Ticket Status . . . 165
2.14.8. Checking the Ticket Status by Request Identifier . . . 166
2.14.9. Approving the Manager Analyst Relations Role . . . 167
2.14.10. Viewing the Request Workflows . . . 167
2.14.11. Checking the Final Ticket State . . . 168
2.15. Managing Personas . . . 168

2.15.1. Enabling Persona Management . . . 168
2.15.2. Handling Personas in Web Center . . . 169
2.15.2.1. Creating a User's Main Identity . . . 169
2.15.2.2. Verifying John Smith's Main Identity . . . 169
2.15.2.3. Creating a Persona for John Smith . . . 169
2.15.2.4. Viewing John Smith's Main Identity and Persona . . . 170
2.15.3. Preparing the Persona Environment . . . 171
2.15.3.1. Modifying the PersonaCommon.xml Object Description . . . 171
2.15.3.2. Modifying the PersonaFromUser.xml Object Description . . . 172
2.15.3.3. Modifying the Java Script CommonNameForPersona.js . . . 172
2.15.3.4. Modifying defaultRenderer.properties . . . 173
2.15.3.5. Verifying the Updated Persona Environment . . . 174
2.15.4. Creating a New Persona for an Imported Non-Primary Account . . . 174
2.15.4.1. Preparing the Tutorial . . . 175
2.15.4.2. Verifying John Smith's Accounts . . . 175
2.15.4.3. Modifying the Consistency Rule CreatePersonasForNonPrimaryAccounts . . . 176
2.15.4.4. Modifying the Java Script AccountToPersonas.js . . . 177
2.15.4.5. Running the Consistency Rule CreatePersonasForNonPrimaryAccounts . . . 177
2.15.4.6. Approving the Persona Create Workflow . . . 178
2.15.4.7. Verifying the New Persona and the Main Identity . . . 178
2.16. Managing Functional Users . . . 178
2.16.1. Enabling Functional User Management . . . 179
2.16.2. Preparing the Functional Users Environment . . . 179
2.16.2.1. Creating the Workflow Create Trainee for Department . . . 180
2.16.2.2. Creating the Object Description TraineeFromUser . . . 180
2.16.3. Creating a New Trainee for the Department . . . 182
2.17. Using Risk Management . . . 183
2.17.1. Activating Risk Management . . . 183
2.17.2. Configuring Risk Values for Target Systems and Groups . . . 184
2.17.3. Checking User Risk Parameters . . . 185
2.17.4. Changing a Target System Weight . . . 185
2.17.5. Changing the Risk Limits in the Risk Policy . . . 186
2.17.6. Scheduling the Risk Calculation Workflow . . . 187
2.17.7. Configuring the Risk Approval Workflow . . . 187
3. About the My-Company Sample Domain . . . 191
3.1. Logging In . . . 191
3.2. Users . . . 191
3.2.1. My-Company . . . 192
3.2.2. Suppliers . . . 193
3.2.3. Customers . . . 193
3.2.4. User Properties . . . 193
3.2.4.1. General User Properties . . . 194

3.2.4.2. Relationships to Other Users . . . 195
3.2.4.3. Operational Information . . . 196
3.2.4.4. Communication Information . . . 196
3.2.4.5. Authentication Information . . . 197
3.2.4.6. Links to Organizations . . . 197
3.2.4.7. Links to Locations . . . 197
3.2.4.8. Links to Contexts . . . 198
3.2.4.9. User Privilege Assignments . . . 198
3.2.4.10. Account Ownerships . . . 199
3.2.4.11. Order Information . . . 199
3.2.4.12. SoD Exceptions . . . 199
3.3. Business Objects . . . 199
3.3.1. Companies . . . 200
3.3.2. Countries . . . 202
3.3.3. Projects . . . 203
3.4. Privileges . . . 204
3.4.1. Roles . . . 204
3.4.2. Permissions . . . 207
3.4.3. Groups . . . 207
3.5. Policies . . . 208
3.5.1. Access Policies . . . 208
3.5.2. Attribute Policies . . . 215
3.5.3. Delete Policies . . . 216
3.5.4. Event Policies . . . 216
3.5.5. Password Policies . . . 216
3.5.6. Operations . . . 217
3.5.7. Rules . . . 217
3.5.8. SoD Policies . . . 218
3.6. Request Workflows . . . 219
3.7. Target Systems . . . 220
3.8. Auditing . . . 222
3.8.1. Status Reports . . . 222
3.8.2. Audit Trail . . . 223
3.9. Domain Configuration . . . 223
3.9.1. General Domain Parameters . . . 224
3.9.2. Policy Parameters . . . 224
3.9.3. Timing Parameters . . . 224
3.9.4. Permission Parameters . . . 224
3.9.5. Privilege Resolution Parameters . . . 225
3.9.6. Request Workflow Parameters . . . 225
3.9.7. Domain Compliance Parameters . . . 226
3.10. Project Organization . . . 226

Legal Remarks . . . . . 229

# Preface

This manual provides tutorial information that helps to understand and using the features of DirX Identity. It consists of the following sections:

- ch1_gettingStarted.pdf provides getting started information that uses a step-by-step approach to describe how to work with the product.
- ch2_tutorials.pdf provides information how to use more sophisticated features of the product.
- ch3_sampleDomain.pdf provides a description of the sample domain and the default applications delivered with DirX Identity.

# DirX Identity Documentation Set

**Version: 9.0.0 | Build: 29615 | Date: 2026-07-14**

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. Getting Started

This quick start demonstrates the most important features of DirX Identity and illustrates the typical way to work with DirX Identity in a customer environment. It consists of several nearly independent sections that describe DirX Identity provisioning and connectivity administrative procedures.

The quick start scenario provided here is based on the My-Company sample domain, the My-Company connectivity scenario, and the default connectivity applications - referred to in this quick start as the "default connectivity scenario" - that are delivered with DirX Identity. We recommend that you read the **"Users"** and **"Target Systems"** sections in "About the Sample Domain" to become familiar with the My-Company sample domain's structure and its objects before you proceed with the quick start.

This quick start consists of the following sections:

- Prerequisites provides the prerequisites you must fulfill before you can start to work.
- Logging In provides information how to log in to DirX identity Manager.
- Preparing to Use the Quick Start provides information how to set up initial sample data.
- Working with the Quick Start provides information how to back up the database and check the database.
- User Self-Registration explains how to use DirX Identity's self-registration feature to add a single identity - a new user - to the My-Company sample domain and automatically provision the new user in the My-Company domain according to the requests for My-Company services made during user self-registration.
- Adding a New User explains how to add a new user to the My-Company sample domain by hand and how to provision the new user in the My-Company domain, both by hand and automatically.
- Importing Identities shows how to set up a workflow that imports multiple identities from an external source into the My-Company identity store and automatically provision them in the My-Company domain.
- Setting up a New Target System shows how to add a new target system to the My-Company scenario.
- Using Password Management shows how to set up a password management solution so that My-Company users can change their own passwords

After performing these quick start sections, you should be able to:

- Understand and use most of DirX Identity's powerful features
- Set up customer-specific solutions quickly using DirX Identity concepts rather than creating your own concepts, which might require a lot of effort

## 1.1. Prerequisites

Before you can use the quick start, you must:

- **Install DirX Directory** and its My-Company example database. **Install DirX Identity Professional Suite** (including the Business Suite) - ensure that **Business and Pro Suite Upgrade** and all of its selectable components - on the same machine (no distributed environment) - are checked when installing DirX Identity. The tutorial exercises aren't compatible with the DirX Identity Business Suite.
- Execute the Configuration Wizard. Select the **Sample domain** option in the Domain Configuration step.

## 1.2. Logging In

To log in to the DirX Identity Manager:

- Select **Start** → **All Programs/Apps** → **DirX Identity V***n.n* → **Manager**.
- Click **Provisioning**.
- In **Server**, 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 **Server**. Be sure that **User DN** is set to **cn=DomainAdmin,cn=My-Company**. If not, select it from the list.
- Enter the password, and click **OK**. You can find the default credentials and passwords in the Default Credentials KeePass file.

After a few seconds, the DirX Identity Manager displays its views window.

## 1.3. Preparing to Use the Quick Start

Before we can follow our quick start, we must perform the following tasks:

- Load the connectivity scenario for the My-Company sample domain
- Create a sample connected system node
- Add a certificate to the identity store for encryption
- Add a certificate to the identity store for signed auditing (optional)
- Enable auditing (optional)

- Set up e-mail notification for the privilege approval process (optional)
- Set up menu policies (optional)
- Stop, then restart the DirX Identity and Apache Tomcat servers
- Set up the portal target systems

## 1.3.1. Loading the Connectivity Scenario

Each DirX Identity domain (here the sample domain My-Company) requires a corresponding DirX Identity connectivity scenario. This scenario is not automatically installed.

To install the pre-defined connectivity scenario:

- Log in to DirX Identity Manager.
- Select the **Connectivity** view group and enter the password **dirx** in the Log in dialog.
- Click the **Expert View** icon in the left pane to select it.
- Right-click the top level node **Connectivity Configuration Data**, and select **Import Data**.
- Choose the file *<DXI_INSTALL_PATH>***\data\extension\My-Company_scenario.ldif**.
- Click **Open** and wait until the scenario is loaded. (This action can take a few minutes.)
- Click **No** in the View Trace File dialog.
- Click **Global View**. A new scenario folder **My-Company** is displayed. Open it and click the **Main** scenario. Manager displays the target systems in the My-Company sample domain. If you have not done so already, we recommend that you read the "Target Systems" sections in "About the Sample Domain" to familiarize yourself with the target systems used in the My-Company sample domain.

## 1.3.2. Creating a Sample Connected System Node

We use an additional tree in an LDAP directory as the root node for several sample LDAP connected systems. We create a new context prefix **o=sample-ts** in the LDAP directory.

#### 1.3.2.1. Creating a New Context Prefix

First, we'll add the context prefix: **o=sample-ts** to the LDAP directory:

- In the file *<DXI_INSTALL_PATH>***basic.input.tcl**, enter the relevant passwords in the **`DIR_PW`** (Directory admin password) and **`DEMODOMAIN_PW`** (Sample domain admin password) fields.
- In *<DXI_INSTALL_PATH>***\data\schema\dirx**, run the script **Setup.bat**.
- Check the file **trace.txt** to make sure that everything ran correctly. You should see three occurrences of the exit code 0.

#### 1.3.2.2. Checking the New Context Prefix

Now we'll use DirX Identity Manager's data view to check that the tree **sample-ts** exists:

- Log in to DirX Identity Manager's data view (if not already logged in) and select **Data View** → **Connectivity**.
- Right-click the **data o=My-Company** tree, then select **Server...**. (Ensure that you do not perform this and the following steps for the **DirXmetahub** tree!)
- The "Server" properties window opens. Modify the server profile with these parameters:
  **Name**: **Sample-TS**
  **Default User DN**: **cn=domainadmin,cn=My-Company**
  **Base DN**: **o=sample-ts**
  Click **OK**.
- Now the node **Sample-TS o=sample-ts** is displayed instead **data o=My-Company** tree. That is the **data o=My-Company** tree has been re-configured to **Sample-TS**.

### 1.3.3. Adding a Certificate for Encryption

Some of the tutorial exercises use DirX Identity's encryption function, so we need to add a certificate to the identity store to be used for password encryption:

- Copy the **dirxgenpse**[**.exe**] tool from the folder **DirXIdentity\EncryptionTool** in the DirX Identity DVD's installation structure to a directory to which the PATH variable points; for example, to the DirX Identity installation's **bin** directory.

*do not* use the **dirxgenpse**[**.exe**] that is delivered and installed with DirX Directory.

- Run the following command in the MS/DOS command prompt or in a UNIX shell:
  `dirxgenpse -D cn=server_admin,dxmc=DirXmetahub -s 2 -P 1234 -w <password>`

The **-P** option specifies the PIN of the certificate. In this example, we use the pre-configured value **1234** (this value is already configured in the server *.**ini** files). If you use another PIN (and you should do this in production environments!), you must change the *.**ini** files to use the new PIN.

Since it is the first generated key pair, the serial number **2** is set with the **-s** option.

To verify the successful generation of the certificate:

- Start the DirX Identity Manager.
- Log in to Connectivity and go to **Data View** → **Connectivity** (Connectivity dxmc=DirXmetahub) → **cn=server_admin**.
- Open the All Attributes tab. You should see a **userCertificate** and a value for **keyOwnerPSE** containing ...: **cn=server-admin,**... .

To enable encryption (optional), follow these steps:

- In DirX Identity Manager, go to Connectivity. Select **Expert View** → **Connectivity Configuration Data** → **Configuration** → **Server** and then select **ATTRIB_ADMIN_PW-Attributes and Administrative Passwords** from **Encryption**. Save it.

- Set **encryptionMode** to **1** in *<DXI_INSTALL_PATH>***/server/conf/dxmmsssvr.ini**.
- Reset all the existing passwords, which now need to be encrypted. For example, return to DirX Identity Manager (Connectivity). Select **Global View**→ **Scenarios**→ **My-Company**→ **Main**; select **Identity Store**. Select **Configure...** from the context-sensitive menu. Go to **Bind Profile: Domain_Admin:**. Set **Password** again and then save it.

### 1.3.4. Adding a Certificate for Signed Auditing (Optional)

Although none of the tutorial exercises demonstrate signed auditing, you might want to enable it to see how it works:

- Run the following **dirxgenpse** command in the MS/DOS command prompt or in a UNIX shell:
  ```
  dirxgenpse -D cn=DomainAdmin,cn=My-Company,dxmc=Users,dxmc=DirXmetahub -P 5678 -w <password>
  ```

The **-P** option specifies the PIN of the certificate. In this example, we use the pre-configured value **5678**. If you use another PIN (and you should do this in production environments!), you must set up the **signaturePins** values in the relevant **password.properties** files of the Java-based identity server (IdS-J).

To verify the successful generation of the certificate:

- Start the DirX Identity Manager and log in to Connectivity.
- Go to **Data View** → **Connectivity** (Connectivity dxmc=DirXmetahub) → **dxmc=Users** → **cn=My-Company** → **cn=DomainAdmin**.
- Open the All Attributes tab. You should see a **userCertificate** and a value for **keyOwnerPSE** containing ...**:cn=DomainAdmin,**...

### 1.3.5. Enabling Auditing (Optional)

You may also want to enable DirX Identity's basic auditing feature - this type of audit does not require you to create any certificates. You enable auditing from DirX Identity Manager's **Provisioning** view:

- Log in to DirX Identity Manager.
- In the **Provisioning** view, click **Domain Configuration**. In the **Tree** pane, **My-Company** is already selected and its properties are displayed.
- Click My-Company's **Compliance** tab.
- Click **Edit**, and then check the **Service Layer**, the **Request Workflows** and the **Authentications** check boxes to enable auditing.
- Click **Save**.
- You may also enter an **Audit Trail Folder**. The default folder is '**../AuditTrailDB**'. You can set another folder by selecting **Connectivity** → **Expert View** → **Connectivity Configuration Data** → **Configuration** → **DirX Identity Servers** → **Java Servers** → **My-**

**Company** → **My-Company-S1-**[*your_hostname*]. On the **Status and Auditing** tab, you can find **Audit Trail Folder**.

- Enter '**C:\MetahubData\AuditTrailDB**' as the value. This action directs all audit trail information to the same sample audit trail folder.
- Click **OK**.

This folder definition works for request and all Java-based workflows; that is, workflows that run in the Java-based server.

You can find the folder definition for the service layer in the connected directory of the Extract Audit Trail workflow.

To view this setting:

- Select **Connectivity** → **Global View** → **Scenarios** → **My-Company** → **Main**.
- Right-click the **Audit Trail Database** icon and then select **Configure** from the context menu. The connected directory wizard opens.
- Click the **File Parameters** step.
- Click the **AuditTrailDatabase** row and open it (click the left icon in that line). A dialog opens.
- The Filename attribute contains the path and prefix of the sample audit trail folder: '**C:\MetahubData\AuditTrailDB\auditTrail_**'
- We don't want to change this value, so click **Close** and then **Cancel** again to close the wizard.

To work with auditing properly, some audit policies must be active. The My-Company sample domain contains a set of prepared and activated audit policies:

- In the **Provisioning** view group, click the **Auditing** view. In the tree pane **Audit Trail** → **Audit Policies** → **My-Company**, choose **Accounts**.
- In the **General** tab, check that the **Active** flag is set.
- The next tab **Identifying Attributes** contains some attributes that are included into each audit record for better analysis capabilities in an auditing tool.
- View the attributes to audit for this object type on the tab **Audited Attributes**.

You can view other audit policies to analyze their settings.

## 1.3.6. Setting up Email Notification (Optional)

In a production environment, DirX Identity normally sends email to all the participants in an approval scenario. We are (almost) sure that the people in our sample domain do not exist in your environment as real people who have email accounts. Sending email to these people during the quick start will result in returned email at your mail server, which makes your administrators unhappy. To work around this problem, generation of email from request or Java-based workflows is turned off by default. You can leave the default as it is, or you can turn on email generation and direct it to be sent to your email address during

the quick start exercises.

To configure DirX Identity to send approval email to your email address:

- Log in to DirX Identity Manager.
- In the **Provisioning** view, click **Workflows**.
- Click **Workflows** → **Configuration** → **Services** → **SMTP**.
- Click **Edit**.
- In **SMTP host**, specify your mail server's SMTP server name.
- In **Map mail address**, specify your email address to direct approval mail to be sent to your account.
- Click **Save** to store your changes.

### 1.3.7. Setting up Menu Policies (Optional)

By default, all users can see all Web Center menus and submenus once they have logged in. You can define menu policies for controlling the visibility of Web Center menus and submenus. To enable these menu policies:

- Click **Provisioning** → **Domain Configuration**.
- Click the **Policies** tab.
- Click **Edit** and then check the **Enable menu policies** checkbox.
- Click **Save**.

### 1.3.8. Restarting the Servers

After closing **DirX Identity Manager** we now use the Windows Services tool to stop and then re-start the following servers:

Use the Stop selection to stop the servers in the following order:

1. Apache Tomcat server (name: Apache Tomcat *n*)
2. Java-based Identity Server (name: DirX Identity IdS-J-*domain*-S*n* V*n.n*)
3. C++-based Identity Server (name: DirX Identity IdS-C V*n.n*).
4. Message Broker (name: DirX Identity Message Broker *n*)

Use the Start selection to start the servers in the following order:

1. Message Broker (name: DirX Identity Message Broker *n*)
2. C++-based Identity Server (name: DirX Identity IdS-C V*n.n*).
3. Java-based Identity Server (name: DirX Identity IdS-J-*domain*-S*n* V*n.n*)
4. Apache Tomcat server (name: Apache Tomcat *n*)

do not use the Restart selection to restart the servers; sometimes it does

not work correctly.

### 1.3.9. Setting up the Portal Target Systems

We assume that the Intranet and Extranet Portal target systems also reside in the sample-ts area. We'll use them later in the password management exercises. The corresponding workflows are set up accordingly. To synchronize the target system content to the connected system, we'll run their synchronization workflows:

- Log in to DirX Identity Manager **Provisioning** view group: Select the **Provisioning** view group and then enter the password in the log in dialog.
- Click the **Target Systems** icon in the left pane to select it.
- Right-click the **Extranet Portal** target system. DirX Identity Manager starts reading some configuration data and then displays the context-sensitive menu.
- Select **Connectivity** → **Workflows** → **Ident_Extranet_Realtime** → **Run Workflow** to run the workflow to synchronize the Extranet target system.
- A confirmation window displays the message **Java-based Workflow successfully initiated**.
- Click **OK**.
- Right-click the **Intranet Portal** target system. DirX Identity Manager displays the context-sensitive menu.
- Select **Connectivity** → **Workflows** → **Ident_Intranet_Realtime** → **Run Workflow** to run the workflow to synchronize the Intranet target system.
- A confirmation window displays the message **Java-based Workflow successfully initiated**.
- Click **OK**.
- In **Data View** → **Connectivity** → **Sample-TS o=sample-ts**, check that the two nodes (**ou=Extranet** and **ou=Intranet**) are filled with accounts and groups.(Click the refresh button if they are not visible).

## 1.4. Working with the Quick Start

When you follow the exercises in the quick start, you should:

- Back up the example database between exercises
- Check the example database statistics between exercises, such as the number of entries created and the operations performed

### 1.4.1. Backing up the Database

Once you have completed your preparation for the quick start, you should back up the example database so that you can restore the prepared example database should you run into problems later on.It is also a good idea to back up your example database each time you successfully complete a quick start exercise.This way, you can retry an exercise that

failed for some reason without having to return to the very first exercise.

Refer to your directory service product's user documentation for instructions on how to back up your data.For example, if you are using DirX Identity with the DirX directory service, you can use **dirxbackup** to save and restore the database; this command is described in the *DirX Administration Reference*.

### 1.4.2. Checking the Database

It's also a good idea to examine the example database after you've performed a quick start exercise to see its effect on the database contents and operation.Refer to your directory service's user documentation for instructions on how to view directory database statistics and monitor the database.For example, if you're using the DirX directory service, you can use the **dirxadm** tool's **nmi show** operation after you've performed an exercise to display statistics about the operations performed on the database during that exercise.The *DirX Administration Reference* provides usage details about **dirxadm nmi show**.

## 1.5. User Self-Registration

In this section, we show how an employee of one of My-Company's customers self-registers as a My-Company user and requests some of the services that My-Company offers to its customers.This section demonstrates how to:

- Start the self-registration process
- Enter the user data required for creating the new customer identity
- Enter the user's password
- Select some of the services that My-Company provides to its customers
- Review and submit the supplied user data
- Monitor the self-registration process
- Approve the user creation request and the services that the customer employee has selected
- Log in as the new user and view his data

Self-registration is disabled by default for security reasons. See DirX Identity Web Center Customization Guide how to enable it.

### 1.5.1. Requesting the Services

My-Company's customers include three re-seller companies: Mercato Aurum in Italy, and TakeAway and MultiMarket in the U.S.A. My-Company makes the following services available to employees of these companies:

- The My-Company monthly newsletter, which announces new products, highlights company achievements, and provides general company information of use to its employees and its customers
- Services that allow customers to participate in My-Company's hardware and software

beta programs

Now we'll demonstrate how someone in one of My-Company's customer organizations can self-register for access to My-Company's customer services. Nico Farfello is a new Mercato Aurum employee who is responsible for testing the company's products with upcoming My-Company hardware releases. In order to perform his testing tasks, Nico needs access to My-Company's hardware beta programs services.

#### 1.5.1.1. Starting the Self-Registration Process

We'll use DirX Identity Web Center's self-registration facility to register with My-Company as Nico Farfello and request the necessary services. To start the self-registration process:

- Enter the URL **http://localhost:8080/webCenter-My-Company** in an Internet browser to start the DirX Identity Web Center. (The Web Center may take a little time to start up the first time you use it because it runs some preparation tasks. This is also true for each Web page you visit for the first time. Subsequent visits to the same page are much faster.)
- Click **Register**.

We describe the Web Center handling in English. You can use the **Language** menu in the upper right-hand corner to switch to German.

This action automatically starts the DirX Identity workflow that handles user self-registration requests for My-Company's customers. We will show how to monitor this workflow's progress later on in this sequence.

#### 1.5.1.2. Providing User Data

A few seconds after we click Register, Web Center displays a dialog for entering mandatory and optional data about the person whose identity is to be created in the DirX Identity store. Fields that are mandatory are highlighted in red. We use this dialog to enter required and optional information about Nico Farfello:

- First, we must adjust the **Folder** path to point to the Mercato Aurum part of My-Company's Users directory tree. This is the location in My-Company's identity store (user tree) at which Nico's identity (user entry) is to be created. Click to the right of **Folder**. Navigate to the **Customers** node, and then select **Mercato Aurum**.
- In the fields displayed in the dialog:
- Leave **Title** empty.
- Enter **Farfello** in **Last Name**. Click the TAB key to move to the next field.
- Enter **Nico** in **First Name**. Click the TAB key.
- Select **Mercato Aurum Rome** in **Location**. Click the TAB key.
- Enter **+39 6 2345 8793** in Phone. Click the TAB key.
- Enter **nico.farfello@mercato-aurum.it** in Email.
- Click **Save**. Web Center then displays the password selection dialog.

#### 1.5.1.3. Entering a Password

The password selection dialog allows you to specify a password for the identity (user account) to be created for you in the DirX Identity store. You must be sure to specify a password that follows the guidelines for password creation described in the dialog, and make sure you make a note of this password so that you can use it later on. In our case, we need to create a password with one alphanumeric character, one special character, one numeric character, and one uppercase character. The entire password must be no less than 6 characters and no more than 8 characters.

- Enter **1!dirX** in **Password**.
- Enter **1!dirX** again in **Repeat password**.
- Click **Save**. Web Center then displays the services selection dialog.

#### 1.5.1.4. Selecting the Services

The services selection dialog allows you to request the services you want. In DirX Identity, these services are modeled by roles, which we'll discuss in more detail later on in this tutorial. In Nico's case, he wants the Hardware Beta Programs role for access to My-Company's beta program services:

- Click **Search** to the right of **Search for**. Web Center shows the services available to My-Company's customers. You can see that Customer Newsletter, Hardware Beta Programs, and Software Beta Programs are now displayed in **Available roles**.
- Clicking a line toggles the checkbox to the left of a line. Click the **Hardware Beta Programs** and **Customer Newsletter**.

**Hardware Beta Programs** requires approval. This means that someone in My-Company must approve the request for this service. We will talk about this process later on in this sequence.

- Click the down arrow at the bottom of the dialog to select the services.
- Click **Save** to save your selections. Web Center next displays a data confirmation dialog.

#### 1.5.1.5. Confirming the User Data

The data confirmation dialog gives you the opportunity to review and change the identity (user data) information you've provided to the registration process.

- Click **Accept** to approve the information you have provided and proceed with the registration process.
- After a few seconds, Web Center displays its Log in page and a message indicating that the registration request has been accepted.
- Since we have no more to do in this step, click **OK** to close this window and terminate the Web Center browser.

### 1.5.2. Monitoring the Self-Registration Request

We can use DirX Identity Manager's Monitor view to see the progress of Nico Farfello's registration request:

- Log in to DirX Identity Manager's **Provisioning** view.
- Click **Workflows**, and then click **Workflows** → **Monitor** → **My-Company** → **Users** → **Customer Self Registration**. In a production environment, this folder contains sub-folders named with dates; for example, **2025-07-17**. Each sub-folder contains status information about the request workflows that have executed on these dates.
- In our case, we have one folder with today's date. Click this folder, and then look for **Nico Farfello** in the list of workflows (it should be the only one there). Click it. In the **General** tab, Manager displays a graphical representation of the user self-registration request workflow that is processing Nico's registration request, as shown in the following figure.

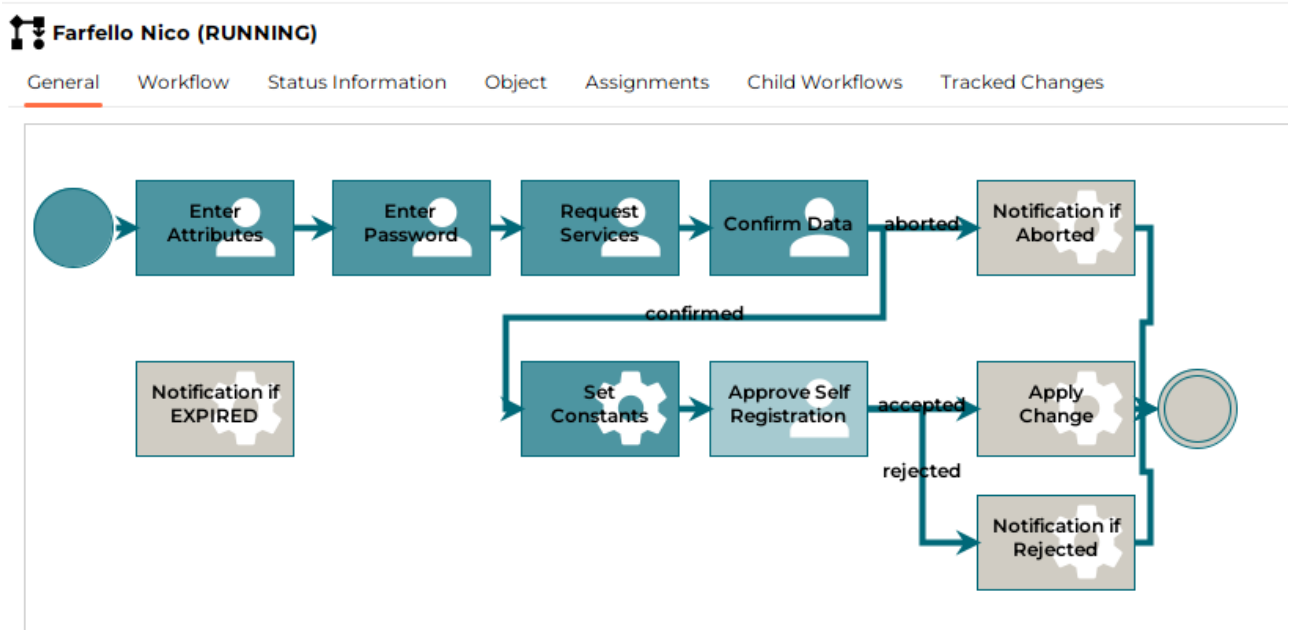



*Figure 1. Customer Self Registration Workflow for Nico Farfello*

You can see that the steps we have just completed are all highlighted in green, which means that these activities have completed successfully. The next step – approving the self-registration request - is highlighted in light-green to indicate that it is the activity that is currently being processed. Process activities that have not run yet are colored in gray.

### 1.5.3. Approving the Self-Registration Request

Nico's request for registration as an identity in My-Company's identity store (user in My-Company's directory tree) requires approval from a member of My-Company's customer registration hotline. Three of My-Company's Sales employees are members of this group:

- Briner Ruben, who is the assistant to My-Company's General Sales Manager
- Klarmann Bruno, who is the manager of the Sales Europe organization
- Ratnam Dilip, who is the manager of the Sales US organization

The request workflow's approval activity will automatically notify each of these people by email about Nico's registration request. Only one of these people needs to approve the request in order for it to be successfully processed. In the next sections we'll show how one of these people - Bruno Klarmann - uses the Web Center to view and approve Nico's request. We can do this with or without email notification; it depends on how you set up email notification when you prepared to use the quick start.

#### 1.5.3.1. Approving without Email Notification

If you left the SMTP configuration in the default state when you prepared to use the quick start (no email notifications are performed):

- Start the Web Center by entering the URL **http://localhost:8080/webCenter-My-Company** in your Internet browser.
- Enter **Klarmann Bruno** in **Name** and enter the password. (Remember that all persons in the sample domain are set up to have the same password.)
- Click **Log in** (or press RETURN) to log in.
- Follow the steps described in "Approving the User Creation Request".

#### 1.5.3.2. Approving with Email Notification

If you entered your email address into the **Map mail address field** of the SMTP service in "Setting up Email Notification", you can use the email notifications you received from the user self-registration workflow to approve the privilege assignment.

- Open your mail client. You should have three email messages that inform you about user creation approval requests: one for Briner Ruben, one for Klarmann Bruno, and one for Ratnam Dilip.

  View one of these emails and then click the link that starts the browser and displays the login page of the Web Center.
- Enter the full name of the approver you selected in **Name**.
- Enter the password and click **Log in**.
- Follow the steps described in "Approving the User Creation Request".

#### 1.5.3.3. Approving the User Creation Request

Now we're logged into the Web Center as one of the approvers of Nico's request and can approve her request either from our task list dialog or by clicking directly on the **Approve Self-Registration** Task for Nico Farfello in the home page on the right side below **My Tasks**.

In the first case, you must follow the next two bullet points. In the second case, you can follow on directly with the third bullet point.

- Select **Work List** → **Task list**. Web Center displays Nico Farfello's data in the "Approve Self-Registration" dialog.
- Click on the task for Nico Farfello.

- Enter **N**. **Farfello is an employee of Mercato Aurum** in **Reason**. (It's good practice to provide a reason for your decision, especially if you reject a request.)
- Click **Accept** to grant the request.

**Task list** is now empty.

#### 1.5.3.4. Checking the Result of User Creation Approval

While we're logged in as Bruno Klarmann (or one of the other approvers), we can use the Web Center's User Management view to see if an entry for Nico has been successfully created in the identity store.

Simply enter **F** in the User Quick Search bar and press RETURN. Follow on with the fourth bullet point below.

As an alternative, you can select the user by using the menubar options:

- Select **Users** → **Select user**.
- Since Nico is an employee of Mercato Aurum, his user entry should be located in the Mercato Aurum part of the My-Company directory tree. Click to the right of **Search base**, then select **Mercato Aurum** under **Customers** from the tree view.
- Enter **F** in **Search for** and then click **Search**.
- An entry appears for Nico. Click it. If Nico is not yet visible, wait a short time until and try again. The reason is that the workflow task is not yet completed.
- Click **Roles**. You can see four entries:
- Two for the services that Nico has requested (indicated by **Manual** in the last column **Mode**): one for the customer newsletter, and one for the hardware beta program service.
- Additionally two roles were assigned by rules: a silver customer and a platinum customer role. The first comes from the rule that all users of type **Customer** get the silver customer status, the second was assigned due to the fact that all users of Mercato Aurum are treated as platinum customers (excellence bonus program).
- Click on the **Download all tabs** button ( ) from the button group placed left to the Roles button to see all tabs.
- Look at the **State** column for the four roles. The state for **Customer Newsletter**, **Silver Customer** and **Platinum Customer** is ENABLED, which means that Nico has successfully obtained this service. The state for **Hardware Beta Programs** is *ADD*, which means there is more to do before Nico can gain access to this service.
- Select **Users** → **Show subscription status**. Per default, you can see a running workflow for the hardware beta programs.
- Click on the **Download all tabs** button ( ) left to the **Running Workflows** button to see all tabs.
- You can still see the running hardware beta programs workflow but additionally the

already completed self-registration workflow that has the state **Succeeded**.

- Click the running workflow entry **Farfello Nico** –> **Hardware Beta Programs**. In **Running Workflows**, you can see that the current activity is **Approval by Sales Manager**, with Briner Ruben listed as a **Participant**. This means that Mr. Briner must approve Nico's request for the Hardware Beta Programs service. We'll talk more about this in the next section.
- There is nothing more to do, so click **Logout**.

## 1.5.4. Approving the Hardware Beta Programs Request

Recall that "requires approval" was shown for the hardware beta programs service when Nico requested it ("requires approval" was checked in the information about the role). While satisfying a customer's request for My-Company's newsletter simply adds the person to an email list, satisfying a request for participation in My-Company's hardware or software beta program is more complex. An approval step serves as notification to the person responsible for the program that a new customer wants to be included in it, and triggers the approver to perform all the tasks related to getting the customer into the program.

My-Company's Sales manager Ruben Briner is the person responsible for My-Company's hardware beta programs, and so he must approve all requests for this service from My-Company's customers. Consequently, the approval step for the hardware beta programs service notifies Ruben that he needs to talk with Nico about program schedules, include her in appropriate status meetings, send her detailed information about the hardware beta program, and set up consulting and support contact information for him.

#### 1.5.4.1. Monitoring the Hardware Beta Programs Approval Workflow

DirX Identity is now running a new, completely separate request workflow to handle the approval of Nico's request for the hardware beta programs service. We can use DirX Identity Manager's Provisioning → Workflows Monitor View to check the progress of this new and separate approval workflow:

- In **Provisioning** → **Workflows**, go to **Monitor** → **My-Company** → **Approval** → **Approve Customer Self Services** and look for a folder with today's date.
  If **My-Company** is not displayed it is necessary to refresh your data: Click **Monitor** and the refresh button .
- Select the **Farfello Nico** → **Hardware Beta Programs** workflow - this workflow is handling the approval of the hardware beta programs service.
- In the **General** tab, you'll see the workflow's graphical representation. The activity "Approval by Sales Manager" is highlighted in grey, which means it is currently being processed.

#### 1.5.4.2. Approving the Hardware Beta Programs Request

Now we'll complete the approval process by logging into the Web Center as Briner Ruben, who must approve the Hardware Beta Programs service request.

- Switch to the Web Center again. Enter **Briner Ruben** in **Name** and **dirx** in **Password**

All users in the sample domain have the same password.

- Click **Log in** (or press RETURN) to log in.
- Click on the Approval by Sales Manager Task for Nico Farfello in the **My Tasks** section of the home page or select this task when performing the **Work List** → **Task list** operation. Web Center displays an approval dialog.
- Enter **All open issues clarified. All necessary tasks done.** in **Reason**. (It's good protocol to provide a reason for your decision, especially if you reject a request.)
- Click **Accept** to grant the request.

**Task list** is now empty.

- Since we have no more to do in this step, click **Logout** to exit from Briner's user account.

## 1.5.5. Using the New Account and Password

Now we'll log in to the Web Center as Nico Farfello and examine his user information and privileges. We can do this with or without email notification; it depends on how you set up email notification when you prepared to use the quick start.

### 1.5.5.1. Logging In without Email Notification

If you left the SMTP configuration in the default state when you prepared to use the quick start, email notifications are not performed. To log in as Nico:

- Switch to the Web Center again (or start it by entering the URL http://localhost:8080/WebCenter-My-Company in your Internet browser.)
- Enter **Farfello Nico** in **Name** and the password **1!dirX**.
- Click **Log in** (or press RETURN) to log in.
- Follow the steps described in "Viewing the New User".

### 1.5.5.2. Logging In with Email Notification

If you entered your email address into the Map mail address field of the SMTP service in "Setting up Email Notification", you can use the email notification you received from the user self-registration request workflow to log in:

- Open your mail client. You should have an email message for Nico Farfello that informs you that your registration request was successful. View this email and then click the link that starts the browser and displays the Log in page.
- Enter **Farfello Nico** as **Name**.
- Enter the password **1!dirX** and then click **Log in**.
- Follow the steps described in "Viewing the New User".

#### 1.5.5.3. Viewing the New User

Now we are logged into Web Center as Nico and can view her home page, showing some personal data, her roles and her tasks.

- Click on **Self Service** → **Display summary** to view her user summary.
- Web Center displays a summary of Nico's user information.You can see the user data that we entered for Nico during the self-registration process.
- Click **Roles** to see the roles that have been assigned to Nico.You see that he has **Customer Newsletter** and **Hardware Beta Programs** roles in ENABLED state, which correspond to the services he requested.He also has two more roles that were assigned automatically by rules.This information was also shown in the home page after log in, where you can click on **more...** in that section to display this page.
- Click **Permissions** to see the permissions that have been assigned to Nico. **Customer Newsletter**, **Hardware Beta Programs**, **Platinum Customers** and **Silver Customers** permissions are listed.The state is INHERITED because in the My-Company domain, all permissions except for signature level permissions are inherited from the assigned roles.
- Click **Groups** to see the groups that have been assigned to Nico.You see **Customer Newsletter**, **Hardware Beta Programs**, **Platin Customers** and **Silver Customers** groups.You can read more about My-Company's role, permission and group relationships - called its privilege structure - in the chapter "About the My-Company Sample Domain".

The Target Systems column - it identifies the systems to which these groups belong; in this case, it is the Extranet Portal target system, which is My-Company's extranet application for its customers and suppliers.You can read more about My-Company's target systems in the chapter "About the My-Company Sample Domain".

- Click **Accounts** to see the accounts that DirX Identity has automatically created for Nico.You see one account for Nico in the Extranet Portal target system that was automatically created by DirX Identity.
- There is no more to do in this step, so click **Logout**.

So far, we've shown how to use DirX Identity's user self-registration feature to add a new user to the Identity Store and automatically assign him privileges - roles, permissions, and groups - that correspond to the services he requested during registration.In the next section, we'll discuss how to add a new user to My-Company's user tree and assign him privileges both by hand and automatically with provisioning rules.

## 1.6. Adding a New User

In this section, we hire a new contractor named Mark Teacher as an additional trainer for the Professional Services department. He will work for the Asian market. This section demonstrates how to:

- View the users of the Professional Services department
- Add the new user to this department
- Assign privileges to the new user by hand
- Add a user password
- Approve the user's privileges
- Add the new user to a project
- Assign privileges to the user automatically

## 1.6.1. Viewing Users

Before we add Mr. Teacher, let's take a quick tour of the My-Company user view to become familiar with the sample domain in this area. You will see that some of the user data in the sample domain is not in an optimal state. This is the data that we'll correct step by step in the next sections of this quick start. For now, we'll log into DirX Identity Manager and use the **Provisioning** → **Users** view to look at some of My-Company's personnel data.

- Click **Provisioning** → **Users**. The tree contains folders for Customers, Suppliers, the employees of My-Company and System (empty). It also contains the standard query folders for the user view: Errors, Inconsistent, Relative Time Constraint, To Be Deleted, ToDos and Variable Time Constraint. You can use these query folders 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. We'll demonstrate how to use query folders later on in this quick start.
- Open **My-Company** to view the different departments.
- Open **Professional Services**. You can see the seven employees of this department. Let's take a closer look at some of these employees.

#### 1.6.1.1. Checking Auffret Jean-Marc

In DirX Identity Manager → **Provisioning** → **Users** → **My-Company** → **Professional Services**:

- Click **Auffret Jean-Marc** to view the details about this person in the **General** tab.
- On the **Location** tab, you can see that he is located in Barcelona, Spain.
- On the **Organization** tab, you can see that he works for Professional Services.
- On the **General** tab, you can see in the **Description** field that he handles the Southern Europe area. His employee number is 7836 and he is an Internal employee.
- On the **Operational** tab, you can see that he is in the ENABLED state, which means he is an active user.
- On the **Relationships** tab you can see that his manager is Lionel Bellanger. Click to the right of this entry to display this person's information.

#### 1.6.1.2. Checking Bellanger Lionel

Lionel Bellanger is the manager of Professional Services and handles the Central Europe area. He's located in Paris, France. His manager is the general manager of My-Company, Olivier Hungs.

- Click the **Location** tab to check for **Country** and **Location**.
- Clich the **General** tab to check for **Employee Type**.
- Click the **Relationships** tab for **Manager**.
- Click the **Organization** tab for **Organizational Unit**.

These fields represent the user attributes that are used for automatic privilege assignment and provisioning. We'll talk about this concept later on in the tutorial. Now let's check another user: Blander Dyan.

#### 1.6.1.3. Checking Blander Dyan

In DirX Identity Manager → **Provisioning** → **Users** → **My-Company** → **Professional Services**:

- Click **Blander Dyan** to see that he works in Chicago and is responsible for Northern USA / Canada.
- Click the **Communication** and the **Location** tabs to view his communications-related user attributes like e-mail and phone.

Now let's take a look at this user's privileges.

#### 1.6.1.4. Checking Blander's Privileges

In DirX Identity Manager → **Provisioning** → **Users** → **My-Company** → **Professional Services** → **Blander Dyan**:

- Click the **Assigned Roles** tab to view the roles that are assigned to this user. In the **Assigned by** column, you can see that the first role has been assigned automatically by a provisioning rule (**rule**), that the second role is inherited from a business object (**BO**) and that the third role has been assigned by hand (**manual**). Because Dyan is an internal employee he has received the role **Internal Employee** by rule. We will analyze the content of this role later on in this quick start. The My-Company administrators have determined that the tools required for members of the Sales and Professional Services departments are the same. Consequently, the role **Sales Tasks** has been automatically assigned via inheritance from the organizational unit business object to all members of Professional Services. This assignment is not correct and you will see in a later exercise how to change it. Additionally, Dyan works as a trainer, and thus has the role **Trainer** assigned to him by hand.
- Click the **Assigned Permissions** tab to view the permissions assigned to Dyan Blander. Signature Level 1 is assigned to all internal employees. In My-Company, only three signature levels exist. All other permissions are inherited from the assigned roles.
- Click the **Assigned Groups** tab to view the groups assigned to this user. The **Target Systems** column shows the target systems to which these groups belong. Click the

column header to sort the target system column. You can see that most groups belong to the Windows Domain USA target system but that the user also has access to the Intranet, Extranet and SAP R3 target systems. The Signatures target system is a virtual target system - there is no physical connected system behind it, because signatures in My-Company are pure paper processes. Due to this fact, synchronization to this connected system is not necessary, which results for this group assignment in an immediate ENABLED state. The Intranet and Extranet target systems are 'real' target systems (modeled as LDAP trees in this sample scenario). All other groups are in state ADD which means that the target system synchronization has not yet been performed. The reason is that these connected systems do not exist in this sample scenario.

Now let's take a look at this user's accounts.

#### 1.6.1.5. Checking Blander's Accounts

In DirX Identity Manager → **Provisioning** → **Users** → **My-Company** → **Professional Services** → **Blander Dyan**:

- Click the **Accounts** tab to view the corresponding accounts in the target systems that DirX Identity has created automatically for this user.

The Signature target system does not have an account.

The **State** of all four accounts is ENABLED whereas the **State in TS** in only ENABLED for the accounts of the Intranet and Extranet target systems. This means that these accounts have already been provisioned to the connected systems. The other two accounts are still in **State in TS** NONE.

The "Users" section in "Understanding the Sample Domain" provides more information My-Company's users and its department structure, while the "Privileges" section provides more information about My-Company's privilege structure (its roles, permissions, and groups). For now, let's move on to the next topic, "Adding the User".

### 1.6.2. Adding the User

We'll use the DirX Identity Web Center to create the new identity for Mark Teacher. First, we'll run it from an Internet browser and log in as Taspatch Nik, since he is the user administrator for the Professional Services Department. Then we'll use the Web Center's User Management view to add Mark Teacher's information to the identity store.

You can also import identities from other sources; "Importing Identities" explains how to do this.

- In an Internet browser, enter the URL **http://localhost:8080/webCenter-My-Company** to start the Web Center.
- Enter **Taspatch Nik** in **Name** and enter the password. (Remember that all users in the sample domain are set up to have the same password.)

- Click **Log in** (or press RETURN).
- Select **Users** → **Create new user**.
- A workflow selection dialog appears. In a production environment, the DirX Identity customer defines the workflows he requires and assigns them to specific groups of people. In our case, the workflows listed here are My-Company sample domain workflows that Nik is permitted to run. Select **create a user stepwise without approval** because Nik, as a user administrator, is permitted to create users without the need for any additional approvals.

A dialog appears to enter the new user's properties. Mandatory fields are shown in red. Use the TAB key to move to the next input field.

- Click to the right of **Folder** to select the location in the sample domain's user tree at which to create the user. A dialog opens and shows the available organization tree. Open **My-Company**, and then select **Professional Services**.
- Enter **Teacher** in **Last name**. Click TAB to move to the next field.
- Enter **Mark** in **First name**. Click TAB to move to the next field.
- Enter **Trainer for Asia** in **Description**. Click several TABs to move to the **Employee type** field.
- Select **Contractor** in **Employee type**. Click TAB to move to the next field.
- Enter **83730** in **Employee number**. The employee number must be a unique value in your system.

In a productive environment you could create this number automatically.

- Click TAB to move to the next field.
- Select **My-Company San Jose** in **Location**. Depending on this value, the correct country is specified automatically.
- Click the icon to the right of **Manager**. The **Select a Manager** dialog appears. Enter **B** in **Search for**, and then click **Search** to get a list of names that begin with "B". Select **Bellanger Lionel** from the list.
- Select **My-Company** in **Company**.
- Select **Professional Services** in **Organizational Unit**. Click into the next field.
- Enter **+1 408 876** in **Phone**. Click several TABs to move to the **E-Mail** field.
- Enter **Mark.Teacher@My-Company.com** in **E-Mail**.
- Click **Save**. Web Center then displays a privilege assignment dialog.

### 1.6.3. Assigning Privileges by Hand

Privileges (roles, permissions, and groups) can be assigned to a user either by hand or automatically via a provisioning rule or via inheritance from a business object (for example an organizational unit). Here we show how to assign them by hand as part of the Web

Center's user creation process.

Because Mark shall work as a trainer, he needs the Trainer role. Privileges can only be assigned by people who are permitted to do so; access policies specify the privileges a person can assign and the people to whom he can assign them. The My-Company sample domain defines an access policy for user administrators that permits them to assign roles to users. Because he is a user administrator, Nik Taspatch has the necessary access rights (which comes from the user administrator access policy) to assign the Trainer role to Mark Teacher.

- Enter **T** in **Search for**, and then click **Search** to get a list of roles that begins with "T".
- Mark the **Trainer** role for selection (set the flag or simply click on the line), and then click the down-arrow button to move it to **Assigned roles**.

The state of this new role assignment is ToBeApproved, which means that a new approval workflow is to be started to notify the appropriate approvers.

- Click **Save**.

  Next Nik Taspatch will assign a password to this user.

## 1.6.4. Adding a User Password

The last administrative task that Nik Taspatch needs to perform is to specify a default password for Mark Teacher's new user account. We'll use the Web Center's Reset Password dialog to carry out this task:

- Select **Teacher Mark** in **Users** → **Select user** dialog or search for **T** in the Users quick search bar and select **Teacher Mark**.
- Select **Users** → **Reset password**.
- Enter a password of your choice. Be sure to follow the password policies shown in the dialog - for example, **1!dirX** is a password that complies - and remember this password for later use
- Click **Save**. DirX Identity starts a password management workflow that assigns the password you specified to the new user Mark Teacher.
- There is no more for Nik Taspatch to do, so click **Logout**.

## 1.6.5. Approving the User's Privileges

In this section, we'll show how to use the Web Center to approve the privileges that Nik Taspatch has requested for Mark Teacher, and how to monitor the progress of these approvals from three different tools: the Web Center, DirX Identity Manager, and DirX Identity Web Admin.

### 1.6.5.1. Checking the Approval Workflow

Recall that DirX Identity automatically creates an approval workflow to handle the approval

of Mark's Trainer role. You can view this approval workflow in several ways: with Web Center, with DirX Identity Manager, and with DirX Identity Web Admin. Let's do it here with DirX Identity Manager:

- In **Provisioning** → **Workflows** click **Monitor** → **My-Company** → **Approval** → **4-Eye Approval**. You will see a folder with today's date. This folder contains the created workflow. Open it.
- Click **Teacher Mark** → **Trainer**. The **General** tab displays the workflow's structure.

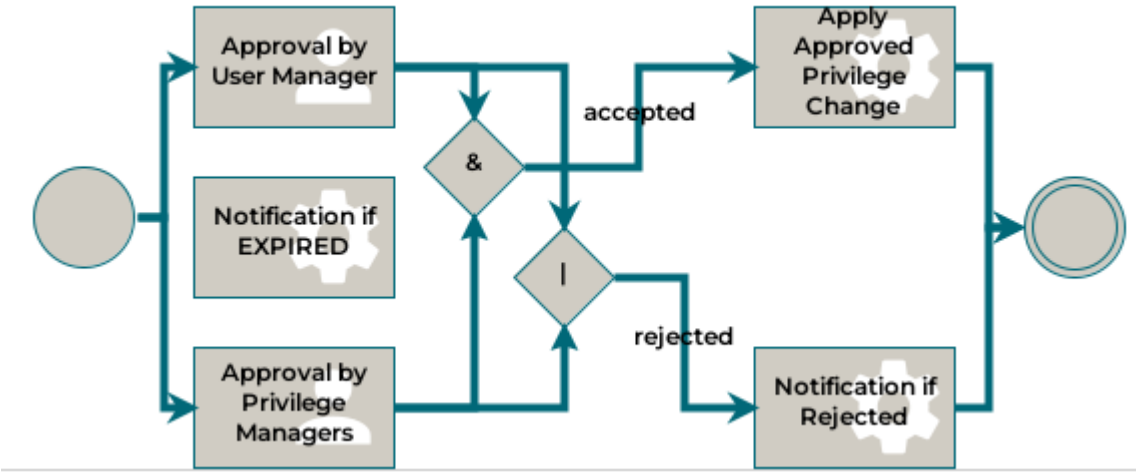



*Figure 2. Approval Workflow Before Any Approvals*

- In the structure view, you see two parallel steps: Approval by User Manager and Approval by Privilege Manager. In the My-Company sample domain, both a user's manager and the manager of the privilege must approve a role assignment. Right-click **Approval by User Manager**, and then select **Open**.
- Click the **Status Information** tab. In **Participants**, you see the name **Bellanger Lionel**. Mr. Bellanger is Mark Teacher's manager, and so he must approve the role assignment.
- Click **Close**. Manager returns you to the workflow structure view.
- Right-click **Approval by Privilege Managers** to open it, and then click the **Status Information** tab. In **Participants**, you can see that **Costello Marcella**, who owns the Trainer role, must approve its assignment to Mark. Click **Close** to return to the structure view.
- The plus sign icon indicates that both people must make a decision to approve the privilege assignment; if one person denies it, the privilege will not be assigned.

#### 1.6.5.2. Approving the Request

Now we are ready to approve the assignment. First, we'll log in to Web Center as Lionel Bellanger (remember that you can use email to log in if you enabled email notification when preparing to use the quick start):

- Start the Web Center by entering the URL **http://localhost:8080/webCenter-My-Company** in your Internet browser (or switch back to it if you left it running).
- Enter **Bellanger Lionel** in **Name** and enter the password **dirx**.

- Click **Log in**.
- Select **Work List** → **Task list** and click the task in the list. The approve assignment dialog appears (**Approval by User Manager**). You can also click directly on **Approval by User Manager** for Teacher Mark in the **My Tasks** section of the home page.
- Click the task in the list. The approve assignment dialog appears (**Approval by User Manager**).
- Enter **Trainer for Asia** in **Reason** (remember that it's good practice to give a reason for your decision, especially when you reject a request)
- Click **Accept**. The work list is now empty.

#### 1.6.5.3. Checking Approval with Web Center

We can check the result of Mr. Bellanger's approval in several ways. First, let's check it with the Web Center while we're logged in as Mr. Bellanger:

- Select **Users** → **Select user**, enter **T** in **Search for**, and then click **Search** to return a list of users whose last names begin with "T". You can also enter **T** in the user quick search bar and press RETURN.
- Select **Teacher Mark** from the list. Web Center displays a user summary for Mark.
- Click **Roles**. You can see three roles:
- **Contractor** - this role was assigned by rule.
- **Sales Tasks** - this role is inherited from the organizational unit business object.
- **Trainer** - this role is in state *ADD* which means it is still in approval.
- Select **Users** → **Show subscription status** to see details about the approval workflow that is running for the Trainer assignment. You can see that the workflow's state is **Running**.
- Click the line to view more details of this workflow.
- In the section **Running activities** you can see the pending approval activity for Marcella Costello.
- The section **Finished activities** shows the completed approval from Lionel Bellanger.
- There is nothing more to do, so click **Logout**.

#### 1.6.5.4. Checking Approval with DirX Identity Manager

Next, let's check it with the DirX Identity Manager.

- Go to the folder with today's date in **Provisioning** → **Workflows** → **Monitor** → **My-Company** → **Approval** → **4-Eye Approval** and click **Teacher Mark – Trainer** workflow again.
- Click the refresh button if necessary. You can see that the approval activity for Mr. Bellanger is now green, indicating success.

#### 1.6.5.5. Checking Approval with Web Admin

We can also use DirX Identity Web Admin to check the status of the approval process. Workflow Service Administration is a simple interface intended for use by administrators (not end users!) to view and control the DirX Identity Java-based servers. To use Web Admin:

- Start a new Internet browser instance and enter **http://localhost:40000/admin**. The Web server prompts for an administrative account and a password. Enter **admin** in **User Name** and **wE3!dirx** in **Password**.
- Open **Java Server** → **Workflow definitions** → **Request workflows** → **My-Company** → **Approval**. Here you see, for example, the definitions for approval workflows that are loaded and active in the Java-based Server.
- Click **Workflow instances** → **Request workflows**. click on **Search**. You can see the **Teacher Mark → Trainer** workflow in the state RUNNING as well as the creation workflow for Mark Teacher, which has already completed (it is in the state SUCCEEDED). Click on the workflow to open the details.
- That's all we'll use the Web Admin for now, so you can close the browser window to exit the program.

#### 1.6.5.6. Continuing the Approval Process

Now we'll complete the approval process by approving the request from the Web Center as Marcella Costello:

- Switch to the Web Center again. Enter **Costello Marcella** in **Name** and enter the password and then click on **Approval by Privilege Managers** for Teacher Mark in her **My Task** section or use **Work List** → **Task list** and click the entry in the list. The approve assignment dialog appears.
- Enter **New Trainer for Asia** in **Reason**.
- Click **Accept**.
- There is nothing more to do, so click **Logout**.

#### 1.6.5.7. Checking the Results of the Approval Process

Now we'll check the results of the approval process: we'll check the workflow's structure and the user Teacher's privileges.

##### 1.6.5.7.1. Re-Checking the Workflow's Structure

First, we'll re-check the request workflow's structure (we assume that you have kept a DirX Identity Manager session open during this exercise):

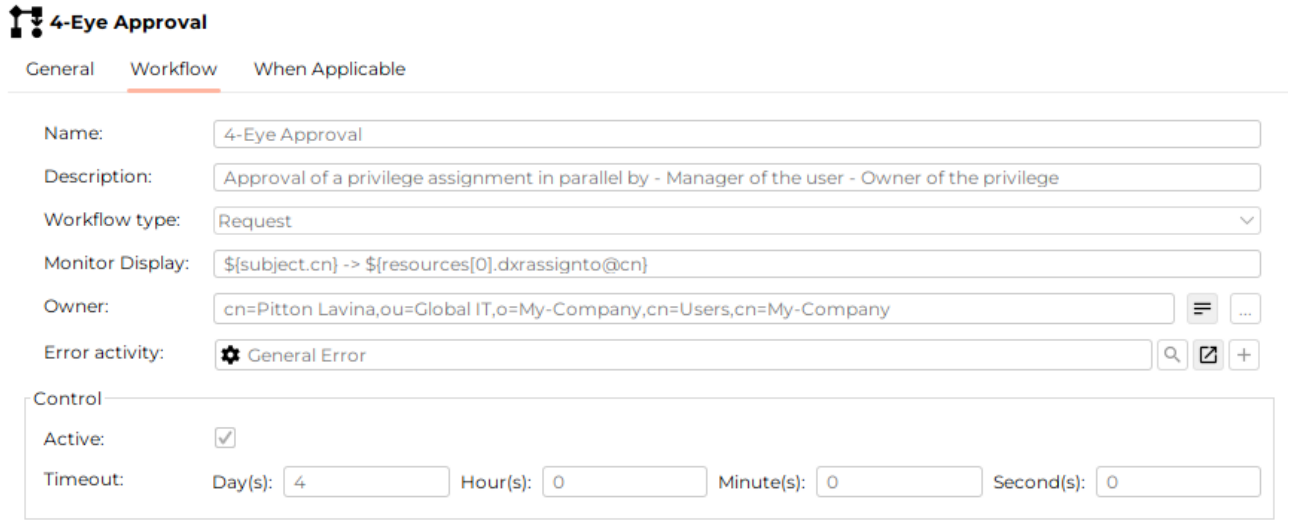
- Switch to the DirX Identity Manager and select **Provisioning** → **Workflows** → **Monitor** → **My-Company** → **Approval** → **4-Eye Approval** → *today's date* → **Teacher Mark - Trainer**. You can view the workflow structure again.
- Click the refresh button to view the updated figure.

Figure : Approval Workflow Structure After All Approvals

- You can see that all the workflow's approval steps are now highlighted in green, and so is the Apply Approved Privilege Change activity, which has assigned the Trainer role to Mark Teacher. This means that the approval workflow has completed successfully.
- Click the **Status Information** tab. You can use **Status expiration** to see the expiration date at which an administrative task can delete the status entry for the role assignment of Trainer to Mark Teacher. In our case, the administrative task can automatically remove the entry after 4 day (as a default value). If needed, you can customize this behavior - see **Attention this is alternative step** below. After this time, you will find information about this role assignment and the corresponding approval only in the generated audit trail (if audit is activated).

**Attention this is alternative step.** You can skip this step and go directly to **Re-Checking User Teacher's Privileges**. If you want to customize the **workflow's expiration date** for future workflows, follow these steps:

- Navigate **Provisioning → Workflows → Definitions → My-Company → Approval → 4-Eye Approval → Control view**.
- Click **Edit** button.
- Change the expiration date and click **OK** button.
- Go to the Tree view on the right side, navigate to **Workflows → Definitions → My-Company → Approval → 4-Eye Approval**, then right-click and select **Load IDS-J Configuration**.
- Start the new workflow. The customized expiration date will now apply.




#### 1.6.5.7.2. Re-Checking User Teacher's Privileges

Now let's use the Web Center to check Mark Teacher's privileges after the approval process has run. To do this, we'll log in as Mark:

- Switch to the Web Center again. Enter **Teacher Mark** in **Name** and Mark's default password (the one you supplied when you reset his password, for example, **1!dirX**).

Web Center opens a dialog that asks you to reset your password because this is Mark's first log in. Follow the instructions displayed on the screen and reset your password, for example to **2!dirX**.

- Select **Self Service** → **Display summary** to get Mark's user summary if not yet displayed.
- Click **Roles**. You can see the **Trainer** role with state ENABLED to indicate that the assignment is active.
- Click **Permissions**. You can see the **Trainer** permission with the state INHERITED, which means that the permission automatically comes from the **Trainer** role assignment. The other permissions come from the other role assignments.
- Click **Groups**. Mark is now member of two more groups (previously he had only 8 group memberships). He has access to the **Training Portal** of the Intranet Portal target system. This group is in the state ENABLED because this target system is configured to use DirX Identity's real-time target system provisioning. As a result, Mark's access to the Training Portal is automatically and immediately provisioned in the connected system. On the other hand, Mark is allowed to work on the **FS Training** file share in the Windows Domain USA. **FS Training** is in the state ADD, which indicates that the target system provisioning has not yet been run. The reason for this is that the Windows Domain USA target system has not been configured for real-time provisioning. As a result, the **FS Training** state will change to ENABLED after we run the target system provisioning.
- Click **Accounts**. You will see four accounts. DirX Identity has configured the accounts for the **Intranet Portal** and **Windows Domain USA** target systems; these accounts are in state ENABLED. The connected system state (State in CS) for **Intranet Portal** is ENABLED, while the state for **Windows Domain USA** is NONE. It will change to ENABLED after we run the target system provisioning.
- There is nothing more to do, so click **Logout**.

In the next exercise, we'll add another role to the user Teacher by making Mark a member of a project.

## 1.6.6. Adding the User to a Project

This exercise shows how role parameters work. The sample domain contains a set of projects that make use of role parameters. We'll make Mark Teacher a member of the MoreCustomers project.

### 1.6.6.1. Assigning the Role with Role Parameters

Because Mark shall be a member of the project MoreCustomers, we'll assign the Project Member role to him. We'll use the DirX Identity Web Center here to set the role by hand. (You can also use DirX Identity Manager.)

- In an Internet browser, enter the URL **http://localhost:8080/webCenter-My-Company** to start the Web Center.
- Enter **Briner Ruben** in **Name** and enter the password. (Remember that all persons in the sample domain have the same password.) Ruben Briner is the project manager of the project MoreCustomers. Thus, he is the person who must assign Mark to his project.
- Click **Log in** (or press RETURN) to log in.

- Enter **T** in the user's quick search bar and press RETURN to return a list of users whose last names begin with "T".
- Position the cursor over the **Teacher Mark** line and select **Assign privileges** from the context menu.
- Enter **P** in **Search for**, and then click **Search** to return a list of roles whose names begin with "P".
- Select the **Project Member** role in the upper pane and then click the down-arrow button between the two panes. Ensure that you do not block popups in your browser.
- The **Parameters** window opens. It shows that the **Project Member** role requires the parameter **Project** to be set. Click the **Project** field and select the **MoreCustomers** role parameter from the drop-down list. Click **Confirm**.
- Now the new role is visible in the lower pane. Click **Save** to store this result.
- There is nothing more to do, so click **Logout**.

#### 1.6.6.2. Checking the Result of the Role Assignment

Now let's check the results of the role assignment with DirX Identity Manager:

- Click **Provisioning** → **Users** and open **My-Company** → **Professional Services** (click the refresh button if it's not visible). Click **Teacher Mark** and then the tab **Assigned Roles**.
- Mark is now a project member of the **MoreCustomers** project. This parameter is visible in the column **Role Parameters**. Alternatively, you can click  at the end of the row. A window opens where you can view all parameters of the assignment of the user Mark Teacher to the role **Project Member**. The role parameter **Project** (**MoreCustomers**) is visible in the **Role Parameters** tab.
- The role parameter is displayed as cn: **cn=MoreCustomers**. This is an example of a role parameter that is derived from an object structure.

Besides the **Trainer** and **Project Member** role Mark Teacher has two additional roles (**Contractor** and **Sales Tasks**) that were automatically assigned by DirX Identity. In the next step, we will show how automatic provisioning works based on the powerful features of DirX Identity.

### 1.6.7. Assigning Privileges Automatically

It might be interesting to see how automatic provisioning works. Because automatic provisioning (as the name implies) works in the background, we can only view the configuration and then run some examples.

In this chapter, we'll explore three important aspects of DirX Identity provisioning:

- Provisioning based on rules
- Inheritance of privileges from business objects
- Influence of user attributes on provisioning results (permission and role parameters)

#### 1.6.7.1. Rule-based Provisioning

We can use DirX Identity Web Center to understand rule-based provisioning. We'll log in as Nik Taspatch, since he is My-Company's role administrator, and then look at the Sales Tasks provisioning rule.

##### 1.6.7.1.1. Checking the Contractor Provisioning Rules

First we'll view the Contractor provisioning rule:

- Start the Web Center by entering the URL **http://localhost:8080/webCenter-My-Company** in your Internet browser (or switch back to it if you left it running).
- Log in as **Taspatch Nik** with the password.
- Select **Rules** → **Provisioning Rules** → **Select rule**.
- Click **Search** to get all rules that are configured for the sample domain.
- Click the **Contractors** provisioning rule in the **Corporate, Role based scenario** folder. This rule grants privileges to users of type contractor (the rule).
- **Filter** shows that the rule starts a search from **My-Company, Users** with the search filter (employeeType=Contractor), which means that it searches for all users with attribute employeeType set to "Contractor".
- **Privileges** show that the role **Contractor** is assigned to these users.
- There is nothing more to do, so click **Logout**.

The definitions for the other provisioning rules are similar. In the next step, we'll learn how these rules can be applied to user objects.

##### 1.6.7.1.2. How Event-based Provisioning Works

The simplest form of event-based provisioning is automatic rule-based provisioning triggered by events. It works as follows:

- Any time you change a user object - either from Web Center, Manager or via an external workflow - DirX Identity's service layer issues a user change event. You cannot see this action because it occurs in the background.
- The event contains the user's distinguished name (DN) and all changes that were performed.
- The EventBasedUserResolution workflow watches for these events and handles this user in the background. This workflow performs a number of checks and procedures, including:
- Checking whether the entry is consistent.
- Applying the matching provisioning rules, assigning new privileges or removing existing privilege assignments based on the new attribute values and resolving the user, if at least one of the attributes is among the domain's permission parameters. These actions can result in additional or changed accounts as well as changes of group memberships.
- Updating the attributes of the user's accounts, if a permission parameter has been

changed. This procedure applies especially to attributes that the user masters (we will learn more about this concept in one of the next sections).

- Updating only the user's account attributes, if no permission parameters are changed but one of a set of configurable attributes is changed (the user is not resolved in this case).
- Storing the entry, if it is modified during this procedure.
- Updating the accounts or groups triggers real-time provisioning of these objects to the connected systems.

In the next sections, we'll explore this concept in more detail.

#### 1.6.7.1.3. Configuring Event-based Provisioning

Enabling event-based provisioning requires a few configuration steps. You need to set up an event policy, define some important parameters at the domain object and set up an event-based workflow for users. These three configuration items define event-based provisioning for a user object in real time.

Setting up Event Policies

You can check the event policy configuration here:

- Log in to the Manager's **Provisioning** view group.
- Open **Policies** → **Event Policies** and then click the **My-Company** policy.
- First, you can see that it is set to active.
- Click **Configuration**.
- In the Selected area, you can see several lines. Some of them have **Send** set to true. This means that changes at organizations, organizational units, users, and SvcTSAccount will produce events.
- The latter is the setting that is important for user changes.

Configuring Permission Parameters at the Domain

Now let's look at the **Domain** objects **Permission Parameter** configuration. These parameters are used by the EventBasedUserResolution workflow to decide whether provisioning rules shall be applied and whether a privilege resolution shall be performed. These parameters can help to reduce the number of time-consuming privilege resolutions enormously.

- Open **Domain Configuration** and then open the **My-Company** top level node.
- Click the **Permission Parameters** tab. Here you can see that the attribute **employeetype** is treated as a permission parameter. This setting indicates that a change to this attribute value enforces a recalculation of the user's privileges and access rights.

Setting up an Event-based Workflow

Next, we'll view the EventBasedUserResolution workflow.

- Log in to the Manager's **Connectivity** view group and then click **Global View**.
- Open the **My-Company** scenario and then click **Main**.
- Right-click on the workflow line between the two **Identity Stores**.
- Select **EventBasedUserResolution** → **Configure** from the context menu. The workflow wizard opens.
- Click **Event Parameters**. This section allows you to define which attribute changes will trigger account updates. Click **Help** if you are interested in more details.
- Click **Cancel** to close the wizard.

#### 1.6.7.1.4. Using Rule-based Provisioning

This section shows how a user attribute influences provisioning via rules.

- Start the Web Center by entering the URL **http://localhost:8080/webCenter-My-Company** in your Internet browser (or switch back to it if you left it running).
- Enter **Bellanger Lionel** in **Name** and enter the password **dirx**.
- Search for **Teacher Mark** either by using the user quick search bar or by using **Users** → **Select user**. Perform a display summary to view the user.
- Click the **Download all tabs** button ( ) from the button group placed left to the Roles button to see all tabs.
- Verify that Mark is an employee of type **Contractor** and that the **Contractor** role is assigned to him.
- Make notes about the assigned permissions and groups.

We assume that Mark's work has been excellent for some time and so the company has decided to offer him the position of internal employee.

- Select **Users** → **Modify user data** and then change the employee type to **Internal**.
- Click **Save** to store the entry.
- Click the **Download all tabs** button ( ) again and view the result.
- After some seconds, the role **Contractor** is replaced with the role **Internal Employee** (click the refresh button several times if this has not yet happened). This is the result of the event that was triggered by the attribute change and that executed the **EventBasedUserResolution** workflow. One of the tasks of this workflow was the execution of all rules.
- Look at the permissions. The **Contractor** permission is replaced with the **Internal Employee** permission and Mark has three other permissions automatically assigned (**Accounting**, **Group File Share** and **Standard Tools**).
- Check the groups. You can see that the **Contractor** group and the **Restricted File Share** groups are in state DELETED (both in the Windows Domain USA). Mark has a set of groups assigned that are required by internal employees.

- There is nothing more to do, so click **Logout**.

This example shows that the change of a single attribute changes the access rights of the employee. The next section shows that a relationship to a business object can have a similar effect.

#### 1.6.7.2. Business Object Inheritance

The primary goal of DirX Identity is to manage users (identities). Business objects like organizational units or locations can simplify this task. This section shows that you can assign privileges, for example a role, to business objects. If a user is linked to such a business object, he inherits these privileges automatically.

##### 1.6.7.2.1. Viewing Business Objects

First, we'll view some business objects:

- Log in to the Manager.
- Select **Provisioning** → **Business Objects**.
- Open **Companies** → **My-Company**. You can see all organizational units of My-Company.
- Click **Finances**. This business object contains a description and a department number (**FI**).
- Click the **References** tab. In **Privileges:**, you can see that the role **Finances Tasks** is assigned to this business object.
- Now click the **Professional Services** and view the **References** tab again. This organizational unit has the **Sales Tasks** role assigned.

All users that are assigned to one of these business objects inherit the assigned role. Let's check this:

- Click **Users** and then open **My-Company** → **Professional Services**.
- Check in the **Assigned Roles** tab that all users in this department have the **Sales Tasks** role assigned.
- Click the **Organization** tab and then check that these users have the organizational unit **Professional Services** assigned. This link is the reason why all of these users have the associated role.

##### 1.6.7.2.2. Assigning an Additional Role

Due to a project release bottleneck in the test group, more testers are required. Hiring additional testers, including training, is a time-consuming procedure. As a result, the management of My-Company decides to have the professional services employees test the products until the bottleneck issue is resolved.

- Log in to the Manager.
- Select **Provisioning** → **Business Objects**.
- Open **Companies** → **My-Company** and click **Professional Services**.

- Click the **References** tab and then **Edit**.
- Add a new line to the **Privileges** section.
- Open the object browser (the last icon in the new row) and select **RoleCatalogue** → **Corporate Roles** → **Department Specific** → **Test Tasks**.
- Click **OK** and then click **OK** again.

After some seconds you can check the users of the Professional Services organizational unit:

- Select **Provisioning** → **Users** → **My-Company** → **Professional Services**.
- Click some of the users, select the **Assigned Roles** tab and then verify that all users have the additional role **Test Tasks** (you may need to refresh this tab). This result shows that a privilege resolution was performed in the background.

The advantages of this method are:

- Changing the name of a business object does not require a change to the corresponding rule filter.
- This method is more intuitive than the rule method.

#### 1.6.7.3. Using Permission and Role Parameters

Permission and role parameters are highly effective methods for reducing privilege structure complexity and eliminating the need for hundreds or thousands of roles and permissions. They help to keep privilege structures simple and easy to use. You can combine permission and role parameters with rule-based provisioning and business object inheritance.

Using permission parameters means that you define a match rule at the permission that selects the assigned groups dynamically based on a user attribute. If this attribute changes, the groups are automatically switched. To select the correct group, the groups must contain these parameter values. We will check the **Sales Tasks** role as an example of permission parameters.

Using role parameters means that parameters are requested during role assignment. These parameter values are stored at the assignment object between user and role and used via a role match rule to select the assigned groups dynamically. Role parameters allow multiple assignment of the same role to a user with different parameters.

##### 1.6.7.3.1. Checking the Sales Tasks Privilege Hierarchy

We will use Web Center's report generator to explore a privilege hierarchy: how groups relate to permissions and how permissions are part of the role definition.

- Log in as **Taspatch Nik** in Web Center.
- Select **Tools** → **Reports**.
- Click the tree icon to the right of **Search base** to select the role in the sample domain's role catalogue on which to run the report. You see **RoleCatalogue**. Open it. Open

**Corporate Roles**, **Department Specific**, and click **Sales Tasks**.

- The system retrieves all reports suitable for an object of this type (a role).
- In **Templates**, click **Privilege hierarchy**.
- After a few seconds, Web Center displays a report on the Sales Tasks privilege hierarchy. You see one **Sales Tasks** role in the **Role** column, and one **Sales Tasks** permission in the **Permission** column that is inherited from the role and defines this role.
- In the **Group** column, you can see a list of the groups that are assigned to the **Sales Tasks** permission. A person who is assigned this permission gets the following groups:
- **CRM Access** in the SAP R3 target system and the group **SAP R3 Client** in the Windows system. These groups allow the user to work via an SAP R3 client in My-Company's CRM application. Because there are two Windows domains, DirX Identity automatically selects the relevant group later on via the match rule definition.
- **Corporate User** in the Extranet Portal target system. This group might allow the user to perform tasks related to the customers in this system; for example, making updates to the portal Web page content.
- **Sales Portal** in the Intranet Portal target system. This group allows a user to access to the internal sales portal.
- **FS Sales** in the Windows target system. This group allows the user to access the group file share of the Sales department. The relevant group (**Windows Domain USA** or **Windows Domain Europe**) is selected later on via the match rule definition, which we'll show you in the next step.

The exact meaning of the groups in the target system is determined by the access rights set in these target systems. The rights must be consistent with the intended effect of the privilege in DirX Identity (this is not controlled by DirX Identity and must be solved and checked with organizational processes).

#### 1.6.7.3.2. Checking the Sales Tasks Permission

Now we'll look at the Sales Tasks permission:

- Select **Permissions** → **Select permission**.
- Enter **S** in **Search for**, and then click **Search** to get a list of permissions that begin with "S". Click **Sales Tasks**. Web Center displays a summary of the permission.
- Look at **Match Rules**. The match rule is used to assign the correct groups. In the match rule shown here, the country (c) attribute of the user is compared to the country attribute(s) of the groups. All groups that match are assigned.

#### 1.6.7.3.3. Checking the FS Sales Group

Finally, we'll examine the FS Sales group:

- Select **Groups** → **Select group**.
- Enter **FS** in **Search for**, and then click **Search** to get a list of groups that begin with "FS".
- Click the **FS Sales** group from the Windows Domain USA target system.Web Center displays a summary of the group.Look at **Permission Parameters**. **Country** is set to **U.S.**

**of America**, which means that this group is assigned to all users from the USA.

- Select **Groups** → **Last selection list: (cn=fs*)**.Click the other FS Sales group.Look again at **Permission Parameters**.All other countries are listed for this group.
- Select **Groups** → **Select group**.This time, search for groups with "Sales".The Sales Portal group is now displayed.. Look at **Permission Parameters**.In this case, **Country** is set to an asterisk (*****) which means that this group is assigned regardless of the country in which the user works.
- There is nothing more to do, so click **Logout**.

Now we have completed the task of adding a user to DirX Identity.This exercise has demonstrated how to create a user in the DirX Identity store and how to assign privileges in various ways that provision the user in the connected systems.

# 1.7. Importing Identities

My-Company has bought another company named New-Company that supports its own Human Resources (HR) database (delivers a CSV formatted file) and several target systems.In this section, we show you how to import new identities from an external source - in this case, the New-HR database - either once or on a regular basis.First, we tour the default connectivity scenario supplied with DirX Identity to become familiar with the DirX Identity Manager's Connectivity view and the configuration concepts of connected directory, workflow, activities and jobs.

Next, we configure New-HR as the source for the identity import and create, configure, run, and monitor the synchronization workflows that import the identities in New-HR into My-Company's Identity Store.

At last, we analyze the imported user entries and the automatically assigned privileges.

## 1.7.1. Viewing the Default Connectivity Scenario

The default connectivity scenario is a sample Connectivity configuration supplied with DirX Identity to illustrate configuration concepts and to streamline the process of creating new Connectivity configurations.The Default scenario provides template scenarios for identity creation, identity provisioning, and target system maintenance and provides Connectivity configuration template objects that support these template scenarios.Let's examine the Default scenario to understand its structure.

- Select the **Connectivity** view in DirX Identity Manager (we assume you have already logged in.)

DirX Identity Manager opens the Global View and displays the Default scenario in this view. The Global View is a high-level representation of Connectivity configuration that permits you to perform all the basic Connectivity configuration tasks without having to deal with the underlying complexities of the DirX Identity system. You can read more about the Global View in "Using the Global View" in the *DirX Identity User Interface Guide*.

In this part of the exercise, we'll:

- View the default connectivity scenario's connected directories and workflows
- Use the Global View to look at the structure of a representative workflow
- Create a report on the workflow to view its structure in another way

#### 1.7.1.1. Viewing Connected Directories and Workflows

In the Global View, a connectivity scenario consists of connected directories (the "tin" icons) and workflows (the lines between the connected directory icons). DirX Identity represents all input and output sources, whether they are actual target systems, virtual target systems, or file-based sources like history and report files or LDIF or CSV files, and even the identity store itself - as connected directories. The workflows are the processes that synchronize the data between the connected directories. The Default scenario consists of four separate scenarios: the Identity Store scenario, which contains workflows that maintain these identities in the identity store, the Source Scheduled scenario, which contains the workflows that import identities either once or regularly to the Identity Store, the Target Realtime scenario, which contains the event driven realtime workflows that synchronize and validate the connected target systems, and the Target Scheduled scenario, which contains the Tcl-based target system provisioning workflows.

- Click the **Source Scheduled** scenario. You can see icons for all pre-configured connected directories and blue lines in between that represent the pre-configured workflows. Almost all lines start from the central connected directory **Identity Store**. Most lines have arrows at both ends, some have an arrow at only one end. The arrows give hints about the direction in which the synchronization flows.
- Now click the line between **HR-ODBC** and **Identity Store** with the left mouse button. The line becomes orange.
- Right-click the line between **HR-ODBC** and **Identity Store**. You can see the menu options **New** and **Assign**, which allow you to create or assign new synchronization workflows, and a list of synchronization workflows: **Ident2ODBC, ODBC2Ident** and **ODBC_Ident**. The last workflow is a nested workflow that simply runs other two workflows. The **ODBC2Ident** and **Ident2ODBC** workflows are not nested. As you can see, a workflow line can contain any number of workflows.

#### 1.7.1.2. Viewing a Workflow's Structure with the Global View

Now we'll look at the structure of one of the default connectivity scenario's workflows. In the Global View:

- Right-click the line between the **HR-ODBC** and **Identity Store** connected directories.
- Select **ODBC2Ident** → **Show structure**.

A window opens that displays information about the workflow's structure.

The two fields at the top show the workflow (**ODBC2Ident**) and the associated DirX Identity server where this workflow will run. The table below these fields shows the complete workflow structure.

The **Activity** column contains the steps - called activities - that the workflow will perform in sequential order. The activities here are **ODBC2Ident_ODBCExport** and

**ODBC2Ident_metaCP**. The **Identity Server** column specifies the DirX Identity server on which these activities will be performed - here, it is your main server. The **Job/Workflow** column defines the object to be run, either a job (which is a configured agent) or a workflow (if the workflow is nested). The **Channel**, **D**(direction) and **Connected directory** columns show the data flow. For **ODBC2Ident**, data is read from the connected directory **HR-ODBC** via the channel **ODBC2Ident** into the job **ODBC2Ident_ODBCExport**. Data is processed and then stored via the channel **OutData** into the intermediate file-based connected directory **Data**. The next job **ODBC2Ident_metaCP** reads the data via the channel **InData,** processes it and stores it in the **Identity Store** via the channel **ODBC2Ident**. You can learn more about activities, jobs, channels, and agents in the *DirX Identity Connectivity Administration Guide*.

You can see that the structure view presents all elements of a synchronization workflow. You can view more details if you click **Edit**, but at this point it does not make sense to go into too much detail.

- Click **Close** to exit the structure view.

#### 1.7.1.3. Viewing a Workflow's Structure with a Report

Another way to view a connectivity scenario and its parts is to use the DirX Identity report generator:

- In DirX Identity Manager's **Connectivity** view, click **Expert View**
- Open **Workflows** → **Default** → **Target Scheduled** → **ADS** → **ADS2Ident_Validation.**
- Right-click **ADS2Ident_Validation** and select **Report**. A dialog opens.
- Select **Workflows** as a report template and check **Output to viewer**.
- Click **Run report**. After some time, the report result is displayed. The report shows all the parameters that are set for this workflow and its related objects and illustrates the workflow's control and data flow. You can see that this workflow consists of four activities. The first activity reads account data from the ADS connected directory and stores it into an intermediate file (**Data**). This file is read by the meta controller (the DirX Identity agent that synchronizes LDAP- and file-based connected directories), which compares this information with the content of the ADS target system in the Identity Store. The third and fourth activities perform the same sequence for groups.

You can also write your reports in XML or HTML format into an external file; to do this, specify a file name and do not check **Output to viewer** in the Report dialog. You can use these files as the basis for your project documentation.

As you can see, these reports can contain an enormous amount of information. You can customize this easily; see Customizing Status Reports for details.

### 1.7.2. Creating a New Scenario

In this section, we assume that you have loaded the pre-defined My-Company Main connectivity scenario as described in "Loading the Connectivity Scenario" in "Preparing to Use the Quick Start". We'll use parts of this scenario as the basis for our new scenario, which we'll name NewCompany.

#### 1.7.2.1. The Main Scenario

In DirX Identity Manager → **Connectivity** → **Global View**, use the directory tree in the middle pane to open **Scenarios** → **My-Company** → **Main**. You can see that this scenario consists of:

- An **Identity Store** icon that represents the My-Company identity store. This icon is shown twice to enable a workflow line in between. Right-click the workflow line and verify that **ConsistencyCheck**, **PolicyExecution** and **PrivilegeResolution** workflows are available for selection.
- Connected directories for audit trail database and report files and their corresponding workflows (right-click the workflow lines to see these workflows).
- Connected directories for each target system that exists in the My-Company organization (**Extranet Portal** through **Windows US**). The connected workflow lines here contain validation and synchronization workflows.

The next step is to add the new identity source. We have two options:

- We can integrate the new identity source into the existing My-Company Main scenario
- We can create another scenario that contains only the identity creation connected directories and workflows available in My-Company Main.

Because the second choice allows us to use more features of DirX Identity, we'll use it and set the name of the new scenario to NewCompany.

#### 1.7.2.2. Creating the NewCompany Scenario

To create the NewCompany scenario from the My-Company Main scenario:

- In **Connectivity**, click **Global View**
- Right-click the **My-Company** folder in the middle pane, and then select **New** → **Scenario**. Enter **NewCompany** into **Name**, enter **Additional scenario for My-Company** in **Description**, check **Use Grid** and enter **10** into **Grid-X** and **Grid-Y**. Click **OK**.

You have successfully created a new scenario that you can now use to set up the new identity source.

### 1.7.3. Adding a New Identity Source

This section shows you how to set up a new connected directory that models the external New-HR database. In this section, we:

- Set up the actual New-HR ODBC database
- Add the identity store to the new scenario
- Create and configure the New-HR connected directory

#### 1.7.3.1. Setting up the ODBC Database

Before we can run a real example of an identity source, we need to set up a real New-HR ODBC database. We will perform this step on Windows Server 2012 R2 for a Microsoft

Access database. For other versions of Windows, the procedure for finding the Data Sources (ODBC) is slightly different.

##### 1.7.3.1.1. Copying the PreDefined ODBC Database

First, we'll use the file system to copy the prepared database **new-hr.mdb** from the DirX Identity *install_path***data\extension** directory to the folder **C:\MetahubData**.

##### 1.7.3.1.2. Creating the ODBC Data Source

Now we'll create the new data source for the New-HR identity source. In Windows Server 2012 R2:

- Open **Server Manger Application**.
- Choose **Tools**.
- Choose **ODBC Data Source (32-bit)**.
- Click the **System DSN** tab.
- Click **Add** and select **Microsoft Access Driver (*.mdb)** from the list.
- Click **Finish**.
- Enter **new-hr** as **Data Source Name**. This is an important parameter that the DirX Identity ODBC agent uses later on to access the database.
- In the Database area, click **Select**. Locate the file **C:\MetahubData\new-hr.mdb** and click **OK**.
- Click **OK** twice to close the application.

Now we have prepared the new data source **new-hr**.

#### 1.7.3.2. Assigning the Identity Store

We now need to create a representation of the identity store in our (blank) scenario. DirX Identity Manager supports two ways to add existing connected directories to a scenario. You can either:

- Create a copy, which creates a copy of the original connected directory that can then be individually configured.
- Assign the existing instance, which creates a link to the original connected directory. If you then configure this link, you will change the original.

We will use the assign method because we want to use the existing identity store definition as it is. In DirX Identity Manager → **Connectivity** → **Global View** → **Scenarios** → **My-Company** → **NewCompany**:

- Right-click in an empty area of the right pane, and then select **New Connected Directory**. This action displays a crossbar cursor in the pane.
- Move the crossbar cursor to the location in the pane at which you want to position the new connected directory, and then click the left mouse button. This action creates a connected directory icon for the new connected directory with the default name (**new**).

- Right-click and then select **Assign**. The dialog **Select connected directory** opens.
- There are two **Identity Store** entries in the list. Select the one from the folder **My-Company/Main/Identity Store** and click **OK**.

This action creates the link to the original identity store of the My-Company scenario.

#### 1.7.3.3. Creating the New Connected Directory

Next, we create the new identity source New-HR. In DirX Identity Manager → **Connectivity** → **Global View** → **Scenarios** → **My-Company** → **NewCompany**:

- Right-click in an empty area of the right pane, and then select **New Connected Directory**. This action displays a crossbar cursor in the pane.
- Move the crossbar cursor to the location in the pane at which you want to position the new connected directory, and then click the left mouse button. This action creates a connected directory icon for the new connected directory with the default name (**new**).

The next step is to start a DirX Identity Manager configuration wizard to set up the New-HR connected directory.

#### 1.7.3.4. Using the Configuration Wizards

Before we start to configure the New-HR connected directory, let's take a quick tour of the Configuration Wizards. The Global View provides two configuration wizards:

- The Configure Connected Directory wizard
- The Configure Workflow wizard

These wizards function like many Windows-based software installation wizards (with some nice enhancements).

The top of each wizard dialog describes the current configuration step to be performed in the wizard. The left side of the dialog displays all of the configuration steps contained in the wizard. Tasks highlighted in green indicate completed steps. Tasks highlighted in red indicate outstanding steps. The current step is highlighted in grey. The right side of the dialog displays the work area for the current configuration step. The content of the area differs depending on the step at hand.

Use the buttons at the bottom of the dialog to move between wizard dialogs and exit the wizard:

- Click **Next** to move to the next dialog.
- Click **Previous** to move to the previous dialog.
- Click **Finish** when you reach the last wizard step to save your configuration changes and exit the wizard.
- Click **Cancel** to exit the wizard.
- Click **Help** to get help on the current wizard dialog.

After the first configuration of an object with a wizard, you can click directly on one of the task buttons on the left side to move quickly to the step you want to re-configure. This feature allows very fast but wizard based reconfiguration of your objects.

Click **Cancel** to close the open wizard.

#### 1.7.3.5. Configuring the New Connected Directory

We'll now use the Create a Connected Directory wizard to create the New-HR connected directory. To start the wizard, right-click the new connected directory icon and select **Configure ...**. We'll use this wizard to:

- Select a template for New-HR
- Supply general information about New-HR, including its name
- Check the attribute configuration
- Check the bind parameters
- Set operational attributes
- Name the connected directory

##### 1.7.3.5.1. Selecting a Connected Directory Template

This step allows you to select a template directory configuration to apply to the new connected directory. You can use the Directory Type, the Description fields and the folder information in the list to locate the connected directory template that you want to use. Because we want to set up an HR identity source, we use the HR-ODBC template (and not the ODBC template that is used when setting up an ODBC target system).

- Click **HR-ODBC** in the list to select it. This action selects a template ODBC database that the wizard uses as a base for your connected directory.
- Click **Next**.

The **Next steps ...** icon in the left-hand pane is replaced by the individual steps that are necessary to configure New-HR.

##### 1.7.3.5.2. Supplying General Information

This step allows you to define general parameters for the connected directory. The wizard automatically supplies a new connected directory name (HR-ODBC) and supplies values for the other general parameters.

- Enter **New-HR** in **Name**.
- Click **Help** to view the definitions of all the fields that are visible in the dialog.

Each step provides detailed help information. Because individual objects can have more or fewer fields displayed (there can be hidden fields that have no meaning for this object instance), there might be more fields described in the help than are visible in the dialog you're

viewing.

- Close the Help window.

The service field shows where your physical connected directory (in this case, the ODBC database) is located. In this quick start scenario, New-HR is located on the service HR ODBC Service. We'll now change this to the New-HR Service:

- Click to open the service object.
- Change **Name** from **HR ODBC Service** to **New-HR Service**.
- Change **Server Name** from **personal** to **new-hr**. This is the name of the Microsoft Windows ODBC system data source we created in "Setting up the ODBC Database".
- Click **Apply**.

Next we should adapt the viewer command, which allows you to open the connected directory with a viewer and to edit applications after setup.

- In this case, the viewer is the Microsoft Access application (C:\Program Files\Microsoft Office\Office11\MSACCESS.EXE). Change this path if it is not correct on your system.
- Change the file to be viewed from **personal.mdb** to **new-hr.mdb**.
- Click **Next**.

#### 1.7.3.5.3. Updating the Attribute Configuration

This step allows you to view and edit the attribute configuration file, which provides the format description of your connected directory. Use the **Global Info** tab to view the definition of parameters that are valid for the entire file; for example, the record separator. Use the **Attribute List** to view each attribute definition and its corresponding values. You can read more about attribute configuration files in the *DirX Identity Connectivity Administration Guide* and in the *DirX Identity Meta Controller Reference*.

We need to update the ODBC template attribute definitions shown here to reflect the attribute format for our New-HR database. Although you can edit the various lines in the table by hand, we'll import the necessary attribute configuration definitions from a file to keep it simple.

- Click **Import CFG File**.
- Select the file **new-hr_attrConf.cfg** from the folder *install_path***data\extension**.
- Click **Import**. The new definition is loaded. It defines the intermediate format of the file that is used for data transfer between the ODBC agent and the meta controller. Each line describes an attribute in the **new-hr** ODBC database; for example, Department and EntryDate attributes.
- Click **Next**.

#### 1.7.3.5.4. Checking the Bind Parameters

This dialog allows you to change the properties of the bind profile that the ODBC agent

uses to access the ODBC database. This step contains only one bind profile.

- Click the ODBC Admin line and then click .
- By default, the password is displayed in a scrambled form. It is set to a default value and this is the password necessary to access the **new-hr.mdb** database. You can change the password of the **new-hr.mdb** database to another value, but you must then change the value here accordingly
- Click **Close** and then **Next**.

#### 1.7.3.5.5. Supplying Operational Attributes

In this step, you can adjust the operational parameters master name and the attributes for the global unique ID generation, which are used for local GUID generation. In this case, the GUID is composed of the GUID prefix - here the fixed string **OD** which must be unique in the entire scenario - and a unique key of the source connected directory - in this case, the employeeID. We change these settings:

- Set **Master Name** to **NEWHR**. All entries that are created by this workflow will be marked with this value. With this technique, you can easily see who's the master of an entry in the identity store.
- Change **GUID Prefix** from **OD** to **NH** and set **Local GUID Attribute** to **PersonalNr**. DirX Identity will create global unique IDs that are composed of the fixed string **NH** and the personal number (for example, NH00001).
- Click **Next**.

#### 1.7.3.5.6. Naming the Connected Directory

This step allows you to name the new connected directory if you haven't already done so in the first wizard step.

- Leave everything in this dialog as it is.
- Click **Finish** to save the configuration information for this new connected directory and exit the wizard.

The wizard changes the default name of the new connected directory to **New-HR** and displays it on the new connected directory icon in the Global View.

### 1.7.3.6. Checking the Database Viewer

Now we should check to see if the viewer we specified during the configuration process is working correctly.

- In the **NewCompany** scenario (**Connectivity** → **Global View** → **My-Company** → **NewCompany**), right-click **New-HR**, then select **Open**.
- The Microsoft Access tool opens and asks for the password.

The viewer does not use the bind profile. This example shows that DirX Identity can be used as a central user interface to view and control the

content of connected directories.

- Open the EMPL table and view its content. This is the table from which the data is extracted.

The table contains a column DepartmentNo.

- Close the application.

Now we have two connected directories in our new scenario. We need a workflow line to connect them.

### 1.7.4. Creating a New Workflow

Now we'll set up an import workflow between the New-HR connected directory and the Identity Store. First, we'll create the workflow in the DirX Identity Manager Global View, and then we'll configure it with the Workflow Configuration wizard. To create the workflow:

- Go to **Connectivity** → **Global View** → **Scenarios** → **My-Company** → **NewCompany**.
- Right-click in an empty area of the NewCompany scenario view pane, and then select **New Workflow Line**. This action displays a crossbar cursor in the view pane.
- Move the crossbar cursor onto the **New-HR** connected directory icon, and then click the icon. A line attaches to the cursor.
- Move the crossbar cursor with the line onto the **Identity Store** connected directory icon, and then click the icon.

You have now created a new workflow line between the **Identity Store** and the **New-HR** connected directories. This line allows you to establish workflows between these two connected directories.

### 1.7.5. Configuring the New Workflow

Now we'll use the Create a Workflow wizard to configure the workflow line we just created into a workflow that exports information from the New-HR connected directory to the identity store.

- To start the wizard, right-click on the new workflow line, and then select **New**.

We'll use the wizard to:

- Select a template to use for the workflow
- Supply general information about the workflow, including its name
- Select the attributes to be exported from the New-HR connected directory.
- Select the attributes in the Identity Store to which the selected New-HR attributes are to be mapped
- Configure the attribute mapping between the New-HR and identity store attributes
- Check the workflow's export properties

- Check the delta handling for the workflow
- Check the export tracing parameters
- Check the operational properties
- Check the import properties
- Check the entry handling properties
- Check the import tracing parameters

#### 1.7.5.1. Selecting the Workflow Template

This step allows you to select a template workflow configuration to apply to the workflow. The workflow templates that are available for selection depend on the connected directories at each end of the workflow line. In this example, they are the **Identity Store** with type **LDAP** and the **New-HR** directory with type **ODBC**. The wizard displays the workflow templates (creation, synchronization, and validation) that you can use for these directories.

- Click **ODBC2Ident** in the list of workflow types. This action selects the workflow template for importing data from an ODBC database to your identity store.
- Click **Next**.

If your workflow works between two connected directories of the same type (for example the **Meta2LDAPdir** workflow between two LDAP directories), the workflow wizard displays a **Define Direction** dialog and displays a proposal for the **Source** and **Target** directory. If this selection is not okay, click **Reverse Direction**.

You cannot return to the **Define Direction** dialog. Instead you must exit the workflow wizard with **Cancel** and restart it from the beginning. If the direction proposal is okay, click **Next**.

Now the **Next steps ...** icon is replaced by the individual steps that are necessary to configure the ODBC2Ident workflow.

#### 1.7.5.2. Supplying Workflow General Information

This step allows you to set general parameters for the workflow, such as its name and its description (you can also set the name at the end of the wizard).

- Enter **NewHR2Ident** into **Name**.
- Add a description like **Workflow to maintain identities from New-HR** to **Description**.
- Click **Next**.

#### 1.7.5.3. Selecting the Attributes to be Exported

This step (Source Selected Attributes) allows you to choose the attributes that the workflow is to export from the New-HR connected directory. The workflow template selects a default

set of attributes from the attribute configuration defined in the template ODBC connected directory. Because we replaced this default attribute configuration with our New-HR attribute configuration when we configured the New-HR connected directory, some of the selected source attributes on the right side are displayed in red because they do not exist in the New-HR database; for example, **EmployeeID**. We need to remove the default attributes from the selected source attribute list that aren't supported in the New-HR attribute configuration and add the New-HR attributes that correspond to the default ones. To make these changes:

- Click **EmployeeID** on the right side and then click < . The red field is removed from the right pane.
- Click **EMPL.PersonalNr** on the left side and then click > . The field **EMPL.PersonalNr** is displayed on the right side as the last field.
- Perform the same procedure for the rest of the attributes. Exchange **GivenName** with **EMPL.FirstName**, **Surname** with **EMPL.LastName, Country** with **EMPL.Land** (the German term for country).
- Click **EMPL.DepartmentNo** on the left side and then click > .
- The Role and Type attributes do not have corresponding attributes on the left side. Simply remove them. Click **Role**, hold down the CTRL key and click **Type**. Both attributes are now selected. Click < to remove them. Now there are no more red attributes displayed.
- Click **Next**.

#### 1.7.5.4. Selecting the Mapping Attributes

This step (Target Selected Attributes) allows you to choose the attributes from the target connected directory (the Identity Store directory) to be used for the mapping step. The template selects a default set of attributes, and you can use this dialog to tailor the default selection to your requirements:

- Click **dxrOULink** on the left side and then click > . The field **dxrOULink** is displayed on the right side as the last field.
- Click **dxrOrganizationLink** on the left side and then click > . The field **dxrOrganizationLink** is displayed on the right side as the last field.
- Click **Next**.

Now you have selected the attributes from the source and target directory that you want to use. If you find that you need to change these selections later on in the workflow configuration process, click **Previous** to return to these dialogs.

The next step is to map the source attributes into target attributes.

#### 1.7.5.5. Configuring the Attribute Mapping

This step (Attribute Mapping) allows you to:

- Define how the attributes you have selected for synchronization will be mapped from

the source connected directory to the target connected directory

- Define the mapping functions that will be used to carry out the mapping for each individual attribute set

The workflow template supplies a default attribute mapping between source and target directories. You can use this dialog to make changes to the template. The topic "Using the Mapping Editor" in the *DirX Identity Connectivity Administration Guide* describes how to use the dialog.

For this workflow, the dialog shows the template for how New-HR attributes (attributes with the prefix **file**) will be mapped to the Identity Store attributes (attributes with the prefix **ldap**) that you have selected for synchronization. It also shows the mapping functions that DirX Identity will use to carry out the mapping. Notice that lStringEscape is often used. lDNcreate is a more complex function; it allows DirX Identity to build the distinguished name (DN) for each of the entries. In this case, two constant strings ("cn" and "ou") are combined with two variables (cn and ou) and are then extended with a constant string base_obj.

Note the red fields in the **Input Arguments** column. These are the same attributes that we corrected in the source selected attributes step. We need to correct them here, too:

- Click into the field **file.GivenName**. Select **file.EMPL.FirstName** from the drop-down list in that field.
- Perform the same procedure for these fields: **file.Surname** to **file.EMPL.LastName**, **file.EmployeeID** to **file.EMPL.PersonalNr** and **file.Country** to **file.EMPL.Land**.
- Scroll down to exchange the **file.Surname** and **file.GivenName** attributes that are arguments of the lStringCompose function.
- Click the **file.Role** field. Click to delete the entire column.
- Click the **file.Type** field. Replace its content with **"Internal"** (do not forget to enter the quotes into the text field!). This value means that the field **ldap.employeeType** is filled with the constant "Internal" because all the identities coming from New-HR are treated as internal employees.
- Scroll up and down the mapping list to make sure that no red attributes remain You will find **file.Country** again. Replace it with **file.EMPL.Land**.
- Locate the line that composes the cn (see the **Output** column). This is an example of a predefined mapping function **lStringCompose**. It combines the two attributes LastName and FirstName from the file and separates them with a blank. If one of these attributes is empty, it does not add the blank. The value is then stored in a variable 'cn' to be used in the next function.
- The function lDNcreate creates the distinguished name of the entry. It is composed of the previously calculated cn and the ou and o parts and the base_obj (the constant part of the dn, in this case **cn=Users,cn=My-Company**).
- Next we want to create a link to the organizational unit business object **Product Testing**. Use also the lDNcreate function and build the **ldap.dxrOULink** attribute:

It consists of the elements "ou", **file.EMPL.Department**, "o=My-Company,cn=Companies,cn=BusinessObjects,cn=My-Company".
This should create a link like: "ou=Product Testing,o=My-Company,cn=Companies,cn=BusinessObjects,cn=My-Company".

- We also want to create a link to the organization business object. All these users are employees of My-Company. Use the "=" function to store the constant value "o=My-Company,cn=Companies,cn=BusinessObjects,cn=My-Company" in the **ldap.dxrOrganizationLink** attribute. Don’t forget to enter the values, including the quotes!
- Click **Next** to proceed to the next step.

#### 1.7.5.6. Setting the Export Properties

This step allows you to set the export parameters for the ODBC agent. We need to change these parameters to reflect the New-HR database:

- The workflow template uses the table HR. Our New-HR database uses the table **EMPL**. Set this value in **From**.
- Replace **GivenName, Surname** with **FirstName, LastName**. This action sets the parameters used for the delta handling index (hash file) when using delta mode (which we are not using at the moment).
- Click **Next**.

#### 1.7.5.7. Setting Delta Handling

This dialog shows the delta handling parameters. Delta synchronization is not enabled.

- Do not change these settings and click **Next**.

#### 1.7.5.8. Setting the Export Trace Parameters

This step allows you to define the export tracing parameters.

- Do not change these settings and click **Next**.

#### 1.7.5.9. Setting Operational Parameters

This step allows you to influence the behavior of the workflow.

The second group of properties allows you to control the workflow’s notification behavior (click **Help** for more information).

The third group of properties defines the GUID generation method. By default, **Local-Local GUID Generation** is selected as **GUID Generation Type**. This setting means that the GUID is composed of a fixed string (the GUID prefix you configured in the New-HR connected directory) and a unique key attribute from the source connected directory (the Local GUID attribute, also set in the New-HR connected directory).

The fourth group of properties contains control parameters. You can use **Minimum Source**

**Entries** and **Exact Action** to influence the behavior of the workflow when strange situations occur. Click **Help** if you want more information.

The last group of properties helps to debug your workflow during the design phase. **Init Mode**, **Test Mapping Only** and **Test Max Entries** are available options.

You can leave the default values as they are, so:

- Click **Next** to use the current set of properties.

#### 1.7.5.10. Setting Import Properties

This step permits you to specify the parameters for the identity import operation. The dialog provides several groups of properties that control the basic behavior of the user creation workflow.

The first group contains the most important properties: **Base Object** and **Subset**. These two properties define the subtree into which the data is to be imported. **Base Object** is defined as a reference:

<?Job@OutputChannel-DN@ConnDir-DN@SpecificAttributes(Role_user_base)/>

It fetches the value from the attribute **Role_user_base** in the Provisioning Parameters step of the target connected directory (in this case, the identity store). You can read more about references in the *DirX Identity Connectivity Administration Guide*.

**Object Class Collection** defines the type of object to be handled. This value must correspond to one of the values in the first column of the **Object Classes** field of the **Operational Attributes** tab of the connected directory. If nothing is specified, "users" is assumed.

The next parameter sets the **Import Mode**. **Merge** (which is the selected option) means that the entries from the ODBC database are merged into the target directory. **Replace** means that the existing set of entries in the target directory is replaced with the set of entries from the ODBC database. This action guarantees identical entries in the source and target directory but needs a lot of time to run if the number of identities to be imported is very large.

Depending on the **Import Mode** that is set, the **Filters for Merge** or **Filters for Replace** properties must be specified. In this case, **Filters for Merge** is the relevant set. The **employeeNumber** is used as the **Join Expression** to find the corresponding entry in the identity store.

The other default values currently shown can be left unchanged for the moment.

- Click **Next** to use the current set of properties.

#### 1.7.5.11. Setting Entry-Handling Properties

This step lets you define detailed parameters for the entry operations add, modify and delete.

The **Add**, **Modification** or **Deletion Properties** fields let you control add, modify and delete entry operations. You can choose to prevent the operation from being performed, to allow it to execute, to notify an administrator but prevent the operation from being performed, or to allow the operation to execute and send a notification about it. The settings you make here may allow you to set additional properties. Click **Help** if you want more details. In our case, **Add** and **Modification Properties** are enabled, and **Deletion Properties** is not. This selection means that entries that do not exist will be created and existing entries will be modified. We can leave all the default values shown, so:

- Click **Next** to use the current set of properties.

#### 1.7.5.12. Setting Import Tracing Parameters

This step allows you to set tracing and debugging parameters for the workflow. The parameters to be configured in this dialog depend on the specific target connected directory and the DirX Identity agent that is used for the operation.

In this case you can set the corresponding **Trace file,** the **Debug Trace**, the **Trace Level** and other parameters of the meta controller, which is used as the agent here. Click **Help** if you want more information.

- Click **Next** to use the defined tracing parameters.

#### 1.7.5.13. Naming the Workflow

This step allows you to name the new workflow:

- Because you entered the name earlier, leave everything as it is.
- Click **Finish** to save the configuration information for this new workflow and exit the wizard.

After a refresh, the wizard displays an arrow on the workflow line in the correct direction to indicate that one or more workflows for this direction are configured. To view the configured workflows, click the workflow line to select it and then right-click to view the context menu. The menu lists the **New** and **Assign** selections and all configured workflows. In this case, only **NewHR2Ident** is displayed.

You have now set up two connected directories and a workflow in the NewCompany scenario. The next step is to test the workflow.

### 1.7.6. Running the New Workflow

In this section, we'll first look at the existing Product Testing users in the identity store, and then we'll run our new NewHR2Ident workflow to load 9 new users from the New-HR connected directory.

#### 1.7.6.1. Checking Product Testing Users

First we'll check the current data content of the identity store to compare it with the result of the workflow run. In DirX Identity Manager → **Provisioning** → **Users**:

- Open the tree **Users** → **My-Company** → **Product Testing**. There should be 6 user entries and one user facet entry. Our new workflow will add 9 new user entries.

#### 1.7.6.2. Running the Workflow

Now you can start the workflow run. In DirX Identity Manager → **Connectivity** → **Global View** → **Scenarios** → **My-Company** → **NewCompany**:

- Click the workflow line between the two connected directory icons to select it.
- Right-click on the line, select **NewHR2Ident**, and then select **Run**. This action starts the workflow and opens a separate status dialog window. The status dialog displays a series of status messages and a status bar that shows the progress of the **NewHR2Ident** run. You could abort the workflow run with **Abort Workflow**. However, we want to wait until the blue progress bar reaches the end, which indicates that the workflow has run successfully. (When the workflow run is not successful, the color of the progress bar changes to red and the bar stops).

You can click **Details** in the status dialog to display the changing status messages in a sub-window, but this information isn't relevant to this part of the quick start.

- Click the **Structure** tab in the running workflow's open status window. Two icons for the two activities of the workflow are displayed. The icons change color during the workflow run from yellow (not yet started) to blue (running). After the run, the icons should either be green (the workflow run was successful) or red (the run was not successful). When the icon is light red, it indicates that minor problems occurred (warning). Perhaps some entries could not be processed correctly but most of the workflow ran correctly.

*Figure 3. Running Workflow Structure Window*

Leave the Structure window open for now (do not click **Close**). We'll need it for the next exercise.

## 1.7.7. Monitoring the Workflow Run

Now we'll demonstrate DirX Identity's monitoring features. DirX Identity provides two ways to monitor a workflow run: from the running workflow's Structure tab, and from DirX Identity Manager's connectivity Monitor View. We'll show you how to use these features, and we'll show you how to manage the generated status information.

### 1.7.7.1. Monitoring from the Structure Tab

First, we'll use the running workflow structure window we left open in the last exercise to show you how to view status information from the open run window directly after a workflow run:

- Click the **Structure** tab in the open status window from the **NewHR2Ident** workflow run if you haven't done so already.
- Double-click the second activity **ODBC2Ident_metaCP**. A new dialog opens and shows the status information from this activity. Its name is **ODBC2Ident_metaCP 20120620141822Z** (the time stamp value shown here might differ from yours).

Now we'll look at the status information provided for this activity.

#### 1.7.7.1.1. Viewing Activity Status Data

Click the **Activity Status Data** tab (if it is not already displayed). This tab displays some general information, for example the start and end time of this run and the result (Status) which is **closed.completed.ok**. You can also see the exit code the agent returned to DirX Identity. For details, click **Help**.

#### 1.7.7.1.2. Viewing Input/Output Data

Now click one of the other tabs to display the files for this specific workflow run that have been saved in the status area. The saved files can be configuration files ("ini" files or script files) in the **Config** tab, input and output files in the **Input/Output** tab and trace or report files in the **Trace** tab.

- Select one of the list entries and then click to view the content of each file.

You can select the editor with which you view the file with the **file.editor** property in the **dxi.cfg** file. See Customizing DirX Identity Manager for details.

- In the **Input/Output** tab, click the entry **data.txt** in **Input** and then click (if you leave the cursor over this button, the help pop-up text **Displays the file content** is displayed for a few seconds). This action displays the TXT file content that was imported into the directory. Close the editor window.

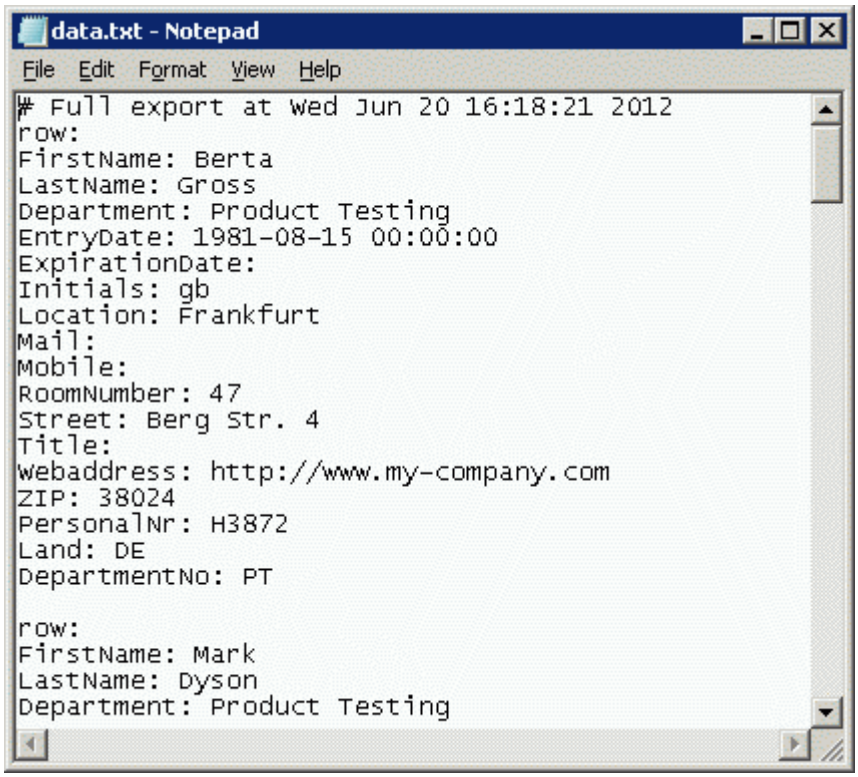



*Figure 4. TXT File Content*

#### 1.7.7.1.3. Viewing Trace Information

The **Trace** tab contains a **Process Data** field that contains information about the workflow run that the DirX Identity server has generated. The entry shows the executable that has been run and the corresponding command line parameters.

- Click the **Trace** tab, then click to the right of **Process Data:**. You can see a lot of information like the **AgentRunTime** (the job needed about 6 seconds), the generated **CommandLine** and the **Executable** that was run. In this case, the meta controller (**metacp**) was started. Close the editor window.
- Now select the **trace.trc** file in the **Trace:** field. Click to open the trace file. Because minimum trace information was configured and no error occurred during the workflow run, the statistics are the interesting part of the information here. The meta controller reports **Number of file entries**: 9 and **Successful Additions**: 9.

#### 1.7.7.1.4. Viewing Statistics

Click the **Statistics** tab, which displays statistics information from the **metacp** trace file in tabular and XML-type representations. Only the DirX Identity meta controller outputs information to this tab.

### 1.7.7.2. Monitoring from the Monitor View

You can also use DirX Identity Manager's connectivity Monitor View to examine the results of the NewHR2Ident workflow run in detail at any time:

- Click **Connectivity** → **Monitor View**.
- Open the folders **My-Company** → **NewCompany** → **Source Scheduled** → **ODBC** → **NewHR2Ident**, which contains all the status entries that result from runs of this workflow.
- Double-click the **NewHR2Ident** status entry in the tree with the relevant date and time. The status entries for the activities that comprise this workflow are shown in the tree and at the top of the right-hand pane, while the properties of the workflow status entry are shown in the lower part of the window. You can move the horizontal separator bar up to enlarge the lower window and view its entire content.
- Look at the start and end times and the result **closed.completed.ok**. If an error occurs during the run, the status will be **closed.completed.error**, and if warnings are detected, the status will be **closed.completed.warning**.
- Click the activity status entry under the workflow status entry in the tree or in the list at the top of the right-hand pane to display its properties. This is the information we already checked from the run window.
- Click the workflow status entry again and then the **Statistics** tab. It contains the same information as the **metacp** activity status entry. Workflow entries collect the statistics information from all contained activities even if these activities represent nested workflows.

DirX Identity stores all of these status files in a safe place that you can configure (you did it

during installation but you can change it easily for each DirX Identity server using its corresponding DirX Identity server configuration object). The next run of this workflow will not delete these files, because DirX Identity has automatically copied them from the work area to the status area. You can read more about the work area and status area in the *DirX Identity Connectivity Administration Guide*. The next section will show you how to manage these status entries.

#### 1.7.7.3. Cleaning up Workflow Status Entries

DirX Identity provides a cleanup mechanism that automatically manages workflow status entries; you can also delete them by hand.

To remove a monitor workflow status entry that you no longer need, select the entry, click the right mouse button, and then click **Delete**. The DirX Identity Manager deletes it from the monitor list and removes all related files from the status area in the file system.

This information is permanently removed from the database and from the file system. There is no way to recover it.

DirX Identity provides a cleanup mechanism that automatically removes status entries from the monitor list after a certain length of time has passed from the time the workflow was run. You can set up a specific status lifetime in each workflow entry or use the global default in the central configuration object.

The **Status Life Time** field in the global configuration object controls by default the time period that DirX Identity uses to determine which status entries to delete. It is set by default to one month from the time the workflow was run. To change the **Status Life Time** default value, in DirX Identity Manager:

- Click **Expert View**.
- Click **Configuration** in the tree in the middle pane. This action displays a global configuration dialog that shows general parameters for DirX Identity.
- Click the **Status Tracker** tab.
- Look at **Status Life Time**. It is set to one month (720 hours).
- Click **Edit**.
- Enter a two-month value in **Status Life Time** (1440 hours).
- Click **Save** to store the information permanently in your configuration database.

You can use **Expert View** to set up any configuration parameter you want and you can use **DirX Identity Manager** to configure every aspect of DirX Identity operation. Changing configuration parameters can have wide-ranging effects on DirX Identity operations. Do not change configuration parameters before you clearly understand the impact of the change. Read the documentation carefully.

## 1.7.8. Analyzing the Result

In this section, we'll analyze the result of the workflow run by:

- Checking the newly created data entries
- Analyzing the privileges

This section also provides information how automatic assignment works.

### 1.7.8.1. Viewing the Data Entries

Now let's look at the created data entries. In DirX Identity Manager → **Provisioning** → **Users**:

- Open the tree **Users** → **My-Company** → **Product Testing**.
- There should be 17 entries now including the user facet (click the refresh button if they're not visible). Our new workflow added 9 additional entries.
- Click the entry **Bader Hans** and then the tab **General** (if not yet selected).
- Check that the attributes **Name**, **Last Name**, **First Name**, **Title**, **Employee Type** and **Employee Number** are set correctly as well as the **Master** (NEWHR) and **Identifier** (NHH8753) attributes.
- The **Location** tab shows the **Country** attribute.
- The **Organization** tab shows the **Organizational Unit** attribute which represents the link to the related business object. Click the link button to view this object. View the title, the description and the **Department Number** attribute. Close the Window. The user attribute Department Number in the Organization tab is set to the same value. This shows that users can inherit attributes from business objects.
- The **Operational** tab shows that the entry is in the ENABLED state. In our mapping for the workflow we have set the dxrState attribute to NEW. The change to the ENABLED state was performed by the EventBasedUserResolution workflow that was triggered by the meta controller and that ran in background. If you are fast enough you can see the NEW state for some time during the import operation.

Now let's look at the user's privileges.

### 1.7.8.2. Viewing the Permissions

Now we check one of the user's privileges.

- Click the **Assigned Roles** tab.
- All imported users have two roles assigned: **Internal Employee** and **Test Tasks**.
- View the **Assigned Permissions** tab.
- Besides the **Signature Level 1** permission that was assigned by a rule, all other permissions are inherited from roles.
- Click the **Assigned Groups** tab and then sort the **Target System** column.
- The users have group memberships in the **Intranet Portal**, **MVS**, **Signatures** and

**Windows Domain Europe** target systems. Only the memberships of the Intranet Portal and Signatures target systems are in state ENABLED.

- Now click the **Accounts** tab.
- You can see that the users have only accounts in the **Intranet Portal**, **MVS** and **Windows Domain Europe** target systems. The Signatures target system does not need an account. It is a virtual target system where the user is directly a member of the related groups.

In the next section we will analyze where the privileges come from that we did not explicitly assign.

#### 1.7.8.3. How Automatic Assignment Works

We did not assign any privileges during import of the entries. The question is where the two roles came from.

First we analyze the Internal Employee role.

- View the **Assigned Roles** tab again.
- You can see from the **Assigned by** column that the Internal Employee role was assigned by **rule**. So let's check the available rules.
- Open **Policies → Rules**. Under this folder all rules are located as there are provisioning rules, consistency rules and validation rules. In our case we look for provisioning rules.
- Open the **Role based scenario** under **My-Company** and then the **Corporate** folder. Here you can find the **Internal Employees** rule.
- It assigns all users that have the employeeType set to "Internal" the **Internal Employee** role.

You might remember the **Change Notification** section in the **Import Properties** section of the import workflow. The meta controller has sent JMS message notifications that triggered the EventBasedUserResolution workflows. These workflows evaluate the provisioning rules. See Event-based User Resolution Workflow Operation.

Next we check where the Test Tasks role comes from.

- View the **Assigned Roles** tab again.
- You can see from the **Assigned by** column that the Test Tasks role was inherited from a business object (**BO**).
- Open **Provisioning → Business Objects → Companies → My-Company → Product Testing**.
- Click the **References** tab.You will see the **Test Tasks** role that is assigned to this organizational unit.

All users that are linked to this object inherit this role.You might remember that we have set a link in the import workflow to this object.

This section showed that you can assign privileges automatically either via provisioning rules or via business object inheritance.

Another concept is, to assign roles directly from the import workflow (for example if the HR system delivers information that you can map directly with roles).The JMS message notification will then trigger a resolution via the EventBasedUserResolution workflow.

# 1.8. Changing the Workflow Configuration

This exercise shows how to change a workflow's configuration.In this example, we will change the NewHR2Ident workflow to set some additional attributes.

## 1.8.1. Checking User Bader's Attributes

First, we'll use DirX Identity Manager's **Provisioning** → **User** view to look at user Bader's attributes:

- In **Provisioning** → **Users** → **My-Company** → **Product Testing**, open **Bader Hans**.
- On the **General** tab, the **Description** field, and on the **Relationships** tab, the **Manager** field is not set.

## 1.8.2. Changing the Workflow's Parameters

Now we'll use DirX Identity Manager's **Connectivity** view and run the Configure a Workflow wizard to change some of the **NewHR2Ident** workflow's parameters:

- Change to **Connectivity** → **Global View**, select **My-Company** → **NewCompany** in the scenario pane's tree view.
- In the scenario map, right-click the workflow line and select **NewHR2Ident** → **Configure**. The workflow wizard opens.

### 1.8.2.1. Changing the Source Selected Attributes

In the Configure Workflow wizard, click the **Source Selected Attributes** step and add the attributes **EMPL.Comment** and **EMPL.Manager** to the **Selected attributes** list.

### 1.8.2.2. Changing the Target Selected Attributes

In the Configure a Workflow wizard, click the **Target Selected Attributes** step and add the **manager** attribute to the **Selected attributes** list (**description** has already been added).

### 1.8.2.3. Changing the Attribute Mapping

In the Configure a Workflow wizard, click the **Attribute Mapping** step to change the mapping definition:

- In **Input arguments**, click **file.EMPL.EntryDate**, then click to create a new empty mapping line above it. Select **file.EMPL.Comment** in **Input arguments**, **lStringEscape** in **Mapping function** (it escapes some special characters if present) and **ldap.description** in **Output**.
- In **Input arguments**, click **"dxrUser"** then click to create a new empty mapping line.

- In **Mapping function**, select **IDNcreate.**
- In **Input arguments**, add these fields, using to create additional lines in the column:

  "CN"
  file.EMPL.Manager
  "OU"
  file.EMPL.Department
  "O"
  "My-Company"
  "{$base_obj}"

- In **Output**, select **ldap.manager**.
- Click **Finish** to close the workflow wizard.

The wizard now automatically adds the information you specified to the workflow's mapping routine (a Tcl script). You do not need to be able to program in Tcl to perform this task.

### 1.8.3. Running the Reconfigured Workflow

Now we'll run the reconfigured NewHR2Ident workflow and check the status. In DirX Identity Manager → **Connectivity** → **Global View** → **Scenarios** → **My-Company** → **NewCompany** scenario:

- Run the **NewHR2Ident** workflow, and then click the **Structure** tab.
- Click the **metaCP** activity and then click the **Statistics** tab.There should be 9 modified entries with the result **OK**.

### 1.8.4. Re-Checking User Bader

Check **Bader Hans** in **Provisioning** → **Users** → **Product Testing** again to see if the **Description** and **Manager** fields are now set correctly.You can find the manager in the **Relationships** tab.

This example shows that it is easy to re-configure a workflow.You do not need to know any details about the complicated setup of specific agents or even sequences of agents.

## 1.9. Setting up a New Target System

Recall from "Adding a New Identity Source" that My-Company bought another company New-Company with its own HR database (a Microsoft Access database named **New-HR**) and several target systems.My-Company determines that only one target system - New-LDAP, an LDAP-based application - must be integrated.All other target systems (for example, a Windows domain) are to be replaced with the existing My-Company target systems.This exercise illustrates how to set up the New-LDAP target system, including how to:

- Install the connected system

- Add the new target system to the My-Company scenario
- Load the content (the accounts and groups) of the New-LDAP connected system into My-Company's identity store
- Join the loaded accounts to users
- Assign the loaded groups to permissions
- Synchronize the accounts and groups
- Validate the content of the New-LDAP target system against My-Company's identity store after making manual changes in the target system

## 1.9.1. Installing the Connected System

Recall from the quick start preparation that we already created a sample node for connected systems (sample-ts). In this part of the exercise we import an additional connected system into DirX Identity Manager's data view. This connected system simulates a Web-based test application that uses this LDAP directory for access control.

#### 1.9.1.1. Creating the Connected System

Next, we'll create (import) the connected system with DirX Identity Manager:

- Select **Data View** → **Connectivity**.
- Click the **o=sample-ts** node (Base DN: o=sample-ts).
- Log in with the user DN **cn=DomainAdmin,cn=My-Company** (simple bind).
- In the **File** menu, select **Import**.
- Select the file **newLDAPts.ldif** from the path *install_path***data\extension**.
- Select **Add entry only**, then click **OK**.

DirX Identity Manager imports 14 entries into the connected system (sample-ts) tree in the LDAP directory.

Now restart DirX Identity's Java-based Identity Server (IdS-J) to reload the object descriptions.

#### 1.9.1.2. Checking the New Connected System

Open the new tree **ou=new-ldap** (click the refresh button if it's not visible). You'll see an accounts and a groups subtree. The accounts subtree contains 9 accounts, while the groups subtree contains two groups. Four accounts are linked to the Firmware Tests group, and five accounts are linked to the Software Tests group.

## 1.9.2. Adding the Target System

Now we'll use DirX Identity Manager's Target System wizard to add the new target system into the My-Company scenario.

#### 1.9.2.1. Configuring the Target System

We'll use DirX Identity Manager's target system wizard to create and configure the new target system, which we'll call New-LDAP:

- Select the **Provisioning** view and then **Target Systems**.
- Right-click the top-level node and select **New** → **Target system**.
- The new target system wizard opens. Click **Next** to continue.

##### 1.9.2.1.1. Selecting a Target System Template

First, we need to select a template for our new target system:

- Select **LDAP** from the available types because we need to connect an LDAP-type target system. Do not check **Accounts and groups in common subtree** because in our new target system the accounts and groups are in different trees.
- Click **Next**.

##### 1.9.2.1.2. Specifying a Name and an Administrator

In this step, we'll name the target system and identify its local administrator:

- Enter **New-LDAP** in **Name** and **Test application for New-Company** in **Description**.
- It's a good idea to set the local administrator field even if it is only for information. Click the ...button, and then select **Dyson Mark** from **Users** → **My-Company** → **Product Testing**. Mark will handle the target system in DirX Identity as well as manage the "real" LDAP directory for the test application. Click **Next**.

##### 1.9.2.1.3. Specifying a Cluster and a Domain

We need to specify a cluster and domain for the new target system (you can read more about clusters and domains by clicking **Help**):

- Enter **New-Company** in **Cluster** and **New-LDAP** in **Domain**.
- Activate **Enable Realtime Provisioning**.
- Click **Next** twice to skip the **Target System Timing** step.

##### 1.9.2.1.4. Configuring the Account and Group Roots

In this step, we need to specify the locations at which the accounts and groups reside in the New-LDAP connected system:

- Set the **Account Root in CS** to **ou=accounts,ou=new-ldap,o=sample-ts**.
- Set the **Group Root in CS** to **ou=groups,ou=new-ldap,o=sample-ts**.
- Click **Next**.

##### 1.9.2.1.5. Selecting the Connectivity Scenario

In this step, the wizard asks in which scenario the new target system is to be created:

- Select the **Main** scenario.
- Click **Next**.

#### 1.9.2.1.6. Selecting the Associated Connected Directory

This step already presents **LDAP**.

- Leave this selection as it is and click **Next**.
- Click **Next** two times more to skip the **Schema** and **Attribute Configuration** steps. In a real scenario, you would load the schema from the new LDAP directory here to adapt the selected attributes and the mapping. These steps are not necessary for this simple example.

#### 1.9.2.1.7. Specifying Connected Directory Configuration Information

In this step, the first two lines show where the accounts and groups are stored in the identity store (within the target system New-LDAP):

- Enter the host name or IP address of your machine in **IP Address**.
- Enter **cn=domainadmin,cn=my-company** in **User**.
- Enter a password to access the New-LDAP directory in **Password**.
- Click **Next**.

#### 1.9.2.1.8. Selecting the Provisioning Workflow Types

This step permits us to choose the workflow types that will be configured to the New-LDAP directory.

- Select only **Java-based Workflows**. (Deselect the **Synchronization** and **Validation workflows**. They reflect the Tcl-based workflows that we do not need.)
- Click **Finish**.

The target system wizard now creates a synchronization and an initial load/validation workflow in realtime technology. This will take some time.

### 1.9.2.2. Checking the New Target System

Now let's check the **Provisioning** → **Target Systems** tree for the new entry **New-LDAP** and open it. It contains **accounts** and **groups** sub-nodes that are both empty except for the default query filters.

- Select **Connectivity** → **Global View** and check to see if the connected directory **New-LDAP** exists in the My-Company scenario map. Move it to your preferred location.
- Right-click the workflow line between the **Identity Store** and the **New-LDAP** connected directory.

The context menu contains a synchronization, a set password and a validation workflow.

Perform the following steps to activate the synchronization workflow:

- Select **Ident_LDAP_Realtime** → **Configure**.
- Set the **Is Active** flag.
- Click **Finish**.

Perform the same procedure for the validation workflow (**Validate_LDAP_Realtime**).

Now you must reload the IdS-J configuration by right-clicking on a workflow - for example, **Connectivity** → **Expert View** → **Connectivity Configuration Data** → **Workflows** → **My-Company** → **Main** → **Target Realtime** → **LDAP** → **Ident_LDAP_Realtime** and choosing **Load IDS-J Configuration** - or simply by restarting the Java based Server.

## 1.9.3. Loading the Accounts and Groups

Now we'll load the accounts and groups of New-LDAP connected system into DirX Identity target system. First, we'll run the validation workflow for the New-LDAP target system. Then we'll monitor the workflow results and check the imported entries in the target system itself. Finally, we'll look for unassigned accounts, and check the imported groups.

#### 1.9.3.1. Running the Validation Workflow

First we'll use DirX Identity Manager to run the New-LDAP target system's validation workflow:

- Select **Provisioning** → **Target Systems**.
- Right-click the **New-LDAP** target system, then select **Connectivity** → **Workflows** → **Validate_LDAP_Realtime** → **Run Workflow**. The validation workflow runs in initial mode and imports the accounts and groups.

#### 1.9.3.2. Checking the New Entries in the Target System

Next, we'll view the created entries under the New-LDAP target system node:

- In **Provisioning** → **Target Systems** → **New-LDAP**, open the **accounts** and **groups** folders.

  You can see the 9 accounts and the two groups (click the refresh button if they're not visible).
- Click **Bader Hans**. You can see on the **Operational** tab that this account is in the state IMPORTED and in the connected system state ENABLED, which means that the account is active (not disabled) and it is currently controlled from the target system (IMPORTED).
- The **LDAP** tab shows the employee number in **Employee number** and the distinguished name in the connected system in **PrimaryKey**.
- In the **Member of** tab you can see that this account is a member of the **Software Tests** group. **Assignment State** is IMPORTED, too and it is marked as a direct assignment, which means that this membership is controlled by the target system.
- Check the other accounts and their settings.

#### 1.9.3.3. Checking for Unassigned Accounts

The **unassigned** folder in a target system's **accounts** folder offers an easy way to check whether all the target system's accounts are assigned to users.

- In **Provisioning** → **Target Systems** → **New-LDAP** → **accounts**, open the **unassigned** query folder. The unassigned accounts are shown here.

#### 1.9.3.4. Checking the Imported Groups

The last step is to check the imported groups:

- In **Provisioning** → **Target Systems** → **New-LDAP** → **groups**, click **Firmware Tests** and **Software Tests**. You can see on the **Operational** tab that both groups are in state ENABLED and in connected system state ENABLED.
- The **LDAP** tab shows again the distinguished name from the connected system in **PrimaryKey**.
- The **Permissions** tab is still empty because there is no permission that controls the memberships in this group.
- The **Members** tab shows the members of these groups (four in **Firmware Tests**, five in **Software Tests**), all in the state IMPORTED.
- All other tabs are empty and of no meaning at the moment.

Now we have successfully imported New-LDAP's accounts and groups into DirX Identity. At this point, DirX Identity can show the status of all accounts and groups in the target system but it does not control them. We'll change this in the next sections.

### 1.9.4. Joining Accounts to Users

In "Importing Identities", we imported the nine persons from New-Company from the New-HR database. In "Loading Accounts and Groups" we loaded nine accounts from the New-LDAP target system. This exercise shows you how DirX Identity joins these two sets of entries.

You can perform this task also in event-based mode if you use the **EventBasedAccountProcessing** workflow. This is not demonstrated in this tutorial. See Event-based Maintenance Workflow for Accounts.

#### 1.9.4.1. Creating a Consistency Rule for Joining Accounts to Users

We know that both users and accounts have the employeeNumber attribute set, which makes it a perfect join key. (If this is not the case in your real world, you must look for a combination of other attributes to use as join keys.) We need to set up a consistency rule that specifies this join key.

DirX Identity comes with a set of pre-defined rules you can modify to your requirements. We'll use the assocAccount2User consistency rule as a template for our rule. In DirX Identity Manager's **Provisioning** view:

- Select **Policies** → **Rules** → **Default** → **Consistency**.
- Right-click the **Rules** node and select **New** → **Rule Container**. In the pop-up dialog, set the name to **New-LDAP**.
- Right-click **assocAccount2User** and copy it to the **New-LDAP** rule container folder (use the **Copy Object** method).
- Edit **assocAccount2User** in the **New-LDAP** rule container and check **Is active** in the **General** tab.
- In the **Parameter Values** area, click in the **Value** field of the **joinFilter** line, then right-click and select **Edit Filter**. The LDAP filter editor starts. Click the **>** button and use **Delete row** to delete the third and fourth rows. Change the second row to **employeenumber equals $(subject.employeenumber)**. Click **OK**.
- In the **Filter** tab, set **Search base** to **TargetSystems** → **New-LDAP** → **accounts**. Click **Save**.

#### 1.9.4.2. Creating a Consistency Workflow that Uses the New Rule

Next, we'll set up a consistency workflow to use the consistency rule we just created and configured. The My-Company scenario already contains a policy execution workflow we can use for this purpose.

- Select **Provisioning** → **Target Systems**.
- Right-click the top-level object **Target Systems**, then select **Connectivity** → **New Workflow**. The workflow configuration wizard starts.
- Select the **PolicyExecution** workflow from the folder **My-Company/Main/Identity Store**.
- Click **Next** twice to skip the **Define Direction** step.
- In the **General Information** step, set **Name** to **AssociateAccounts** and the **Description** to **Associate accounts to users in New-LDAP**.
- Click **Next** twice to skip the **Policy Agent Parameters** step.
- In the **Rule Search Parameters** step, change **Base Object** to **cn=New-LDAP,cn=Rules,cn=Policies,cn=My-Company** and set **Search Filter** to **(objectClass=dxrConsistencyRule)**.
- Click **Next** several times until you reach the end, and then click **Finish**.

#### 1.9.4.3. Assigning the Consistency Workflow to the New-LDAP Target System

Because the consistency workflow we just created belongs to the New-LDAP target system, it makes sense to remove it from the top-level object and assign it to the New-LDAP target system.

- Click **Provisioning** view **Target Systems**.
- Right-click **Target Systems** in the tree pane and select **Connectivity** → **Workflows** → **AssociateAccounts** → **Remove**.
- Click **Provisioning** view **Target Systems**.
- Right-click on **New-LDAP** below **Target Systems** and select **Connectivity** → **Assign Workflow** and select **AssociateAccounts** from the list.

#### 1.9.4.4. Running the Consistency Workflow

Now we'll run the consistency workflow to execute the consistency rule:

- Click **Provisioning** view **Target Systems**.
- Right-click on **New-LDAP** below **Target Systems** and select **Connectivity** → **Workflows** → **AssociateAccounts** → **Run Workflow**. The workflow runs and ends without error.

#### 1.9.4.5. Checking the Results of the Consistency Workflow

Next, we'll check the results of the workflow run, first with DirX Identity's workflow structure window, and then in the target system itself.

- In the workflow's structure window, click the **PolicyExecution** activity and then the **Trace** tab. Open the trace file.
- It shows **cn=assocAccount2User, ... | consistency | 1**, which indicates that one account cannot be assigned to a user.
- Close the three open windows.
- In the Target Systems tree, click **New-LDAP** → **accounts** → **unassigned** again (click the refresh button if necessary). The folder contains only the entry for Konrad Derksen. Notice that the **User** link is not set for this account; all the other accounts have their **User** links set after the join operation. This is because there is no user Konrad Derksen in the New-HR database and thus in the Identity Store.

#### 1.9.4.6. Resolving the Unassigned Accounts

In the real world, having a set of unassigned users after a join operation is a common occurrence depending on how clean your system environments are. To resolve these unassigned users, you should:

- Try to join them with different join filters.
- For accounts that remain unassigned after this step, determine whether each account is still valid:
- If the account is still valid but there is no user in the Identity Store (for example, because it's an administrative account), you can mark it as 'Managed only in target system'. We recommend that you keep the number of such accounts as small as possible.
- If the account is no longer valid, then it may be an obsolete account, which is definitely a security risk. You should remove it from the system.

In our example, we find that Konrad Derksen left New-Company several months ago, so we'll remove the account from the system:

- Right-click the account for Konrad Derksen and select **Delete**.
- DirX Identity notifies that the account is set to be deleted. Check the state (DELETED) and the deletion date. You can see that the account will be deleted in about 1 month, which is the default setting in the **Domain Configuration** view. Select the top-level object and then the **Timing** tab. The maximum time after which an object is deleted is set to 30 days.

- Check the group **Firmware Tests**. The membership of Konrad Derksen is set to DELETED.

## 1.9.5. Integrating the Groups into the Privilege Structure

Next, we'll integrate the groups into the privilege structure. To allow us to demonstrate two different ways of doing this, we'll assume that everyone in the Software and Firmware Test departments will use the new software test tool that comes from New-Company. Firmware Tests are only performed by Binder, Dyson and Karrer (just as before).

### 1.9.5.1. Assigning the Software Tests Group to the Test Tasks Permission

A simple way to assign the Software Tests group to all Product Testing department members is to assign this group to the Test Tasks permission. In DirX Identity Manager → **Provisioning**:

- Select **Privileges** → **Permissions** → **Corporate Permissions** → **Department Specific**.
- Click **Test Tasks** and then **Edit**.
- In **Assigned Groups**, search for **Name begins with s**. Click the **Software Tests** group of the **New-LDAP** target system and use  to assign it to this permission.
- Click **Save**.

DirX Identity resolves the affected users immediately and assigns new users to this group.

### 1.9.5.2. Checking the Results of the Group-Permission Assignment

Now we'll look at the New-LDAP target system's accounts and the Software Tests group to view the results of the group-permission assignment:

- Click **Target Systems** and then **New-LDAP** → **Accounts**. You can see 15 new accounts (the ones with the number as a suffix) in addition to the 9 that already exist. The existing accounts are now in the state ENABLED and in the connected system state ENABLED. The new accounts are in state ENABLED and in the connected system state NONE, which will switch to ENABLED after a while.
- Click **Groups** → **Software Tests** and check that there are 5 members in the state ENABLED and 19 members in the state ADD, which will switch to ENABLED after a while.

From this information, you can see that assignment of a group to a permission is sufficient to bring the accounts and group memberships under DirX Identity's control.

### 1.9.5.3. Creating a New Permission "Firmware Tests"

For the Firmware Tests group, we'll create a role by hand, including a permission. First we'll create the permission:

- Click the **Privileges** view and then select **Permissions** → **Corporate Permissions** → **Department Specific**.
- Right-click **Department Specific** and select **New** → **Permission**.

- Set **Name** to **Firmware Tests** and enter a description for the permission. Select the **Assigned Groups** tab and assign the **Firmware Tests** group from the **New-LDAP** target system. Click **OK**.

#### 1.9.5.4. Creating a New Role "Firmware Tests"

Now we'll create the Firmware Tests role and assign the Firmware Tests permission to it:

- Click **Provisioning** → **Privileges** and then **Roles** → **Corporate Roles** → **Department Specific**.
- Right-click **Department Specific** and select **New** → **Role**.
- Set **Name** to **Firmware Tests** and enter a description for the role.
- Fill the other fields as appropriate. It's a good idea to specify an owner of the role in the event that you want to control its assignment with a request workflow (which we do not require at the moment).
- Select the **Assigned Permissions** tab and assign the **Firmware Tests** permission. Click **OK**.

#### 1.9.5.5. Assigning the Firmware Tests Role

Now we have created a new role that we can assign to users by hand:

- Click the **Users** view and then select **Users** (top-level node) → **My-Company** → **Product Testing**.
- Assign the new **Firmware Tests** role to the users Binder, Dyson and Karrer.

#### 1.9.5.6. Checking the Group Integration Results

Now we'll check the Firmware Tests group and the user Binder to see what their states are in DirX Identity:

- Click the **Target Systems** view and then **New-LDAP** → **Groups** → **Firmware Tests**.

There are three persons in the state IMPORTED.

There are other ways to integrate the access control of New-LDAP into the My-Company domain in addition to the exercises shown here. For example:

- You can define a role and permission for the Software Tests group, too, and assign the role by hand.
- You can define another attribute - for example, testDetail - for the test users that specifies their tasks in more detail, then define a provisioning rule that assigns the Software Tests role automatically.

### 1.9.6. Synchronizing the Target System

In this step, we synchronize the New-LDAP target system by running the New-LDAP target system's synchronization workflow.

#### 1.9.6.1. Running the Synchronization Workflow

To run the New-LDAP target system synchronization workflow:

- Click the **Target Systems** view and then **New-LDAP**.
- Run the synchronization workflow via **Connectivity → Workflows → Ident_LDAP_Realtime → Run Workflow**.

#### 1.9.6.2. Checking the Results of the Synchronization Workflow

Next, we'll look at New-LDAP target system's accounts and groups to see the result of the synchronization workflow:

- Seven additional accounts have been automatically created. There's a total of 24 accounts: 8 accounts from New-Company and 16 new ones (including one functional user account).

The account for Konrad Derksen is still in the state DELETED. The new account names were created according to the naming rule for the name property that is defined by default for LDAP-type directories. Check **Domain Configuration → My-Company → TargetSystems → LDAP → Object Descriptions → TSAccount.xml**. Scroll down to the line that begins with **<property name="cn"** ... to view the naming rule. As you can see, the rule defines a hierarchy of naming specifications according to the available properties at the user object. The first one that is fulfilled is taken.

The naming rules in this file are default rules. If you want to define your own rules, you should not change this file due to the fact that it is overwritten with each upgrade of DirX Identity. Use the file **Target Systems → New-LDAP → configuration → object descriptions → tsaccount.xml** instead.

- The Software Tests group contains 24 members that are completely controlled by DirX Identity.
- The Firmware Tests group contains 3 members as before, but they are now controlled by DirX Identity. The previous member Konrad Derksen is no longer there.

Now the New-LDAP target system is under full control of DirX Identity.

### 1.9.7. Validating the Target System

It's a good idea to compare the total target system content with the content in DirX Identity from time to time, for example, once a week, to detect deviations. We'll show you how to validate the New-LDAP target system in this section.

#### 1.9.7.1. Changing Some Account and Group Data in the Target System

First, we need to make some changes to the New-LDAP connected system to simulate that

an administrator has added a new person to the Firmware Tests group directly and that he has removed the membership of another person:

- Add a new account into the connected system: use the data view and create an inetOrgPerson in the tree **ou=accounts,ou=new-ldap,o=sample-ts**. Set the name (**cn**) to **Zeller Andreas**, set **sn** to **Zeller** and **givenName** to **Andreas** and set **employeeNumber** to **6543**.
- Use the data view to add this new account to the Firmware Tests group. Add a link to the member attribute.
- Remove Horst Binder from the Firmware Test group but do not delete the account.

#### 1.9.7.2. Running the Validation Workflow

Now we'll run the validation workflow:

- Right-click the **New-LDAP** target system and select **Connectivity → Workflows → Validate_LDAP_Realtime → Run Workflow**.

#### 1.9.7.3. Checking the Results of the Validation Workflow

We'll use the default query folders that DirX Identity provides for accounts and groups to verify the results. In DirX Identity Manager → **Provisioning** → **Target Systems** → **New-LDAP**:

- Click **accounts** → **errors** to view the error query folder for accounts. No errors are reported.
- Click **accounts** → **todos** to view the ToDos query folder for accounts. It lists the Derksen Konrad entry that is already in the DELETED state. No action is required, because a consistency workflow will automatically remove the entry when its deletion date arrives.
- Notice that Andreas Zeller is listed. The ToDo field tells you that this account was created in the target system and not by DirX Identity.
- Click the **unassigned** query folder. It lists **Zeller Andreas** as an unassigned account.
- Click the **errors** query folder for groups. No errors are reported.
- Click the **todo membership** query folder for groups. It lists the **Firmware Tests** group. Click it.
- Click the **Members** tab of the group. You can see the IMPORTED state for the membership of **Zeller Andreas**. Note that the membership for **Binder Horst** is in the ADD state, which means that DirX Identity will create this membership the next time the synchronization workflow runs.
- Let's delete the account for **Zeller Andreas**. Right-click it and select **Delete**.

#### 1.9.7.4. Synchronizing the Changes in the Target System

Run the synchronization workflow again and check the results:

- The account for Zeller Andreas in the connected system is deleted.(This behavior might be different for other types of connected systems.Here, the LDAP connected system cannot handle disabled accounts.)

- In DirX Identity's target system the account is still available but in the DELETED state.The ToDo folder shows the accounts in the DELETED state.They will be removed in about a month if they are not reused by then.
- The group membership in the connected system for Zeller Andreas no longer exists.The membership for Binder Horst has been re-created.
- In DirX Identity's target system the group membership for Zeller Andreas is removed, too.All other memberships are in the ENABLED state.There are no ToDos visible in the group area.

This exercise shows that query folders provide an easy way to view and maintain connected systems from DirX Identity.You can add your own query folders as necessary.

The accounts created in the connected system can be automatically deleted with the help of validation rules.See DirX Identity Provisioning Administration Guide and DirX Identity Customization Guide.

# 1.10. Using Password Management

In this section, we show you how to set up and use a password management solution.The section demonstrates how to:

- Prepare the environment for password management
- Change passwords
- View the effects of password management

## 1.10.1. Preparing for Password Management

When we created the New-LDAP target system, we set up a password synchronization workflow.Before we can demonstrate this feature, we need to prepare our environment for password management operations.Recall from section "Setting up a New Target System" we already set up the target system and the connected system when preparing the quick start.Now we use these systems for our password management exercises.To prepare password management we now perform the following steps:

- Adding the New-LDAP Resource Family to the DirX Identity Java server Configuration
- Stopping, then starting the Java-based DirX Identity Server
- Checking the Java-based identity server state

### 1.10.1.1. Adding the New-LDAP Resource Family

First we need to add a new resource family of the New-LDAP target system to the DirX Identity Connectivity configuration so that we can run password workflows for the New-LDAP target system on our machine. To add the New-LDAP resource family with DirX Identity Manager:

- Log in to DirX Identity Manager and select the **Connectivity** view.

- Click **Expert View**, and then open **Connectivity Configuration Data** → **Configuration** → **Resource Families**. You will see the folder **Default** that contains all the default resource families provided with DirX Identity. (You can read more about resource families in the *DirX Identity Connectivity Administration Guide*.)
- Create a new folder named **My-Company** to hold the resource families to be created for the My-Company domain.
- Create a new resource family in the **My-Company** folder. Enter **New-LDAP** in **Name** and provide a description of the resource family.
- Open **Connectivity Configuration Data** → **Configuration** → **DirX Identity Servers** → **Java Servers** → **My-Company**.
- Click the **My-Company-S1-*machine* server with your machine name and click *Edit.**
- Select the **Resource Families** tab and select **New-LDAP** from the **Available** resources. Add this to the **Selected** resource families by clicking the button. By default, a number of two threads is added to the Java server for the selected resource family. You may change this number in the column **Thread number**. Click **Save**.

#### 1.10.1.2. Restarting the Java-based Identity Server

DirX Identity's Java-based Identity Server (IdS-J), which supports the password management solution, will now create by default two threads that handle this new resource family when you stop, then re-start it. Use the Windows Services administration tool to stop, and then start the IdS-J server.

#### 1.10.1.3. Checking the Java-based Identity Server State

Next, we'll check the state of the IdS-J server we just re-started:

- Start a Web browser and enter this URL for the DirX Identity Web Admin: **http://*your host*:40000/admin**
- Enter **admin** for the account and **wE3!dirx** for the password.
- The initial screen shows that the server is running. Click **Java Server**. You will see the number of processors, the memory used and other parameters.
- Click **Expand all**. Click **Worker containers**. Now you can see all worker threads that run within this server.
- Check that two threads are waiting for New-LDAP tasks.
- Close the browser window to exit the program.

Now the password management solution is ready for use. You can read more about the Ids-J server and worker containers in the *DirX Identity Connectivity Administration Guide*.

### 1.10.2. Changing Passwords

Now we'll change a user password and synchronize it to the Intranet Portal, Extranet Portal and New-LDAP connected systems. First, we'll use DirX Identity Manager to make sure password management is enabled, and then we'll use the DirX Identity Web Center to make the password change.

#### 1.10.2.1. Ensuring that Password Management is Enabled

Before we try to change a password, we should first make sure that the **New-LDAP**, **Intranet Portal**, and **Extranet Portal** target systems are configured to permit password management. This feature is controlled by the "Disable Password Synchronization" flag for the target system. We'll use DirX Identity Manager's Provisioning view to check this flag:

- In DirX Identity Manager → **Provisioning** → **Target Systems**, open each target system, and then click the **Advanced** tab. You will note that **Disable Password Sync** is set for all target systems except the **New-LDAP**, **Intranet Portal**, **Extranet Portal**, and **DirXmetaRole** target systems.
- In the **Connectivity** view, verify that all password sync workflows are active. You must activate the last workflow **SetPassword in LDAP**:
- **Connectivity Configuration Data** → **Workflows** → **My-Company** → **Main** → **Password Synchronization** → **UserPasswordEventManager**
- **Connectivity Configuration Data** → **Workflows** → **My-Company** → **Main** → **Target Realtime** → **Extranet Portal** → **SetPassword in Extranet**
- **Connectivity Configuration Data** → **Workflows** → **My-Company** → **Main** → **Target Realtime** → **Intranet Portal** → **SetPassword in Intranet**
- **Connectivity Configuration Data** → **Workflows** → **My-Company** → **Main** → **Target Realtime** → **LDAP** → **SetPassword in LDAP**

#### 1.10.2.2. Changing User Duplan's Password

Now we'll use the DirX Identity Web Center to change a user password:

- Start the Web browser and enter the URL **http://*your_host*:8080/webCenter-My-Company**
- Enter **Duplan Frederic** as the name and enter the password.
- Select **Self Service** → **Change password**.
- Enter the old password and a new password for example **1!dirX** twice. Click **Save**.

Web Center will change to the **User - Summary** page. It has sent a message to DirX Identity's password event manager for changing the password.

### 1.10.3. Viewing the Effects of Password Management

When a user changes his password, DirX Identity determines where this user has accounts that need to be synchronized. Check the **Accounts** tab of user Duplan in **Provisioning** → **Users** → **My-Company** → **Product Development**. You'll see that he only has an account in the **Intranet Portal** target system that needs to be synchronized.

In this example we use the simple case that only one target system password is to be synchronized to keep it simple. If you use another user that has accounts in more than one target system (that are enabled for password synchronization), you can generate update of passwords to several connected systems.

To track the password change, you can use:

- The DirX Identity Manager **Connectivity** → **Monitor View** to view the generated status entries.
- DirX Identity Web Admin to view statistics and to check various components of the IdS-J server.

The next sections describe how to use both of these tools to view the password change you made to user Duplan.

#### 1.10.3.1. Viewing the Effects with the Monitor View

First we'll show how to use DirX Identity Manager's **Connectivity** → **Monitor View** to view the effects of the password change. We'll check the results from the points of view of the password event manager, the Internet Portal target system, and the user Duplan.

##### 1.10.3.1.1. Checking the Results at the Password Event Manager

First we'll use the Monitor View to look at the status entry for the password change at the password event manager:

- Log in to DirX Identity Manager and select **Connectivity** → **Monitor View**.
- Open **Status Data** → **My-Company** → **Main** → **Password Synchronization**. For each workflow you can see a folder. Each one contains one workflow status entry.
- Click **UserPasswordEventManager** ***your timestamp***.
- The status of this entry is **closed.completed.ok** which means that this part of the change worked well.

##### 1.10.3.1.2. Checking the Result at the Intranet Portal Target System

Next, we'll look at the status entry for the password change at the Intranet Portal target system:

- Open **Status Data** → **My-Company** → **Main** → **Target Realtime** → **Intranet Portal** → **SetPassword in Intranet**. For each workflow you can see a folder. Each one contains one workflow status entry.
- Click **SetPassword in Intranet** ***your timestamp***.
- The status of this entry is **closed.completed.ok** which means that this part of the change worked well.

#### 1.10.3.2. Viewing the Effects with Web Admin

You can also use the Web Admin to view the effects of a password change:

- Start a Web browser and enter the URL for DirX Identity Web Admin: **http://** *your host* **:40000/admin**
- Enter **admin** for the account and **wE3!dirx** as password.

In the next sections, we'll show you how to view and interpret statistics from the password change listener, event listener, and dead letter queue adaptors, and how to view server and audit log files. You can read more about Web Admin adaptors and log files in the *DirX Identity Connectivity Administration Guide*.

#### 1.10.3.2.1. Checking the Password Change Listener Adaptor

To check the adaptor for the password change listener:

- Click **Expand all**.
- Click **Java Server** → **Adaptors** → **PasswordChangeListener**. The number of sent requests and received responses is 1. This means that this listener could read the password request from the messaging server and put it into the internal queue.

#### 1.10.3.2.2. Checking the SetAccountPasswordListener Adaptor

To check the SetAccountPasswordListener adaptor:

- Click **Java Server** → **Adaptors** → **SetAccountPasswordListener**. The number of sent requests and received responses is 1. This means that this listener could read the password request from the internal queue. It tried to start the corresponding workflows.

#### 1.10.3.2.3. Checking the Dead Letter Queue Adaptor

To check the dead letter queue adaptor:

- Click **Java Server** → **Dead letter queue**. The number of stored requests should be zero. This means that all requests were processed correctly. If there are stored requests, you can view them, run them again (once you've determined the reason why the event failed) or clear them (if it is not possible to solve the problem)

#### 1.10.3.2.4. Viewing the Server Log Files

You can also use Web Admin to look at the server log files when something in password management does not work correctly. To view these files:

- Click **Java Server** → **Logging** → **View log files**

# 2. Follow-on Tutorials

The follow-on tutorials in this chapter describe how to:

- Re-use the rules and concepts of the sample domain in your customer environment
- Integrate a customer-specific agent into DirX Identity
- Maintain the privilege structure
- Change a workflow's structure
- Apply segregation of duties (SoD) checking
- Enable re-approval for a privilege assignment
- Certify a role
- Certify a user
- Apply attribute modification approval
- Use scheduled privilege assignment
- Create a nested workflow
- Use manual provisioning
- Work with internal tickets
- Work with source tickets
- Create and handle personas
- Create and handle functional users
- Use risk management

## 2.1. Reusing the Sample Domain

It's a good idea to reuse parts of the sample domain scenario to build a customer domain, especially the policies and request workflows in the Policies and Workflows collections.

To re-use components of the sample domain:

- Configure your customer domain.
- Export the objects to be reused from the sample domain.
- Import the objects to your customer domain.
- Adapt the objects to your customer domain.

You can also add or remove objects from the default collection or create your own collections to meet your requirements.

### 2.1.1. Configuring the Customer Domain

We assume that you have already installed DirX Identity and that you have not created your own customer domain (if you have already created a customer domain, you can use

this domain in these exercises instead of creating another one). Now we'll create the new domain a customer domain - we'll name it **YourDomain** - and check that it exists.

#### 2.1.1.1. Creating the New Domain

To create the YourDomain domain:

- Start the Configuration Wizard.
- In Welcome step, select the Credentials file you have used before.
- In Configuration Options step, select all options.
- In Domain Configuration step, select **Configure a customer domain**. Enter **YourDomain** to the **Name** field. Enter password for the **DomainAdmin** user, for example, dirx"(it will be saved in the credentials file). All the other fields should be prefilled automatically.
- In Java-based Server step, select server tagged ***(NEW)*** from the **Server to configure** list. The display name should contain **YourDomain**.
- Finish the Configuration Wizard.

The wizard creates your customer domain with the name **YourDomain**, creates a server profile for the new domain for Identity Manager, creates Java server for the new domain and creates instances of web applications and services for the new domain.

#### 2.1.1.2. Checking the New Domain

Now let's check the domain we just created:

- Start DirX Identity Manager and choose Provisioning. In the Login dialog, select **cn=YourDomain** in **Server** and **cn=DomainAdmin,cn=YourDomain** in **User DN** (the Login dialog should display a server profile that contains these parameters). If there is no such server profile, create one. To do so:
  1. Click **...** button at the end of Server row.
  2. *Manage Server Profiles* dialog appears, press **New...** button
  3. *Server* dialog appears, in its *General* tab:
     a. In *General* section, set ***Name*** field value to **YourDomain**. You can pick the ***Color*** to visually distinguish this profile from others (see Identity Manager's Title Panel).
     b. In *Host Info* section, set ***Host***, ***Port*** and ***SSL*** *(Use secure connection (LDAP version 3 only) checkbox)* based on configuration of your Provisioning Database (if you don't remember it, you can find it in Configuration Properties file, relevant properties are in DirX Directory Server for Provisioning step) and set ***Base DN*** to **cn=YourDomain**.
  4. In *Authentication* section, set ***Default User DN*** to **cn=DomainAdmin,cn=YourDomain**
  5. Press OK, then press Close, and back to Login dialog. You should see the new profile in the list now.
- Login to **YourDomain** with the created or existing server profile. Use the password you have defined during Configuration (see Creating the New Domain).

## 2.1.2. Exporting Objects from the Sample Domain

In this step, we export the objects we want to use in our new domain from the My-Company sample domain. To do this, we use the collection feature of DirX Identity. A **collection** defines a set of objects and subtrees that can be exported to an LDIF file for subsequent transfer to another domain. We'll export the Policies and Workflows default collection in the My-Company domain.

#### 2.1.2.1. Enabling Full-Text LDIF Output

Before we export the default Policies and Workflows collection, we need to change a setting in DirX Identity's configuration file to permit DirX Identity to generate LDIF output in full-text format. Although this format is not standards-compatible, it is very useful when you need to change parameters in all parts of an object, as we will do in the "Importing" step. If we do not change this setting, the output of text is in base64 format and it is impossible to change the text content. To change the setting in the configuration file:

- Exit DirX Identity Manager.
- Open the file **dxi.cfg** in the *install_path* **/gui/bin** directory.
- Set the variable **collection.base64** to **false** if it's not already set and then save the file.

#### 2.1.2.2. Copying the Default Policies and Workflows Collection

To copy the default collection from the My-Company sample domain:

- Start the DirX Identity Manager. In the Log in dialog, select the My-Company server profile (**Server** is **cn=My-Company**, **User DN** is **cn=DomainAdmin,cn=My-Company**).
- Select the **Domain Configuration** view and open **My-Company** → **Collections** → **Default** → **Policies and Workflows**.
- Copy the complete collection (**Copy Object**) and then check the content. If you need additional objects or subtrees, change the copied collection.

#### 2.1.2.3. Exporting the Default Collection

To export the collection you just copied:

- Right-click the copied collection and select **Export Collection**.

Check that the exported file exists (see the **Path** field in the **General** tab of the collection).

## 2.1.3. Importing Objects to the Customer Domain

In this step, we'll modify the exported collection and import it into our customer domain.

#### 2.1.3.1. Editing the Exported Collection File

First, we'll edit the collection we just exported:

- Open the exported collection file with any text editor.

- Change all occurrences of **My-Company** to **YourDomain**. (We can perform this step only because we produced the LDIF file in full-text format rather than base64.)
- Save the file.

#### 2.1.3.2. Importing the Modified Collection File

To import the collection file:

- Click the top node in any view of DirX Identity and select **Logout**.
- Log in to **YourDomain**.
- Create a new rule container named **YourCompany** within **Provisioning** → **Policies** → **Rules** by right-clicking on **Rules** and choosing **New** → **Rule Container**.
- Switch to **Data View** → **Provisioning**.
- Select **File** → **Import**.
- Select the previously modified file, click **Modify entry if it exists, add entry otherwise** and **Replace attribute values** and then click **OK**.

The file is imported into YourDomain. If errors occur, correct them in the text file and import the file again.

### 2.1.4. Adapting the Imported Objects

The collection you've imported from the My-Company sample domain consists of Access Policies, Request and Approval Workflows (including Jobs) and Provisioning Rules. Before you can use these imported objects, you need to adapt them to the structures and requirements of YourDomain. The examples given here describe what you might need to change. Select objects from the collection that most closely resemble the ones you need. If you need more objects from the sample domain, include them into a collection and follow the steps described in this section again.

#### 2.1.4.1. Adapting Access Policies

To adapt the imported access policies:

- Set all imported access policies to inactive.
- We use the **Manager for user** policy for our example.
- Check whether the name of the access policy is to be changed. If yes, use the **Rename** command.
- Adapt the **Subject** and **Resource** definitions to the structures in **YourDomain**, for example the search base of the filters.
- Activate the modified access policy, store it and test the result.

#### 2.1.4.2. Adapting Request and Approval Workflows

To adapt the imported request and approval workflows:

- First set all imported workflows to inactive.
- We use the **Manager** workflow for our example.
- Check whether the name of the workflow is to be changed. If yes, use the **Rename** command.
- Open the workflow and click **GM approval**.
- Change the approver list to existing persons in **YourDomain**.
- Activate the workflow, save the object and test the result.

#### 2.1.4.3. Adapting Provisioning Rules

To adapt the imported provisioning rules:

- First set all imported rules to inactive.
- We use the **Internal Employees** provisioning rule for our example.
- Check whether the name of the rule is to be changed.If yes, use the **Rename** command.
- Adapt the filter in the Filter tab to your requirements.
- Set links to the privileges this rule is to provide during the policy execution process.
- Set the active flag and store the rule.
- Use the privilege resolution workflow to check whether the new rule works correctly.

## 2.2. Integrating a Customer-Specific Agent

In DirX Identity, an agent is a stand-alone executable that runs within a scheduled provisioning workflow and supports the interfaces to a specific target system that enables data exchange between that system and the identity store.The DirX Identity agent integration framework allows you (the customer) to integrate customer-specific connectivity scenarios - batch files and/or executables and the directories or databases on which they operate - into DirX Identity's Connectivity Configuration as agents and connected directories so that you can use DirX Identity to control and manage them.

When your existing connectivity scenario corresponds to an agent that already exists in DirX Identity, and that agent allows connectivity to a target system that corresponds your existing database and supplies the necessary workflows, you can use the procedures described in "Getting Started" to integrate your connectivity scenario based on existing templates with the help of the configuration wizards in the DirX Identity Manager's Global View.You use the agent integration framework when there are no templates available in DirX Identity that correspond to your connectivity scenario.

The topics presented in this section describe how to use the agent integration framework to:

- Integrate a completely new connected directory and the associated customer-specific agent that imports and exports data to and from this directory into DirX Identity

- Add new properties and wizards that support the connected directory and customer-specific agent to DirX Identity Manager

## 2.2.1. Understanding the Agent Integration Framework

DirX Identity's agent integration framework allows for different levels of integration, as shown in the following figure.

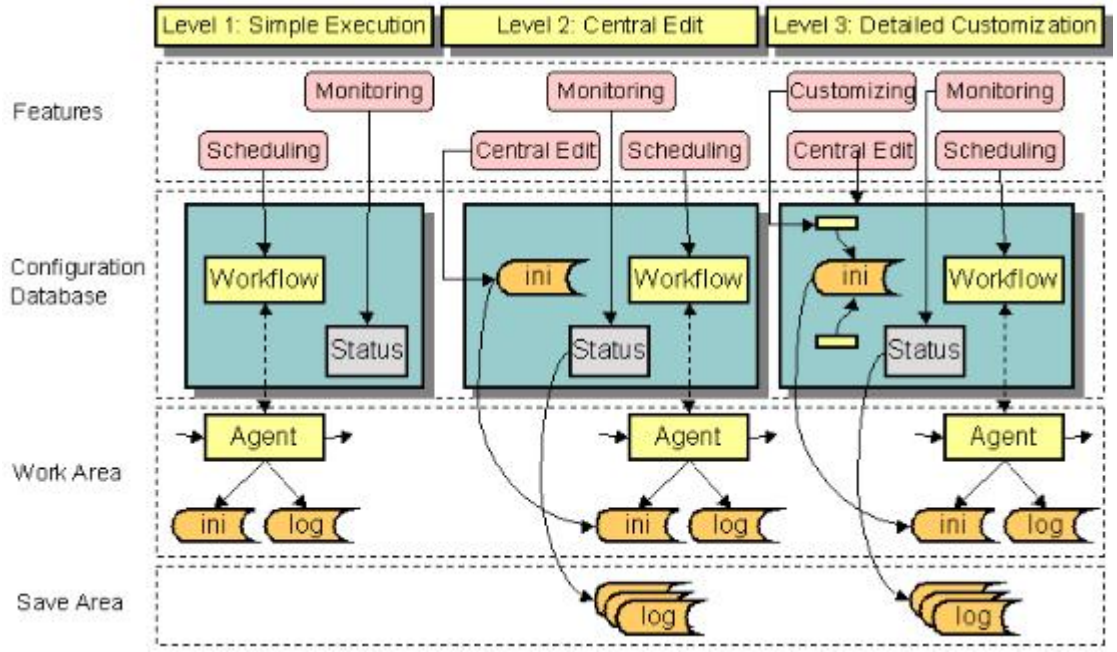



*Figure 5. Levels of Integration*

The next sections summarize the tasks involved in each integration level and the time it takes to complete the integration.

All of these integration tasks assume that your existing connectivity scenario is operating properly and works correctly prior to its integration into DirX Identity. The level of effort that is required to write special scripts for the specific customer connectivity environment is not included in the time estimates provided here.

### 2.2.1.1. Understanding Simple Execution Integration

At the first level (Simple Execution), you integrate your existing connectivity scenario - the batch file or executable and associated database or directory on which it operates - so that the C++-based DirX Identity Server (IdS-C) can schedule it and control it. To integrate to this level, you use DirX Identity Manager's connectivity global and Expert Views to define workflows and jobs for your existing connectivity scenario in the DirX Identity Connectivity configuration. The jobs have no input and output channels, only fixed command lines (which you have copied from the batch scripts of your existing connectivity environment) that start the synchronizations. Integrating to this level requires only a few hours of work.

Once you have completed this level of integration, IdS-C is able to start the connectivity scenario's workflows and their related activities, control their execution (check the exit

codes) and write status entries to the DirX Identity server's status area in the Connectivity configuration. You can schedule the execution of your connectivity scenario's workflows with DirX Identity and have the C++-based DirX Identity server manage their operations. You can also view the generated status entries from DirX Identity Manager's connectivity Monitor View, but you cannot view any related files because DirX Identity has no knowledge of them at this integration level.

#### 2.2.1.2. Understanding Central Edit Integration

At the next level (Central Edit), you use the DirX Identity Manager's connectivity Expert View to integrate your existing connectivity scenario so that all of your agent configuration files, import and export data files, and report and trace files are defined in the DirX Identity Connectivity configuration.

To integrate to this level, you define the name of your input, output, report, and trace files and the location at which to find them before or after a run. You can also specify if they should be deleted after the copy operation. You can also copy the content of your existing agent configuration files into the Connectivity configuration. If you're an experienced DirX Identity user, you'll need about one day to integrate to this level.

When you have completed this level of integration, you have a central location (access to the Connectivity configuration via the DirX Identity Manager) from which to edit these files to manage your connectivity scenario. In addition, the C++-based DirX Identity Server can copy generated files to the status area so that they are not overwritten on subsequent runs of your workflows. You can now view these files from DirX Identity Manager's connectivity Monitor View from any point worldwide.

#### 2.2.1.3. Understanding Detailed Customization

At the highest level (Detailed Customization), you integrate your existing connectivity scenario so that it is tightly integrated into DirX Identity. To integrate to this level, you use the DirX Identity Manager's connectivity Expert View to set up XML descriptions for the objects that you want to extend. You can set references based on these definitions and you can write wizards to make it easy to handle your synchronizations. Integrating to this level can take several days or weeks depending on the features you require.

Once you have completed these steps, you can access the properties of existing Connectivity configuration objects or add properties to your own types of agents and connected directories to the Connectivity configuration (your XML descriptions have defined these types). You can then edit these properties from DirX Identity Manager, and the changes are automatically transferred to the configuration files that control your agents at runtime. You can even define your own wizards to provide for easy configuration. This level allows you to control consistency in a complex workflow environment. Scheduling and monitoring features are also available at this level.

### 2.2.2. Integrating a Connectivity Scenario

This section provides a tutorial that explains how to integrate a connectivity scenario that consists of a new agent and a new connected directory into DirX Identity at each of the integration levels described in "Understanding the Agent Integration Framework". The

topics discussed here include:

- A description of the sample connectivity scenario (its elements and its environment) on which the tutorial is based
- A summary of the tasks required to integrate the sample connectivity scenario into DirX Identity for each integration level
- The level 1 integration of the sample connectivity scenario (simple execution)
- The level 2 integration of the sample connectivity scenario (central edit)
- The level 3 integration of the sample connectivity scenario (detailed customization)

This example does not demonstrate how to integrate an agent into a target system synchronization or validation workflow. It simply shows how to generate a two-step workflow. Synchronization and validation workflows are a combination of such two-step workflows.

#### 2.2.2.1. Understanding the Sample Connectivity Scenario

Now we'll demonstrate how to use the agent integration framework to integrate a sample connectivity scenario - a non-LDAP database and the save and restore tool that manages it - into DirX Identity as a new connected directory and agent and create a workflow from the DirX Identity store to the new connected directory. First, we'll list the pre-requisites for continuing with this exercise, then we'll describe the elements of the connectivity scenario - the non-LDAP database connected directory and agent.

##### 2.2.2.1.1. Prerequisites for Integrating the Sample Connectivity Scenario

Before you can follow this exercise, you must ensure that:

- An identity store has been set up on the main server.
- A DirX Identity C++-based Server installation has been performed on the main server.
- The necessary DirX Identity agents have been installed on the main server.
- A connectivity scenario exists that contains all of the connected directories and workflows provided with DirX Identity.
- The entire scenario runs on one machine.

##### 2.2.2.1.2. Understanding the Sample Connected Directory

The database (we'll name it MyDatabase) needs basic information about a person from the DirX Identity store (**Identity Store**). To keep it simple, we'll model our database as a directory with two constant file names (**import.ldif** and **export.ldif**) in the directory **C:\MyDatabase**. The attributes we want to synchronize to MyDatabase are FirstName, Lastname, PersonnelID, Location, and TelephoneNumber.

##### 2.2.2.1.3. Understanding the Sample Agent

The agent (we'll name it **MyAgent**) that operates on **MyDatabase** uses LDIF file format as its interface: it reads files in LDIF format to import data into **MyDatabase** and produces files

during an export from **MyDatabase** in the same format. **MyAgent** is controlled by command line parameters and a configuration file in "ini-file" (*.**ini**) format (in this section we'll refer to **MyAgent**'s file as its "INI file"). Again, to keep it simple, we'll design our agent to be a batch script that takes a file and copies it to the **MyDatabase** directory. The agent's INI file **control.ini** has the following content:

```
; INI File for MyAgent +
; Generated by Joe Smith at 21.02.2005 12:34:43 +
++[++EntryHandling++]++ +
CreateEntries=true +
DeleteEntries=true +
ModifyEntries=true
```

This INI file does not have any influence on the real behavior of this agent. It simply shows how it could work.

The agent's command line has the following syntax:

**`myagent -i [-t]`** *`ini-file infile`*
for import to **MyDatabase**

The **-t** parameter is optional and creates a trace file if present.

#### 2.2.2.2. Planning the Integration Tasks

Now let's begin to set up the sample connectivity scenario in the DirX Identity Connectivity configuration. We won't perform the entire task in one big step. Instead we'll go through several phases, test, and then go on to the next phase. This process reduces complexity and illustrates each integration level that DirX Identity provides. First, we'll identify and order the tasks to be performed in each phase before we start to enter any data into the Connectivity configuration.

The guidelines given here do not represent the only way to perform the sequence of tasks, but they are straightforward.

##### 2.2.2.2.1. Planning for Simple Execution (Level 1)

To achieve this level of integration, we set up the new agent, the connected directory and a sample import workflow with a job that calls the new agent in the Connectivity configuration. The sample workflow imports from the identity store to the **MyDatabase** connected directory in full mode and contains two steps:

1. An extraction of data from the identity store (we'll use the DirX Identity meta controller **metacp** to perform this task)
2. An import to the new connected directory (we'll use the new agent **MyAgent** for this task).

At this level of integration, we assume that DirX Identity only knows where to start the sample workflow and what the new agent's fixed command line looks like. DirX Identity has no knowledge of the new agent's configuration (the agent's INI file), input and output handling, or tracing. We also assume that the exchange of data between the meta controller and the new agent is carried out through a file (in LDIF format) at a fixed location and that all databases and processes run on one machine (the primary server).

Once we achieve this level of integration, we can:

- Schedule and run the connectivity scenario's sample workflow and its related activities
- View the generated workflow and activity status entries with DirX Identity Manager

At this level, we cannot use DirX Identity Manager to view or trace files.

#### 2.2.2.2.2. Planning for Central Edit

To achieve this level of integration, we:

- Add information about the new agent's INI file to the Connectivity configuration
- Create the input and output channels for the job
- Add information about the new agent's trace file to the Connectivity configuration

In our example, we remove the absolute path for the intermediate file and use relative paths for the entire configuration. This setup facilitates the reconfiguration of entire connectivity scenarios (for example, moving the scenario from one server to another).

Once we complete this level of integration, we can:

- Edit the new agent's INI file as a whole from a central location without having to know where it's located.
- Request that all files generated by the sample workflow be saved to the status area after a run.
- View the generated files or a workflow run.

#### 2.2.2.2.3. Planning for Detailed Customization

To achieve this level of integration, we distribute the configuration parameters in the agent's INI file to the relevant configuration objects so that we can easily edit them with DirX Identity Manager. The INI file now contains references to these distributed parameters (which are now the properties of configuration objects), which are resolved at runtime by the C++-based DirX Identity Server before it starts the agent.

Once we complete this level of integration, we can:

- Monitor the synchronization runs from the DirX Identity Manager and view the generated status files
- Configure the synchronization from the DirX Identity Manager at the object and property level

This level of integration can vary in depth. It is up to you to define how many attributes will be accessible through DirX Identity Manager's graphical interface.

#### 2.2.2.2.4. Entering Connectivity Configuration Data

You primarily use DirX Identity Manager's Expert View to enter Connectivity configuration data, and you should enter it in structured steps. These steps can be separated into:

- Design tasks (planning and designing the new synchronization)
- Setup tasks (for example, installing the new agent)
- Configuration tasks (setting up or modifying objects in the Connectivity configuration with the help of DirX Identity Manager)
- Testing tasks (starting your new workflow and checking to see if it works).

do not forget to put as much descriptive information as possible into the configuration objects, because documentation is important in a complex environment. Use the description field in each object for this purpose.

### 2.2.2.3. Integrating to Simple Execution (Level 1)

This part of the tutorial explains how to integrate the sample connectivity scenario to the "simple execution" level. In this exercise, we'll perform the following tasks:

- Preparation: set up the central configuration object to configure the new connected directory and agent type.
- Copy the workflow Ident2ODBC as the basis for our workflow and adapt it.
- Set up the new agent MyAgent.
- Set up the connected directory MyDatabase.
- Set up the connected directory channels.
- Set up the new job MyJob.
- Perform the rest of the configuration tasks.
- Test: Check the new workflow including start and monitoring.

#### 2.2.2.3.1. Setting up the Central Configuration Object

This preparation step is only necessary when you need to introduce new connected directories or agent types to DirX Identity and if you plan to go on to higher integration levels later on. If this is not the case, you can skip this step.

First, we need to set up a new connected directory type for our connected directory using the DirX Identity Manager Expert View:

- In **Connectivity** → **Expert View**, open the central configuration object **Configuration**.
- Right-click **Connected Directory Types**, and then select **New Connected Directory Type**.

- A dialog asks for the name of the new object. Enter **MyDirType**.

Setting the **Type** field is only necessary for types that are based on the meta controller (in this case we define a connected directory that is used exclusively by our own agent).

- Set **Description** to **My private directory type**.
- Click **OK** to close the dialog.

A new leaf is inserted into the tree. You have now created a new generic connected directory type with no special features. We do no further customization at this point.

Next, we perform the same procedure for the new agent type:

- Right-click **Agent Types** and perform the same procedure.
- Create a new agent type **MyAgentType** and set **Description** to **My private agent type**.
- Click **OK** to close the dialog.

The tree below **Agent Types** is extended with a new leaf. You now have created a new generic agent type with no special features. We do no further customization at this point.

#### 2.2.2.3.2. Setting up the Workflow MyWorkflow

Although you can create a new workflow from scratch, it's much easier to copy an existing workflow and modify it afterwards. We use an Ident2ODBC workflow as a template.

Before we copy the workflow, we should first create a new scenario to work with:

- Select **Connectivity** → **Global View**.
- Click the top-level node and select **New** → **Scenario**. Enter **MyScenario** into **Name**.
- To copy our workflow, we need an Identity Store and an ODBC instance. Create two new connected directory icons and create **MyIdentityStore** from Identity Store of the Default scenario (use the **Configure** method) and assign HR-ODBC from the Default scenario (use the **Assign** method).

The **Assign** operation does not create a new object. This is okay in this example because we only need this object for the workflow copy operation (if we copy this object, we have to delete it afterwards).

Now we can work in our private environment.

Because we want to configure a workflow with two activities (the first extracts data from the identity store, the second transfers this data to our database MyDatabase) we choose a two-step workflow that works in the same direction. In this case the **Ident2ODBC** workflow is a good selection.

- Create a workflow line between the **HR-ODBC** and **MyIdentityStore** icons.
- Click the workflow line and then select **New**.

- The workflow wizard opens. Click **Ident2ODBC** from **Default/Source Scheduled/ODBC** and then click **Next**.
- In the General Information step, change **Name** to **MyWorkflow**.
- Step to Export Properties and change **Search Filter** to **objectClass=dxrUser** without Persona, Facets and Functional User because we want to export all user entries under the base node **cn=Users,cn=My-Company**. Use the filter editor (click the button to the right of the field) to remove the upper three lines.
- Step to the end of the wizard and click **Finish**.

The wizard creates a copy of the original workflow and duplicates all dependent objects. Use the structure view to view the copied object.

- Click the line between the **HR-ODBC** and **MyIdentityStore** icons and select **MyWorkflow** → **Show Structure**.
- The structure view window opens.

Here you see all objects that belong to this workflow (the ones marked with a (1) in the figure "Level 1 integration"):

- Workflow **MyWorkflow**
- First activity **Ident2ODBC_metaCP** (to be renamed to **MyMetaCP**)
- First job **Ident2ODBC_metaCP** (to be renamed to **MyMetaCP**) with all scripts and the mapping information
- Intermediate connected directory **Data** with its attribute configuration information (no renaming necessary)
- Export channel from **Identity Store** to the first job **Ident2ODBC** with the selected attributes information
- Import channel **OutData** to the intermediate connected directory **Data** with the selected attributes information
- Export channel **InData** from **Data** to the next job with the selected attributes information
- Second activity **Ident2ODBC_ODBCImport** (to be renamed to **MyAgentActivity**).

Seven large objects together with a lot of detailed information have been copied in seconds. To build these structures by hand would result in hours of work. Of course we will adapt the naming and other things to fit the needs of our new workflow. We also need to substitute the second job with one that calls our new agent (MyJob) and create a new connected directory MyDatabase.

First, let's use the workflow structure view to modify the created workflow:

- Click **Edit**.

It is a good idea to click all table cells at the very beginning because otherwise you may perform one of the button operations immediately (you can undo this mistake by clicking **Reset** and then **Edit** again).

- Click the activity **Ident2ODBC_metaCP** and then click . The activity object opens. Change the name to **MyMetaCP** and click **OK**.
- Click the activity **Ident2ODBC_ODBCImport** and then click . The activity object opens. Change the name to **MyAgentActivity** and click **OK**.
- Click the job **Ident2ODBC_metaCP** and then click . The job object opens. Change the name to **MyMetaCP**.
- Click the **Operation** tab and uncheck **Delta synchronization**. (For easier testing it makes sense to switch off this feature.) Click **OK**.
- Click the **Ident2ODBC** channel and then click . The channel object opens. Change the name from **Ident2ODBC** to **MyWorkflow** and click **OK**. We set the names of channels at a connected directory to the workflow name to help us understand better where a specific channel is used when we look at the channel folder of a connected directory.
- Click the **OutData** channel and open it. Select the **Entry Handling** tab. Set **Add Entries** to **ADD** otherwise the job will not create any entries in the intermediate file. Click **OK**.
- Click **Save** to exit the edit mode.

It is not necessary and does not make sense to edit the second job Ident2ODBC_ODBCImport and the connected directory ODBC because we will replace this job with a job that runs our agent and the ODBC connected directory with our new MyDatabase.

- Click **Close** to exit the structure view.

Now we'll check and modify the other objects we created:

- In the Expert View, open **Workflows** → **MyScenario** → **Source Scheduled** → **ODBC**. The workflow **MyWorkflow** exists. Change the name of the folder from **ODBC** to **MyAgent**.
- Open **Jobs** → **MyScenario** → **Source Scheduled** → **ODBC**. There are two jobs: **MyMetaCP** and **Ident2ODBC_ODBCImport**. Change the name of the folder from **ODBC** to **MyAgent**.

The next step is to set up the new agent MyAgent.

#### 2.2.2.3.3. Setting up the Agent MyAgent

MyAgent is the new agent that we must set up:

- To install your new agent, copy the prepared file **myagent.bat** to the DirX Identity **bin** directory:
- Copy the prepared executable from the directory *install_path***\data\extension\MYAGENT.BAT** to the directory *install_path* **\bin**
- Create the directory *work_path***\MyScenario\Source Scheduled\MyAgent\MyWorkflow\MyAgentActivity**.
- Move the prepared INI file **CONTROL.INI** here (from the *install_path***\data\extension**

directory).

If you are not sure how the *work_path* or your *install_path* is set up, open the object with the name of your computer in the configuration database in the **Configuration** → **DirX Identity Servers** → **C Servers** folder.

- Create a new agent object in the connectivity configuration. In **Connectivity** → **Expert View**, right-click the **Agents** folder and select **New** → **MyAgentType**.
- Set **Name** to **MyAgent** and set **Description** to **MyAgent to handle MyDatabase**.
- Enter the name **myagent.bat** in **Executable**. Using a relative pathname makes it easier to make changes later on if the server must be changed.
- The **Agent Type** field is already set to **MyAgentType.** Set **Directory Types** to **MyDirType** (click the upper button to the right of the field and select **MyDirType** from the pull-down list).
- Click **OK** to create the object.

You have now created the new agent with the agent type **MyAgentType** that can handle the connected directory type **MyDirType**.

- In **Configuration** → **DirX Identity Servers** → **C Servers**, open the object with the name of your computer and link the new agent object to it: click the **Agents** tab, click **Edit**, in the Agents box, click , then select **MyAgent** from the object browser.
- In the Versions box, click and enter **MyAgent 1.0** into the text field that appears at the bottom of the list.
- Click **Save** to store the object.

This step indicates that this agent can be run on this DirX Identity server and which version is installed. The DirX Identity installation procedure performs this step automatically for all DirX Identity agents. You must perform this step by hand for any other agents.

#### 2.2.2.3.4. Setting up the Connected Directory MyDataBase

In this example, we assume that the connected directory MyDatabase resides on the primary server. This means that both the agent and the directory are on this machine. For security and performance reasons, we recommend that you keep the connected directory and the agent together on the same machine. Because there is no network traffic when the agent writes to MyDatabase, it will run at maximum speed.

First, you must create the file directory **C:\MyDatabase** with the Windows Explorer (a setup task). Then you must create a relevant description for it in the Connectivity configuration:

- In **Connectivity** → **Expert View**, right-click the **Connected Directories** → **MyScenario** folder (create the folder if it doesn't already exist) and select **New** → **MyDirType**.
- Set **Name** to **MyDatabase**.
- Link the database to the relevant service object (your machine name: select it from the tree browser in the **Configuration** → **Services** → **System** directory).

- Do not set the **Wizard** and **Viewer Command** fields at this point (we'll do this later on).
- Click **OK** to create the connected directory.

In the list of connected directories, a new entry **MyDatabase** is displayed.

#### 2.2.2.3.5. Setting up a Channel to MyDatabase

Now we must set up the channels to the connected directory. First, we need to set up a related channel MyChannel to the MyDatabase connected directory:

- Right click **Connectivity Configuration Data** and select **Reload object descriptors**.
- Open the **Connected Directories** → **MyScenario** → **MyDatabase** object and click **Channels**.
- Select **New** → **MyDatabase** (**Channel**) and enter **MyChannel** in **Name**.
- Click **OK**.

You have now created a new connected directory MyDatabase and a related channel MyChannel to access this directory.

#### 2.2.2.3.6. Setting up the Job MyJob

Now we need to create a new job object because there is no template for this agent in the DirX Identity Connectivity configuration at this point.

- In **Connectivity** → **Expert View**, right-click the **Jobs** → **MyScenario** → **Source Scheduled** → **MyAgent** folder.
- Select **New** and then **MyAgent**.
- Set **Name** to **MyJob** and **Description** to **Job to import the LDIF file to MyDatabase**.
- As you can see, the related agent is **MyAgent**.
- Enter a fixed command line:
  **-i -t control.ini C:\MetahubData\import.ldif**
  This command line directs the agent to perform an import operation with trace switched on based on the INI file **control.ini** and on the data file **import.ldif**.
- Use the default timeout (2 hours).
- Do not touch any settings in the other tabs of this object and click **OK**.

You have now created a simple job object that can be executed and takes its input from a fixed location in the file system.

Because the agent uses the file **import.ldif**, we need to set up the intermediate connected directory correctly.

- Open the **MyMetaCP** job and then the **Data** object.
- Click **Datafile** in the Files tab, then **Edit** and change the file name from **data.txt** to **C:\MetahubData\import.ldif**. This must be the same path and file name that we set up in the job's command line.

- Click **Save** to store this information.

Now we have done all necessary work. We can do the remaining configuration tasks in the structure view.

#### 2.2.2.3.7. Completing the Configuration

Now we can use the workflow structure view to complete our configuration:

- In **Connectivity** → **Global View**, select the workflow line in the scenario **MyScenario** and open the structure view for **MyWorkflow**.
- You can see that the second activity **MyAgentActivity** does not refer to the correct job, channel and connected directory.
- Click **Edit** and select the job. Click ··· to open the object browser and select **MyJob** from the folder **MyScenario** → **Source Schedule**→ **MyAgent** in the folder **Jobs**. Click **OK**. The link is shown in the structure view but the channel and connected directory objects are no longer displayed.
- Open **MyJob**, select the **Input/Output Channels** tab and create a new line in the **Input Channel** field.
- Click ··· to open the object browser and select the **InData** channel from the **Jobs** → **Source Scheduled** → **My Agent** → **MyMetaCP** → **Data** → **Channels** folder. Click **OK**.
- Create a new line in the **Output Channel** field.
- Click ··· to open the object browser and select the **MyChannel** channel from the **Connected Directories** → **MyScenario** → **MyDatabase** → **Channels** folder. Click **OK**.
- Click **OK** to store the job object.

Now you can see the complete structure of your workflow. Click **Save** to store it.

Our scenario does not yet contain our **MyDatabase** connected directory and our workflow line still connects **MyIdentityStore** and **HR-ODBC**. Let's change it:

- In **Connectivity** → **Expert View**, select the workflow **MyWorkflow** again. Click **Edit**.
- Click ↻ to the right of the **Endpoints** field. This action forces DirX Identity to calculate the endpoints to determine which workflows fit between two connected directories. The value should now be set to **LDAP-MyDirType**.
- In **Connectivity** → **Global View**, select **MyScenario** again.
- Create a new connected directory icon (perform **New Connected Directory**).
- Assign **MyDatabase** to it.
- Create a workflow line between **MyIdentityStore** and **MyDatabase**.
- Assign **MyWorkflow** to this line. The arrow is shown now correctly.
- Finally, remove the **HR-ODBC** connected directory from your scenario. It is no longer needed.

To avoid having unused objects in our connectivity configuration, we should now delete the **Ident2ODBC_ODBCImport** job in the Expert View (because it is no longer used):

- In **Connectivity** → **Expert View**, open the **Jobs** → **MyScenario** → **Source Scheduled** → **MyAgent** folder and click the **Ident2ODBC_ODBCImport** job. Select and delete it (set the check for references check box). DirX Identity will delete the object because there are no longer references that point to it.

#### 2.2.2.3.8. Test the Newly Created Workflow

You can use the Global View or the Expert View to test the newly created workflow. If you followed the "Getting Started" exercises, you've already learned how to work in the Global View. Let's now try the Expert View.

We assume that the connected directory and the Identity Store are up and running. We also should check that the connected directory **MyDatabase** is empty before we run the workflow for the first time: examine the directory **C:\MyDatabase** and delete any files.

- In **Connectivity** → **Expert View**, open the **Workflows** folder.
- Right-click **MyWorkflow** in the **MyScenario** → **Source Scheduled** → **MyAgent** folder and select **Run Workflow**.
- The workflow is running. You can watch the workflow's progress by viewing the activities in the Structure tab. Wait until both activities change their color to green. In this case, the second activity shows light red, which indicates a warning from the **stdout** message of the agent.
- In **Connectivity** → **Monitor View**, open **MyScenario** → **Source Scheduled**→ **MyAgent** → **MyWorkflow**.
- Click the entry below and check the attributes. You can see the start and end time and the **Result**. This value should be **closed.completed.ok**.
- If you see another value, something is wrong in your configuration. DirX Identity can help you find the error. Look at the **Remark** field. Normally you'll find a **Workflow execution terminated (at least one activity failed)** error.
- Double-click the workflow entry to open it. You should see entries for the first and the second activities. You can click on each entry and check the result. The **Result** field should show **closed.completed.ok**
- If this is not the case, look at the **Remark** field. You will see one or more error messages. You cannot read long messages directly in these fields. Click the bottom icon to the right of the table and a configurable text editor will open.
- The messages give you hints about where your agents failed. Try to fix the problem and run the workflow again.
- After some debugging, you should have a running workflow.

In our case, there is a message in the **MyAgentActivity** status entry. It gives a hint that the agent reported something at stdout. Check the process info file for details:

- Change to the **Trace** tab and click the icon after **Process Data**. The process info file opens.

- You can see that our batch file reported that a file was copied successfully.
- Look at the work path in the file system (installation area). Open the directory **work**, then **MyScenario** → **Source Scheduled**→ **MyAgent** → **MyWorkflow**. Click the **MyAgentActivity**. You should see two files: the INI file that we copied here by hand and the **trace.trc** file. Open this file and you'll see that our **-t** parameter in the command line worked.
- Now look at the status area (installation area). Navigate to the status path and open the **status** directory and then locate the entry that begins with **MyWorkflow**. Click **MyAgentActivity**.
- You'll find there the **ProcessInfo.txt** file that we examined from the status entry in the Monitor View as well as the **import.ldif** file that DirX Identity automatically copied here (the flag Copy to Status Area is set for this file).
- If you click the **MyMetaCP...** folder you'll see many more files that the system copied after the **Metacp** job executed. Because this job is integrated at level 3, all files are saved: all Tcl files, the attribute configuration files and the data file **import.ldif**.
- Now examine our **MyDatabase** connected directory. It is located in the path **C:\MyDatabase**. Open it and you will find the **data.ldif** file that our agent copied there (this is the operation that generated the message **1 file copied**).

You can see that a lot of events occur even when we have only integrated our agent at level 1. Let's now go on to integration level 2.

#### 2.2.2.4. Integrating to Central Edit (Level 2)

This part of the tutorial demonstrates how to integrate the sample connectivity scenario to the "central edit" level. The following figure highlights the objects that are necessary for the sample connectivity scenario to take it from the "simple execution" level to the "central edit" level.

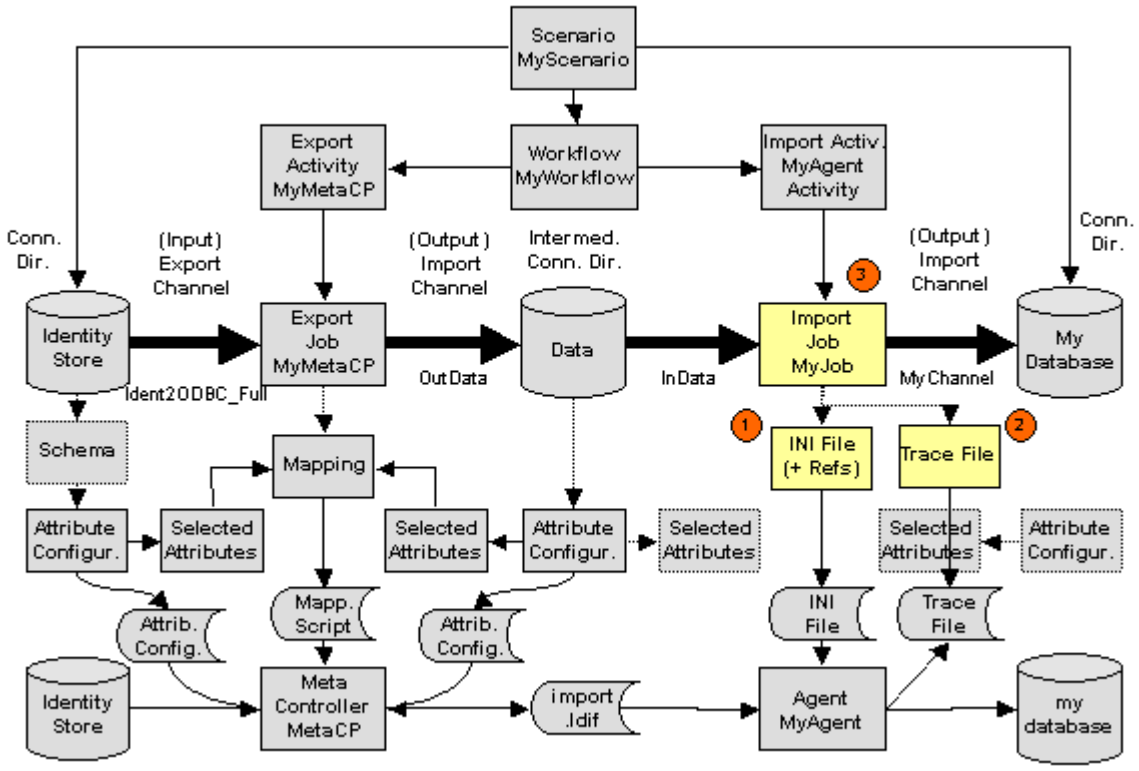



*Figure 6. Level 2 Integration*

To integrate our sample connectivity scenario to "central edit", we'll perform the following tasks:

- Register and import the INI file (1)
- Register the trace file (2)
- Modify the MyJob object (3)
- Test the modified workflow

#### 2.2.2.4.1. Registering and Importing the INI File for MyJob

To register and import the INI file for **MyJob**:

- In **Connectivity** → **Expert View**, open the **Jobs** → **MyScenario** → **Source Scheduled** → **MyAgent** folder and right-click **MyJob**.
- Select **New** → **INI file**.
- A property dialog opens. Do not change the name. Enter **Import INI file** into **Description**.
- Click the File tab and enter **control.ini** into **Filename**. Do not change any other properties.
- Click the **Content** tab and then click **Import text**. A file selection box opens. Navigate to the location where your INI file is stored (in the work area: *work_path***MyScenario\Source Scheduled\MyAgent\MyWorkflow\MyAgentActivity**) and select it. Click **Open**. The content of the INI file is now imported into the

Connectivity configuration and shown in the Content tab.

- Click **OK** to save the new INI file object. .
- Now the INI file is located under the **MyJob** object, but it's not yet linked to it correctly. To link it to the job object:
- Click the **MyJob** object again. Select the **Input/Output Channels** tab and click **Edit**.
- Click  to the right of the **Inifile** field.
- A dialog opens that allows you to navigate through the DirX Identity Connectivity configuration data to the required object to be linked (the current location should be the job object). Open it, click the INI file object that you previously created, and then click **OK**.
- Click **Save** to store the object.

Now the customer-specific agent's INI file with its content is completely contained in the Connectivity configuration. You can view or edit it here with DirX Identity Manager. Any time you run the workflow, this INI file is generated to the file system before your agent executes.

#### 2.2.2.4.2. Registering the Trace File for MyJob

To register the trace file for **MyJob**:

- Open the **Jobs** → **MyScenario** → **Source Scheduled**→ **MyAgent** folder from the Expert View and right-click **MyJob**.
- Select **New** → **File**. A property dialog opens. Change the name to **Trace File**, enter **Trace file for MyAgent** into **Description**.
- It is important to inform DirX Identity about the file name of the trace file. Enter **trace.trc** into **Filename**. DirX Identity will search for a file of this name and save it into the status area (but only if **Copy to Status Area** is checked).
- Do not change anything else and click **OK** to save the new object. It is located under the **MyJob** object but not yet linked to it correctly.

The last step in this sequence is to link the trace file object to the job object.

- Click the job object again. Select the **Tracing** tab and click **Edit**.
- Click the last icon after the **Tracefile** field.
- A dialog opens that allows you to navigate through the DirX Identity Connectivity configuration data to the object to be linked (the current location should be the job object). Open it, click the **Trace File** object you previously created, and then click **OK**.
- Click **Save** to store the object.

Now the trace file is described in the Connectivity configuration. Each time you run the workflow, this trace file should exist after your agent executes.

#### 2.2.2.4.3. Modifying the Job MyJob

Now we'll change the command line to use the correct file names:

- Click **MyJob** and then click **Edit**.
- Change the command line to:
  **-i -t "<?Job@IniFile-DN@FileName/>" "<?Job@InputChannel-DN@SelectedFile-DN@FileName/>"**
  do not forget the quotes around the filenames!

Our command line has introduced some references. The first reference retrieves the filename by looking at the job object, following the reference to the INI file (@IniFile-DN) and taking the content of the **FileName** field as the filename. Now you can change the filename of this object at any time, and the reference will adjust the command line correctly.

The second reference works similarly: it starts with the job object again, follows the reference to the InputChannel (@InputChannel-DN), then to the selected file in the connected directory (@SelectedFile-DN) and takes the content of the **FileName** field as the filename. (See Customizing Object References.)

Both parameters make our job definition more flexible because we can now edit its elements at the user interface and configuration parameters are adapted automatically.

We also no longer want a fixed location for our intermediate data file. A more flexible concept is to transfer the data file in the work directories using the Channel Mapping, which works automatically:

- Set up a reference in your command line (<?Job@InputChannel-DN@SelectedFile-DN@FileName/>) which creates the correct file name for your agent. We just carried out that step.
- Change the fixed location in your data file object under your job object.

To change the fixed file location of the intermediate connected directory:

- Click the **Datafile** entry sub-object under **MyMetaCP** → **Data**.
- Remove the path (**C:\MetahubData**) from the **Filename** field. Leave only the filename **import.ldif**.
- Click **Save**.

#### 2.2.2.4.4. Editing the INI File and Testing the Modified Workflow

Before we run the modified workflow, we'll edit the INI file to contain your name:

- Double click **MyJob**. The sub-objects are shown.
- Click **INI File** and click the **Content** tab.
- Click **Edit** and change **Joe Smith** to your name.
- Click **Save**.

Run the modified workflow again and look at the result in the Monitor View. Nothing has changed in the workflow status entry and in the **MyMetacp** status entry.

- Click the **MyAgentActivity** status entry and select the **Files** tab.
- You can see that there are new entries for the trace file (open it) and the INI file (open it to see that your name is present). This shows that you no longer need to know where your files reside.

#### 2.2.2.5. Integrating to Detailed Customization (Level 3)

Recall that when we set up the central configuration object, we defined a new agent type and a new connected directory type. Now we can use these definitions and define extensions for them at the DirX Identity Manager user interface level. We will only perform this task for the agent type object by defining an XML description for jobs that are based on agents of this type. This way, we can extend the generic job object by defining additional properties in existing tabs or completely new properties for this type of job object. We can also remove properties or complete tabs from this object. See Customizing Connectivity Objects.

Next, we will use these new properties at the user interface level to provide for easy editing. But at the moment there is no mechanism that manages the proper setting of the corresponding attributes in the INI file. To solve this problem, you can use references to connect the definition in the INI file with the property at the user interface level. At runtime, the C++-based DirX Identity Server takes care of the correct substitution of the corresponding values before it starts the agent. The agent is unaware that the INI file is generated each time directly before each execution. Performing this task guarantees consistency between properties at the user interface level and information in configuration files. See Customizing Object References. The following figure highlights the objects that are necessary for the sample connectivity scenario to take it from the "central edit" level to the "detailed customization" level.

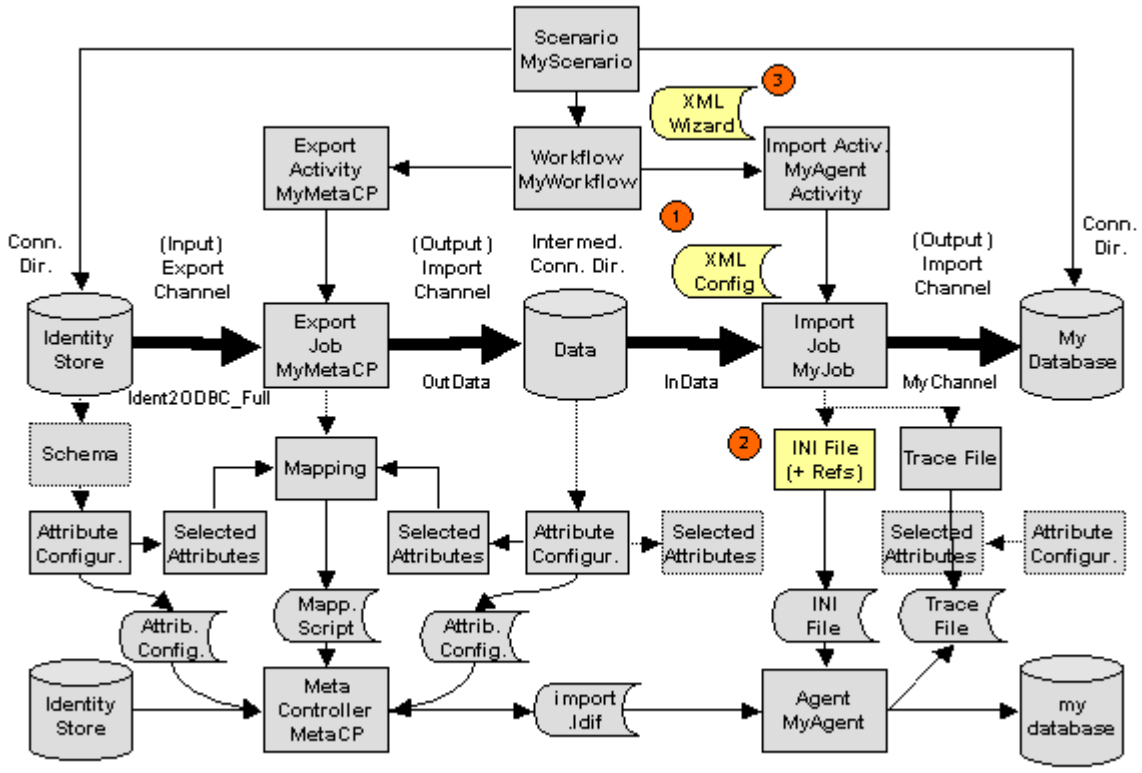



*Figure 7. Level 3 Integration*

To integrate our sample connectivity scenario to "detailed customization", we'll perform the following tasks:

- Create an XML description for property extensions at the job object (1).
- Define references in the agent's INI file to improve consistency (2).
- Test whether changes to the newly added properties with DirX Identity Manager affect the synchronization.
- Create a wizard for the workflow (3).
- Test the wizard.
- Re-run the workflow wizard.
- Export the scenario.

#### 2.2.2.5.1. Creating the XML Job Description

In this step, we define the extensions to the job object that are necessary to resolve the references that will be defined in the next step.

First we create the XML description object for MyAgent:

- Select **Connectivity** → **Expert View**.
- Open **Configuration** → **Agent Types** → **MyAgentType**.
- Right-click **Object Descriptions** and select **New** → **XML File**.

- Enter **myagent-job.xml** into the name field and adjust the description field accordingly.
- Click the **Content** tab and then **Import Text ...**
- In the file dialog, select the file **template-job.xml** in the folder *install_path***data\extension** and click **Open**.
- **Save** the object.

Now you've created an XML description for the MyAgent type. Change it as follows:

- Replace the comment **Template Job** with **MyAgent Job**.

You'll find an entry **superior="Generic-Job"**in the section **<object ...** . This means that your job object's XML description is based on the Generic-Job description (your description inherits all properties from this description). You could also use another base description.

- Go to the **<propertysheet>** section. This is the section that controls the tabs in your job object (these tabs are named **propertypages** here). These pages have a name (**general, deltahandling, channels** and **tracing**) and a related title which is the is the title of tab that appears when you are using DirX Identity Manager (Job Properties, Delta Handling, Input/Output Channels, Tracing).
- You can add propertypages here, delete them, or modify them to create your individual job object. But be careful not to destroy anything that is needed by DirX Identity.
- Do not change the **class** property! This is the default editor (java class) which allows DirX Identity to handle lists of properties.
- The **layout** parameter defines the properties that are contained in your property-page. You can change it to your requirements.
- The **<properties>** section defines the description of the individual property extensions. For the template (the file from which you copied your XML file) only one example dxmSpecificAttributes property is defined (Example).
- Its type is defined as BooleanInteger. Other types as String, Integer and Boolean are possible, too. You can also define default values and much more. The "Customizating Objects" section in the *DirX Identity Customization Guide* describes the possibilities.

Now we understand the basic concept and we are able to change the job description to our requirements.

- We decide to keep the four tabs as they are and add a fifth tab. Copy the fourth tab (tracing) and insert it after the tracing tab.
- Change the tab name to **MyAgentProperties** and the title to **"My Agent Properties"**.
- Now define the properties. Copy the property section with the Example value three times and then change it to (do not forget the <property and /> tags!):

  name="dxmSpecificAttributes(CreateEntries)"
  type="java.lang.Boolean"
  label="Create Entries"
  defaultvalue="true"

name="dxmSpecificAttributes(DeleteEntries)"
type="java.lang.Boolean"
label="Delete Entries"
defaultvalue="true"

name="dxmSpecificAttributes(ModifyEntries)"
type="java.lang.Boolean"
label="Modify Entries"
defaultvalue="true"

name="dxmSpecificAttributes(Trace)"
type="java.lang.String"
label="Trace"
defaultvalue="-t"

- Next we define the tabs that should display these properties. Change the layout definitions of the tabs as follows:

  name="tracing"
  ...
  layout="...,_SP(Trace)"

  name="MyAgentProperties"
  ...
  layout="properties:_SP(CreateEntries),_SP(ModifyEntries),_SP(DeleteEntries)"

We define only the Trace property in the tracing tab. All other properties are defined for the MyAgentProperties tab. The **_SP** is a shortcut to **dxmSpecificAttributes** that makes our definitions easier to read.

- Do not forget to delete the dxmSpecificAttributes(Example) entry in the layout definition of the general tab.
- To activate the new definition, you can either exit the DirX Identity Manager or use the function **Reload Object Descriptors** from the context menu.
- The **MyJob** object should now contain the five tabs together with the four new attributes.

#### 2.2.2.5.2. Defining the References in the INI File

In this step, we define references in the INI file that take the information from the MyJob object.

- Select the **MyJob** object in the **Jobs** → **MyJobs** folder again and open it.
- Select the **INI File** object and then click **Content**.
- Click **Edit** and change the following lines to:

  ; Created by <your name> at <?date/>:<?time/>

  and

CreateEntries=<?Job@SpecificAttributes(CreateEntry)/>
DeleteEntries=<?Job@SpecificAttributes(DeleteEntries)/>
ModifyEntries=<?Job@SpecificAttributes(ModifyEntries)/>

- Click **Save** to store the object.

The created references have the following meanings:

- Date and time are set to the actual date and time.
- Three special attributes at the job object affect the settings in the INI file. At runtime, DirX Identity goes to the MyJob object and examines the requested multi-value attribute **SpecificAttributes** (in this case, it searches for or a name like CreateEntries in the first part). It then takes the value and replaces the reference in the INI file with it.

For example, suppose that SpecificAttributes contains the values:

CreateEntry true
DeleteEntries false
ModifyEntries true

Then the reference

DeleteEntries=<?Job@SpecificAttributes(DeleteEntries)/>

is replaced by

DeleteEntries=true

There is an error in the second line: the **CreateEntry** value should be **CreateEntries**. We'll show how DirX Identity reports this error later on.

You can reference fixed attributes in objects (examples were shown in the command line references).

The last step is to define the trace parameter in the command line to be a reference:

- Click **MyJob** and then click **Edit**.
- Change the **-t** parameter in the command line to **<?Job@SpecificAttributes(Trace)/>**. Make sure that there are blanks before and after this value.
- Click **Save** to store the object.

#### 2.2.2.5.3. Changing a Property and Testing the Modified Workflow

Before we run the workflow again, we'll change one of our new properties:

- Click **MyJob**, click the tab **My Agent Properties**, and then click **Edit**.

The three checkboxes are surrounded by a red border. This means that the value is not defined (no attribute in the database exists that has either the value true or false). To set one of these values, click it; to reset

> it, you must click twice.

- Check all three check boxes and **Save** the object. This is necessary to create SpecificAttribute entries for all three attributes. This action is only necessary for objects that already exist. New objects contain the default values and will work immediately.
- Click **Edit** again.
- Uncheck **Delete Entries**.
- Change to the **Tracing** tab and enter **-t** into the **Trace** field.
- Click **Save** to store the object.
- Now run the workflow again and watch it in the **Structure** tab.
- The last activity turns red because an error has occurred (we know which one it is).
- View the result in the Monitor View (click the bottom icon to the right of **Remark** to display the messages in an editor window).
- The error message tells us that **CreateEntry** is not a valid Property. Instead we must write **CreateEntries**.
- Correct the reference in the INI file and run the workflow again.
- View the result in the Monitor View.
- Look at the INI file in the **Files** tab of **MyAgentActivity** and open it. It should reflect the referenced properties from the job object.

#### 2.2.2.5.4. Creating a Workflow Wizard

The next task is to define a wizard for the workflow **MyWorkflow**. The best way is to use an existing wizard and change it. In this case, we use the wizard from the **Ident2ODBC** workflow because we copied the workflow from this object.

- Click **Connectivity** → **Expert View**
- Open **Configuration** → **Agent Types** → **ODBC** → **wizards**
- Right-click **wf-LDAP-ODBC.xml** and select **Copy Object**.
- Set **Name** to **wf-LDAP-MyDirType.xml** and adjust **Description**. Click **OK**.
- Move the object to the wizards folder of your MyAgentType object.
- Select it again and click **Edit**.
- Change to the **Content** tab.
- Remove the line:
  <illustrator name="Import" title="Import Properties" />
- Remove the entire section that begins with:
  <step name="Import" ...
- Add a new step after the Export Tracing step:

  <step name="MyAgentProperties" title="Set MyAgent Properties"
  illustrator="MyAgentProps"
  node="dxmActivity-DN[dxmIsEndActivity=true]@dxmRunObject-DN">

<description>
Set the specific parameters for MyAgent. When done, click on the 'Next >>' button to continue.
</description>
<propertypage class="siemens.dxm.gui.components.PropertyPageGeneric" reference="MyAgentProperties" />
<prestep class="siemens.dxm.wizard.PreStepImpl" />
<poststep class="siemens.dxm.wizard.PostStepImpl" />
</step>

- Add this line to the illustrator section after the Export Tracing line:
  <illustrator name="MyAgentProps" title="MyAgent Properties" />
- **Save** the object.

We have created the wizard but it is not yet referenced from our workflow. These steps set the reference:

- Select **MyWorkflow** and click **Edit**.
- Click ... to select the **wf-LDAP-MyDirType.xml** wizard.

Now you can test the wizard.

#### 2.2.2.5.5. Testing the Workflow Wizard

To test the wizard:

- Click the workflow line in **MyScenario**.
- Select **MyWorkflow** → **Configure**.
- Check all the steps in the wizard.

You can add, modify or delete other steps if you extend your agent or workflow.

With this step, the integration of your agent and connected directory at the highest level (level 3) is complete.

#### 2.2.2.5.6. Running the Workflow Again

Now we should run the workflow again to make sure that everything is correct:

- Run the workflow.
- Watch the result in the **Structure** tab of the run window. The second activity turns to red.
- Click this activity. The status entry opens and you can see a **Status** of **closed.completed.error**.
- You can see that the **Remark** field contains a lot of information. Click the last button to the right of this field. A text editor opens.
- DirX Identity tells you that the retrieval of the job configuration data (the INI file, see second line) did not work correctly. The third line contains the most relevant

information:
... while evaluating expression Job@SpecificAttributes(CreateEntry) in line 5 at position 16.
... evaluation of @dxmSpecificAttributes(CreateEntry) failed: No matching value found.

- DirX Identity tells you that there is an error in line 5 at position 16 of the INI file. The Specific Attribute CreateEntry does not exist. Of course this attribute's name is CreateEntries.
- Change the INI File content and check with another run that everything is now correct.

Your scenario is now complete and tested up to some level. Next you should prepare it for export.

#### 2.2.2.5.7. Exporting Your Scenario

You can distribute your scenario by exporting it to an LDIF file. There are different ways to do this, but we recommend that you use the collection mechanism because it allows you to have full control over the amount and depth of the objects you'd like to transfer. As discussed in "Reusing the Sample Domain", a collection allows you to define a set of objects and subtrees that you can export to an LDIF file for subsequent transfer to a target location, either another DirX Identity connectivity domain or a configuration management system like Clearcase.

- Open **Connectivity** → **Expert View**.
- Click **Collections**.
- Select **New** → **Collection** (you could first create a folder for your collections).
- Enter **MyScenario Collection** into **Name** and enter a **Description**.
- Click ... to the right of **Path**. Select a location at which to store the LDIF file and set the name for the file.
- **Save** the new object.

Now you can define the collection in detail:

- **Edit** the object again.
- Click the **Objects** tab.
- Create a new line and select the collection itself. This action enables the collection to transfer its own definition to the target location (i.e., the definition is now part of the LDIF file). Otherwise you would lose this information after the transfer. If you created a folder before, do not forget to include this folder into your collection definition here (otherwise the folder is missing at the target location).
- Click the **Subtrees** tab.
- Add the following subtrees to your collection:
  Agents/MyAgent
  Configuration/Agent Types/MyAgentType
  Configuration/Connected Directory Types/MyDirType
  Configuration/Services/MyScenario

Connected Directories/MyScenario
Jobs/MyScenario
Scenarios/MyScenario
Workflows/MyScenario

- Click **Save**.
- Select **Export Collection**.The collection is exported to the LDIF file.

Check the file content to verify that all objects and subtrees are exported correctly.You can now transfer the file to the target location.

# 2.3. Maintaining the Privilege Structure

Now we'll perform some maintenance tasks in the privilege structure to illustrate the features of DirX Identity in this area.In this exercise, we will:

- View the privilege structure to see the privilege data that needs to be changed
- Create separate privileges for the My-Company Sales and Professional Services groups

## 2.3.1. Checking the Privilege Structure

Before we start to maintain the privilege structure, we'll tour My-Company's **Privilege** view to familiarize ourselves with the sample domain in this area.You'll see that some of the data in the structure is not in an optimal state.We'll correct this data in this exercise.

- Click **Provisioning** → **Privileges**. You'll see folders for **Roles**, **Permissions** and **Groups**.

#### 2.3.1.1. Checking My-Company's Department-Specific Roles

First, we'll look at My-Company's Department-Specific role structure and compare it with the roles assigned to a Professional Services user:

- Open **Provisioning** → **Privileges** → **Roles**. In addition to the query folders (Errors and ToDos), you'll see folders for My-Company's B2B Roles and Corporate Roles.
- Open **Corporate Roles** to view the subfolders.
- Open the **Administration** folder. You can see 8 administrative roles for DirX Identity itself (they start with DXR) and some more for other target systems. The owner of all of these roles is Nik Taspatch. Because all roles require approval (the flag is set), Nik must approve these roles when they are assigned to users because he is responsible for these resources.
- Open the **Department Specific** folder. It contains a specific role for most of the departments. The owner is the department manager. You will note that roles for IT and Professional Services are missing. How does that work?
- Click **Provisioning** → **Users**. Open **My-Company** → **Professional Services**. Click **Auffret Jean-Marc** and then click the **Assigned Roles** tab. Besides the Internal Employee role, only the Sales role is assigned to this user and all the other users of this department because the department does not have its own role.

- This works almost perfectly as long as Sales and Professional Services employees use the same services and tools. Click **Auffret Jean-Marc** and then click **Edit**. Scroll down in the **Available Roles** tab to the **Parking Place Munich** role. Select it and assign it to this user. Now click **Save**. You will encounter an error **No group matches** when the Group File Share permission is to be resolved (the full DN is cn=Group File Share,cn=General,cn=Corporate Permissions,cn=Permissions,cn=My-Company).
- Click **Cancel** and then **Reset**.
- Click the **Assigned Groups** tab. You can see the **FS Sales** group, which is strange because the error message told you that there is no matching group.

#### 2.3.1.2. Checking My-Company's Group File Share Permission

Now we'll take a look at the Group File Share permission and the roles and groups assigned to it to find the reason for that error message:

- Click **Provisioning** → **Privileges** and open **Permissions** → **Corporate Permissions** → **General**. Click **Group File Share** and then click the **Roles** tab. You can see that the role Internal Employee uses this permission.
- Now click the **Match Rules** tab. You can see that the ou and c attributes are used to resolve this permission. For Jean-Marc, the values are **ou=PS** and **c=ES**.
- Click the **Assigned Groups** tab. You can see two FS Sales groups. Click  at the end of the row to check these two groups. The first one matches c=US and OU=SA. The second one matches c=DE, ES, FR, GB, IT and OU=SA. This explains the error message. There is no matching group for ou=PS (you can check the other groups as well to verify this). But where does the FS Sales group come from?

#### 2.3.1.3. Checking My-Company's Sales Tasks Permission

To find where the FS-Sales group comes from, let's take a look at My-Company's Sales Tasks permission structure:

- Click the **Sales Tasks** permission in the **Department Specific** folder and then the **Assigned Groups** tab. It uses the same two FS Sales groups but the match rule uses only the c attribute. The resolution of this permission assigns the FS Sales group of the Windows Domain Europe target system.
- Let's check to see how the Sales Tasks permission is assigned to the user. In the **Roles** tab, you can see that the **Sales Tasks** role uses this permission.
- Check the **Assigned Roles** tab of the user to see that this role is assigned by a rule (see the **Assigned by** column).
- Click the **Policies** view. Open **Policies** → **Rules** → **Role based scenario** → **Corporate**.
- Click **Sales Tasks**, and then click the **Filter** tab. The filter searches all users under the My-Company subtree with ou=SA or ou=PS. In combination with the ou matching rule of the FS Sales permission this works well (the total filter is (ou=SA or ou=PS) and c=ES).

#### 2.3.1.4. Identifying the Maintenance Tasks

From the tour we just took of My-Company's privilege structure, we can see two problems that need to be resolved:

- The Professional Services department does not have its own file share FS Professional Services. This is the reason for the error message we received.
- The FS Sales group is assigned twice, once by the Group File Share permission and once by the Sales Tasks permission. This assignment duplication is not transparent.

The powerful features of DirX Identity can sometimes result in erroneous situations as, for example, the double assignment of a group. Nevertheless we can see that DirX Identity provides a lot of features to find the cause of such problems.

### 2.3.2. Correcting the Problems with the Privilege Structure

Now we'll fix the problems we found in the last section. First, we'll remove the FS Sales groups from the Sales Tasks permission and then we'll create a file share privilege structure for the Professional Services department.

#### 2.3.2.1. Correcting the Sales Tasks Permission

To remove the FS Sales groups from the Sales Tasks permission and consequently from the related users:

- Click the **Privileges** view.
- Open **Permissions** → **Corporate Permissions** → **Department Specific** → **Sales Tasks** and select the **Assigned Groups** tab.
- Click **Edit**, select the two FS Sales groups in the lower pane and remove them.
- Click **Save**.

DirX Identity removes the groups directly from the related set of users. Now let's check the Users view to make sure:

- Click the **Users** view. Open **Users** → **My-Company** → **Sales** → **Sales Europe** → **Fani Shelby** and select the **Assigned Groups** tab. The group FS Sales is still there but in state DELETED. Subsequent synchronization of this information to the target system Windows Domain Europe would remove the group membership there.
- Open **Users** → **My-Company** → **Sales** → **Sales Europe** → **Klarmann Bruno** and select the **Assigned Groups** tab. The group FS Sales is there in state ADD. This comes from the Group File Share permission that works only for internal employees. Shelby Fani is a contractor.

#### 2.3.2.2. Creating a File Share Privilege Structure

Now we'll create a separate privilege structure for the Professional Services department. Because this privilege structure is similar to the Sales privilege structure, we can copy many of the objects we'll need for our new structure from the Sales privilege structure and modify them afterwards. We'll start by creating the FS Professional Services group and

then copy and modify the Sales Tasks permission and the Sales Tasks role.

#### 2.3.2.2.1. Creating the Groups

First we'll create a new FS Professional Services group in the Windows Domain Europe and Windows Domain USA group folders:

- Click the **Privileges** view.
- Open **Groups** → **Windows Domain Europe**.
- Right-click the **General** folder and select **New** → **Group**. A dialog opens.
- Enter **FS Professional Services** in **Name** and **Group file share for professional services** in **Description**. Click **OK** to store the new group.
- Click **Edit** and enter some additional values. Create 5 new lines in **Country** and select **DE**, **ES**, **FR**, **GB** and **IT**. Enter **PS** into **Organizational Unit**. This setting means that this group can be selected when the organizational unit is set to **PS** and to one of the country values.
- Enter the link from **Owner** to **Bellanger Lionel**.
- Click **Save** to store the modified object.

DirX Identity reports an error in the **Error** field: **Group is invalid: it is not referenced by any permission** because the group is not referenced by a permission.

- Open **Groups** → **Windows Domain USA**.
- Right-click the **General** folder and select **New** → **Group**. A dialog opens.
- Enter **FS Professional Services** in **Name** and **Group file share for professional services** in **Description**. Click **OK** to store the new group.
- Click **Edit** and enter some additional values. Create a new line in **Country** and select **US**. Enter **PS** into **Organizational Unit**. This setting means that this group can be selected when the organizational unit is set to PS and country is set to US.
- Enter the link from **Owner** to **Bellanger Lionel**.
- Click **Save** to store the modified object.

DirX Identity reports an error in the Error field: **Group is invalid: it is not referenced by any permission** because this group is not referenced by a permission, either.

#### 2.3.2.2.2. Adding the New Groups to the Group File Share Permission

Next, we'll add the new file share groups to the Group File Share permission:

- Click the **Privileges** view.
- Open **Permissions** → **Corporate Permissions** → **General**.
- Click **Group File Share** and then the **Assigned Groups** tab.

- Click **Edit**.
- Select **Name** in **Search for** and enter **fs** into the value field. Click . Four groups are displayed in the upper pane. Select both **FS Professional Services groups** and then click  to move them to the lower pane.
- Click **Save**. DirX Identity processes the change. All users that have this permission assigned are resolved anew.
- Check Fani Shelby from the Sales department again to see that no new group has been added during the resolution. Check Auffret Jean-Marc from the Professional Services department to see the result. He has the new FS Professional Services group (the FS Sales group is still in the DELETE state). All the other members of this department have one of the two groups depending on the country in which they reside.

##### 2.3.2.2.3. Copying the Sales Tasks Permission and Role

Now we'll copy the Sales Tasks permission and role and adapt them to Professional Services:

- Click the **Privileges** view.
- Open **Permissions** → **Corporate Permissions** → **Department Specific**.
- Right-click **Sales Tasks** and select **Copy Object**. A dialog opens. Change the name to **Professional Services Tasks** and click **OK**. The permission is copied.
- Click **Edit** and change **Description** to **Standard tasks in the professional services department** and **Owner** from **Straub Hatty** to **Bellanger Lionel**. Click **Save** to store the object.
- Open **Roles** → **Corporate Roles** → **Department Specific**.
- Right-click **Sales Tasks** and select **Copy Object**. A dialog opens. Change **Name** to **Professional Services Tasks** and click **OK**. The role is copied.
- Click **Edit** and change **Description** to **Standard tasks in the professional services department**, **Role ID** to **0308** and set **Owner** to **Bellanger Lionel**. Change the fields in the **Details** tab to values that fit for Professional Services.
- Click the **Assigned Permissions** tab. The **Sales Tasks** permission is still linked to this role. Remove it and move the **Professional Services Tasks** permission to the lower pane.
- Click **Save** to store the object.

If you expect a resolution process to occur after the save, this action won't occur because the new role is not assigned to any user. Currently all members of the Professional Services department have the **Sales Tasks** role assigned. We need to create a provisioning rule and run it.

##### 2.3.2.2.4. Creating the Provisioning Rule

Next, we'll update the Sales Tasks provisioning rule to remove Professional Services from it, then copy and modify it to create a new Professional Services provisioning rule:

- Click the **Policies** view.

- Open **Policies** → **Rules** → **My-Company** → **Role based scenario** → **Corporate** and select the **Sales Tasks**.
- Click the **Filter** tab. This policy works for all users in the My-Company tree that are either in department Sales (SA) or Professional Services (PS). We need to change this policy.
- Click **Edit** and open the filter editor with the  button. Click the button at the end of the PS line, select **Delete row** and then **OK**. Change the description field (remove "and Professional Services") and then click **Save**.
- Click **Sales Tasks** in the tree again and select **Copy Object**. A dialog opens. Change the name to **Professional Services Tasks** and click **OK**.
- Click **Edit**, change the **Search Filter** in the **Filter** tab to **PS**, select the **Professional Services Tasks** role in the **Privileges** tab instead of the **Sales Tasks** role and change **Description**. Click **Save** to store the object.

Now both policies fit with the new privileges structure. We need to run the policy execution service.

#### 2.3.2.2.5. Running the Policy Execution Service

To run the policy execution service:

- Click DirX Identity Manager's **Connectivity** view and log in.
- Click **Global View** and then the **Maintenance** scenario.
- Click the line between the two **Identity Store** connected directories and configure the **Policy Execution** workflow. Set **Provisioning Mode** in the **Policy Agent Parameters** step to **Assign Privilege and Resolve**. Set **Trace** in the **Tracing** tab to **2-Errors and Warnings** (this setting is already selected).
- Click **Finish** and run this workflow.
- When the run is complete, click the **Structure** tab and double-click the **PolicyExecution** activity. Click the **Trace** tab and open the trace file. The statistics show that the role **Professional Services Tasks** was granted to 7 users. Also the **Sales Tasks** role affected 7 users which resulted in 7 warnings.
- Check the user **Auffret Jean-Marc**. The role **Sales Tasks** is replaced by **Professional Services Tasks**. The same change holds for the permission.

The groups have not changed up to now because the **Professional Services Tasks** role is an exact copy of the **Sales Tasks** role. This does not make much sense unless we change this role.

#### 2.3.2.2.6. Changing the Professional Services Tasks Permission

To change the Professional Services Tasks permission:

- Click **Privileges**.
- Select **Permissions** → **Corporate Permissions** → **Department Specific** → **Professional Services Tasks** and click the **Assigned Groups** tab.

- We check all the groups and decide that it makes sense to keep these groups for the Professional Services employees.In addition we want to introduce a new group **Professional Services Portal**.
- Select **Groups** → **Intranet Portal** → **Group Portals**.
- Create a new group **Professional Services Portal** with the values **Description**: **Portal pages for professional services**, **Owner**: **Bellanger Lionel**, **Country**: *****.
- Select the **Professional Services Tasks** permission again, click the **Assigned Groups** tab and click **Edit**.Add the **Professional Services Portal** group.Click **Save**.
- Verify that all users of the Professional Services department now have this group.

# 2.4. Changing a Workflow's Structure

You can use DirX Identity Manager's Global View wizards to access most of a workflow's properties.If you need to access specific options or the workflow's configuration files, you can use the Expert View, which presents all of a workflow's objects for viewing and editing.In this section, we'll describe the elements of a workflow's structure using the NewHR2Ident workflow as an example, then we'll show how to use the Expert View's configuration objects from a workflow's structure view to make a change to this workflow.

## 2.4.1. Understanding a Workflow's Structure

Usually, you're only interested in the objects that belong to the workflow with which you're currently working.In this case, you can use the workflow's structure view from DirX Identity Manager's Global View to access the objects in the Manager's Expert View.To access the NewHR2Ident workflow's structure from the Global View:

1. Click **Global View**, then **My-Company** → **NewCompany** in the scenario pane's tree.
2. In the scenario map, right-click the workflow line, then select **NewHR2Ident** → **Show Structure**.This action opens a separate window (this window can stay open while you do other things in another view).The window displays the workflow you are working with and the Identity server (IdS-C) on which this workflow runs.

**Activity** shows all of the activities that this workflow contains. **Identity Server** specifies the server on which the activity will run. (If you have set up a distributed environment during installation, you can easily change the distribution of each workflow by changing the DirX Identity server here).

**Run Workflow/Job** identifies the job that this activity uses. **Channels** identifies the channels that this job uses to access the connected directories. In this case, the **ODBC2Ident_ODBCExport** job reads the **New-HR** connected directory via the channel **ODBC2Ident** (indicated by **<** in **D**irection) and writes the information via the channel **OutData** (indicated by **>** in **D**irection) to an intermediate file directory **Data**.

The next activity **ODBC2Ident_metaCP** reads the information from the intermediate file directory **Data** via the channel **InData** and writes it via the channel **ODBC2Ident** to the connected directory **Identity Store**.

### 2.4.2. Changing a Job's Timeout Value

As you can see, a workflow's structure view allows you to view all of the objects that belong to a workflow definition and the internal relationships between them. You can also use this view to edit all aspects of your workflow or simply view more details. In this section, we'll show you how to make a change to one of the HR2Ident workflow's job objects.

When you're editing a workflow in the structure view, always make sure that you double-click the beginning of an entry because there are hidden buttons located at the end of each entry which, if selected, will execute immediately. If you accidentally click one of these buttons, especially **Delete**, click **Reset**, and then click **Edit** again.

- Click **Edit** to open the **HR2Ident** workflow for editing.
- Double-click the beginning of the top-most job entry in **Workflows/Jobs**.
- Three buttons are displayed at the end of the job entry.You can view the job with the first button, delete it with the second button and change the reference to another job with the third button.Click the first button.
- The job object is opened with all its tabs.Click through the different tabs and view their content.
- Click the **Job** tab again and change **Timeout** to **01:00:00**.This action affects the automatic abort feature of DirX Identity.Now your job can run for 1 hour (the default was 2 hours) because you are sure that the amount of data can always be handled in this timeframe.
- Do not change anything else.Click **OK** to close the job object and then **Save** to exit the edit mode of the structure view.Click **Close** to close the structure view window.

This simple example shows that you can use the structure view to access any detail of your workflow.

## 2.5. Applying SoD Policies

In this section, we'll demonstrate how segregation of duties (SoD) checks work.We'll show you how to:

- Activate SoD checking
- Activate an SoD policy
- Make a privilege assignment that causes an SoD violation and examine the effects
- Check the SoD violation
- Override an SoD violation and examine the effects

### 2.5.1. Activating SoD Checking

First, we'll activate SoD checking if necessary. For performance reasons, this feature is deactivated by default in all customer domains. We'll check its activation using DirX

Identity Manager's Provisioning → Domain Configuration view in the sample domain:

- Log in to DirX Identity Manager's **Provisioning** view group.
- Click **Domain Configuration** → **My-Company**.
- In the **Compliance** tab, click **Edit**.
- Check **Segregation of Duties (SoD) checks** if it isn't already checked.
- Click **Save**.

To make this change effective within DirX Identity Manager itself, you must stop and then restart all of your running Manager instances:

- Click **File** → **Exit** to close your Manager instance(s).
- Start Manager again and log in to the **Provisioning** view.

### 2.5.2. Activating an SoD Policy

Next, we'll activate one of My-Company's SoD policies. The My-Company sample domain has an SoD policy that prohibits a person with the Contractor role from getting the Manager role. We'll activate this policy now with DirX Identity Manager:

- In **Provisioning** → **Policies**, click **SOD Policies** → **My-Company** → **Contractor <> Manager**.
- Click **Edit**, and then check **Is active**.
- Click **Save**.
- When enabling the SoD policy, DirX Identity checks whether there are already violations for this policy. In our example, there are no violations found.

Now let's look at the two privileges:

- In **Provisioning** → **Privileges**, click **Roles** → **Corporate Roles** → **General** → **Manager**. Look at the Approval tab. You can see that **Potential SoD conflict** is checked.
- Click the role **Contractor**. You can see that this role also has **Potential SoD conflict** checked.

### 2.5.3. Assigning a Conflicting Privilege

Now we're ready to test the "Contractor cannot be a manager" SoD policy. Let's find a user with the Contractor role and assign him the Manager privilege. We'll use Lavina Pitton in Information Technology as our example, and use Web Center to make the privilege assignment:

- In an Internet browser, start DirX Identity Web Center.
- Log in as **Taspatch Nik** with the password.
- In **Users** → **Select user**, enter **P** in **Search for**, and then click **Search** to return a list of users whose names begin with "P".
- Select **Pitton Lavina** from the list. Web Center displays a user summary for Lavina.

- In the **Users** menu, click **Assign privileges**.
- Enter **M** in **Search for**, and then click **Search** to return a list of roles that begin with "M".
- Check **Manager**, and then move it to **Assigned roles**.
- Click **Save**. Web Center displays a warning dialog that identifies the Manager <> Contractor SoD violation.
- Click **Save anyway**.

## 2.5.4. Checking the SoD Violation

Now we'll look at the effect of our conflicting privilege assignment. First we'll look at the SoD mitigation workflow, which automatically runs to handle the violation. Then we'll check the result at Lavina Pitton's user object.

### 2.5.4.1. Checking the SoD Mitigation Workflow

First, let's use Web Center to look at the workflow that is generated to handle the SoD violation:

- In **Work List**, click **Show initiated workflows**.
- Click **Pitton Lavina** → **Manager**. This is the SoD mitigation workflow.
- Look at **Running Activities**. A single approval activity is running with two potential participants: Hungs Olivier and Morton Gabriela. One or the other of these two company heads must approve the privilege assignment to override the SoD conflict.

We can use DirX Identity Manager's **Provisioning** → **Workflows** → **Monitor** view to get more details about this workflow:

- Switch to Manager's **Provisioning** view, and then click **Workflows** → **Monitor** → **My-Company** → **Approval** → **Manager Nomination**. In the My-Company sample domain, all requests for the Manager privilege must be approved by either Olivier Hungs or Gabriela Morton. These two people can decide whether or not to override the SoD conflict and approve the Manager privilege for the contractor Lavina Pitton.
- Under **Manager Nomination**, you'll see a folder with today's date. Open it, and then click **Pitton Lavina** → **Manager**.
- The **General** tab displays this workflow's structure. You can see that the Approval by Company-Head activity is the current one (it's highlighted in grey). Right-click it, and then click **Open**.
- Click the **Status Information** tab. In the Participants field, you can see the approvers. You can see Olivier Hungs and Gabriela Morton in the participant list.
- Click **Close**.

### 2.5.4.2. Checking the User's SoD Information

Now let's look at Lavina Pitton's user information in DirX Identity Manager's **Provisioning** → **Users** view:

- Switch to **Provisioning** → **Users**, and then click **My-Company** → **Global IT** → **Pitton Lavina**.
- Click the **SoD Exceptions** tab and look at **Pending SoD violations**. Two violations are listed: one for the **Contractor** privilege and one for the **Manager** privilege. Notice that **Approved SoD violations**, which provides a history of SoD violations for this user, is empty.

Now we'll approve the Contractor <> Manager SoD conflict and examine the result.

## 2.5.5. Overriding the SoD Violation

We'll log in to Web Center as Olivier Hungs and approve the Manager request:

- Switch to DirX Identity Web Center and click **Logout** to log out of Nik Taspatch's account.
- In the **Log in** dialog, enter **Hungs Olivier** in Name and enter the password.
- Click **Work List** → **Task list**. Olivier Hungs task list is displayed.
- Click the **Pitton Lavina → Manager** task.
- Enter **L**. **Pitton is a substitute manager for 6 months** in **Reason** and then click **Accept**.
- There is nothing more to do, so click **Logout**.

## 2.5.6. Checking the Effect of the SoD Violation Exception

Now we'll show the effect of Olivier Hung's SoD violation override from three different perspectives: the SoD mitigation workflow, the user-privilege assignment, and the SoD policy. We'll use DirX Identity Manager for this task.

### 2.5.6.1. Checking the Effect on the SoD Mitigation Workflow

First let's return to DirX Identity Manager's **Workflows** → **Monitor** view and examine the **Pitton Lavina** → **Manager** workflow again:

- Go to **Provisioning** → **Workflows** → **Monitor** → **My-Company** → **Approval** → **Manager Nomination** → *today's date* → **Pitton Lavina → Manager**. Click the **Refresh** button if necessary.
- You'll see that the **Approval by Company Head** activity and the **Apply Changes** activity are now green.
- Right-click **Approval by Company Head** and then click **Open**.
- Click the **Status Information** tab. In the Status field, you can see that the approval activity was successfully accepted.
- Click **Close**.

### 2.5.6.2. Checking the Effect on the User's Privileges

Next, let's go to DirX Identity Manager's Users view to look at Lavina Pitton's privileges:

- Click **Provisioning** → **Users** → **My-Company** → **Global IT** → **Pitton Lavina**. Click the **Refresh**

button.

- Click the **Assigned Roles** tab. The **Manager** privilege is now in the state ENABLED. Notice that it has an **End date** and **Reapproval** is checked. The reason for this is that My-Company's SoD policy requires that all privilege assignments approved despite an SoD conflict must be re-approved after 3 months.
- Click the **SoD Exceptions** tab and look at **Approved SoD violations**. Now it shows that Olivier Hungs has approved Lavina's Manager privilege assignment and shows the reason why.

#### 2.5.6.3. Checking the Effect on the SoD Policy

Finally, we'll go to DirX Identity Manager's **Policies** view and look at the **Contractor <> Manager** SoD policy:

- Click **Provisioning** → **Policies** → **SoD Policies** → **My-Company** → **Contractor <> Manager**.You'll see an entry **Lavina Pitton 001**.Click it..
- **Accepted Violations** shows that Contractor and Manager roles have been assigned for this user and **Approval Details** shows who approved it and why.

From this exercise, you can see that DirX Identity provides the flexibility to create SoD policies and override them when reasonable and necessary.You can also see that it provides several ways to audit SoD violation exceptions.

## 2.6. Re-approving a Privilege Assignment

Many companies have a set of critical privileges that need to be protected from potential misuse or abuse.It is imperative that assignments of these kinds of privileges be periodically re-checked by authorized personnel.DirX Identity's re-approval workflows allow sensitive privileges to be re-approved automatically and consistently across the enterprise rather than on a case-by-case basis.

In this section, we'll show you how privilege re-approval works using the Trainer role that is assigned to My-Company's Professional Services employees.

This section provides information how re-approval works and demonstrates how to:

- Select a privilege for re-approval
- Run a report to identify all users to whom the privilege is assigned
- Initialize the re-approval workflow
- Run the re-approval workflow and examine the results

### 2.6.1. Understanding How Re-approval Works

Re-approval of a privilege assignment is configured at both the privilege and domain levels. When set, the **Reapproval** parameter in a privilege object configures all assignments of that privilege for periodic re-approval. The privilege object also contains parameters for setting an end date or a time period after which assignments of the privilege expire if they

are not re-approved. We'll view and use these parameters in the Trainer role in the next section of this exercise.

The domain configuration object specifies default end dates or time periods to be used for privileges that are marked for re-approval but do not specify individual date or period settings. The domain configuration object also contains a parameter (**Approval period**) that controls the time interval before a privilege assignment's end date or period at which re-approval workflows are to be started. The default for this parameter is 14 days. You can view these parameters with DirX Identity Manager's Provisioning view by clicking **Domain Configuration** → **My-Company** and then clicking the **Timing** tab.

In the sample domain, the re-approval process is carried out by the following request workflows:

- The **InitializeReapproval** workflow, which examines all assignments of privileges marked for re-approval and sets their end date or period attributes.
- The **StartReapprovalWorkflows** workflow, which starts the request workflow that has been configured for the privilege that requires re-approval. In the sample domain, the 4 Eye Approval request workflow is the default workflow to be run for any privilege assignments that require re-approval if no other workflow is configured.

Although we'll run **InitializeReapproval** and **StartReapprovalWorkflows** by hand later on in this exercise, these two workflows are normally run at scheduled intervals in DirX Identity production environments. It's a good idea to schedule them to run at night or over a weekend so that they do not compete with workday network resources. To create a schedule for a workflow, you use the **Connectivity** → **Expert View** to create a schedule configuration object and associate it with a workflow. For example, you can create a schedule that runs the workflow every 24 hours between midnight and 4AM. Go to **Connectivity** → **Expert View**→ **Schedules** → **Default** to see some example schedules. Click **Help** to get information about how to set the schedule configuration object's time controls and the rules for creating schedules.

## 2.6.2. Selecting a Privilege for Re-approval

Now we'll use DirX Identity Manager's Provisioning view to select the Trainer role for re-approval:

- In **Provisioning**, select **Privileges** → **Roles** → **Corporate Roles** → **General** and click **Trainer** to select it.
- Click **Edit**.
- Check **Requires re-approval** in the Certification tab.
- In **Re-approval period**, enter **5** in **Day(s)**. Now all assignments of the **Trainer** role are scheduled to expire in 5 days unless they are re-approved.
- Click **Save**.

## 2.6.3. Identifying the Privilege's Users

Next, we'll use Manager's **Provisioning** → **Privileges** view to run a report on the **Trainer** role

that will identify the people in My-Company who are assigned the role and thus who will be affected by the re-approval process.

- Click **Provisioning** → **Privileges** → **Roles** → **Corporate Roles** → **General**.
- Right-click **Trainer**, and then click **Report**.
- Click **Number of users per role** in the list of report templates.
- Check **Output to viewer**, and then click **Run report**. After a few seconds, DirX Identity Manager displays a report on the number of users who are assigned the **Trainer** role. You can see that 5 users are affected by the role and 4 are direct users. Re-approval will affect only the direct users. If you have followed the getting started tutorial, you will see 6 affected users and 5 direct users.
- Click **Close** to close the report.
- Right-click **Trainer** again, select **Report.** Now click **Users of a role** in the list of templates and check **Output to viewer**. After a few seconds, DirX Identity Manager displays a list of the users who are affected by the **Trainer** role. They are: Straub Hatty in Sales, Costello Marcella and Blander Dyan in Professional Services, Kubalke Leo in Product Testing and Telfer Laura in Human Resources. If you have followed the getting started tutorial, you will see Teacher Mark as well.
  +

Costello Marcella will not be affected by the re-approval process since she is an affected user of the Trainer role but not a direct one. This will be explained in greater detail at the end of this chapter.

- Click **Close** to close the report.

## 2.6.4. Initializing the Re-approval Process

Now we need to initialize the re-approval process. In this step, we run the **InitializeReapproval** workflow, which checks for all assignments of the **Trainer** role and updates the re-approval attributes of the affected users. We'll use DirX Identity Manager's **Connectivity** view to run this workflow and then examine the results in the **Provisioning** → **Users** view:

- Return to the Manager window, click **Connectivity**, and log in if necessary.
- In **Global View** → **My-Company** → **Main**, right-click the workflow line between the two Identity Stores.
- Select **InitializeReapproval** and then click **Run**. After a few seconds, the workflow should complete successfully. (The **InitializeReapproval** activity is displayed in green in the **Structure** tab.)
- Click **Close**.

Let's return to the **Provisioning** view and examine one of users highlighted in our report. Let's look at Laura Telfer:

- Click **Provisioning** → **Users** → **My-Company** → **Human Resources** → **Telfer Laura**.

- Click **Assigned Roles** and look at the **Trainer** role in the list. Notice that **End date** now shows a date at which the role must be re-approved (five days from today), and **Reapproval** is checked.

### 2.6.5. Running the Re-approval Workflow

Next, we'll run the **StartReapprovalWorkflows** workflow, which starts all necessary re-approval workflows:

- Return to the Manager window and click **Connectivity**.
- In **My-Company** → **Main**, right-click the workflow line between the two Identity Stores, select **StartReapprovalWorkflows**, and then click **Run**. After a few seconds, the workflow should complete successfully. (The **StartReapprovalWorkflows** activity is displayed in green in the **Structure** tab.)
- Click **Close**.

### 2.6.6. Checking the Re-approval Workflow Results

Now let's check the results in Manager's Workflows → Monitor tree view:

- In **Provisioning**, select **Workflows** → **Monitor** → **Queries** → **Running Workflows**. You can see that there are 4 (or 5) workflows, one for each user who is assigned the **Trainer** privilege. These workflows were started because the **Trainer** role is set to expire in 5 days, which falls within the 14-day approval period specified for starting re-approval workflows.
- Click **Telfer Laura** → **Trainer**. You can see from the **General** tab that there are two activities in grey - **Approval by User Manager** and **Approval by Privilege Managers**. This is the **4-Eye Approval** workflow we've discussed in "Adding a User".
- Right-click **Approval by User Manager**, then click **Open**.
- Click the **Status Information** tab. Look at the Participant field. You can see that the approvers are Berner Hans, who is Laura's manager, and Ormsby Mary-Jane, who is representative of Hans. The workflow has automatically notified them about the re-approval task, as it has the managers and their representatives of all the other affected users.
- Click **Close** to return to the **General** tab. Right-click **Approval by Privilege Managers** and then click **Open**.
- Click the **Status Information** tab and look at **Participant**. The re-approver here is Costello Marcella, the owner of the **Trainer** role.

Now let's use the Web Center to look at Marcella's task list:

- In Web Center, log in as **Costello Marcella** with the password.
- In **Work List**, click **Task list**. You can see that Marcella has 5 (or 7) tasks in her queue: one for each user who needs his **Trainer** role re-approved since she is the role owner; and an additional task since she is the manager of Blander Dyan. She needs to accept each of these requests for these users to retain their **Trainer** roles. If she rejects any of these requests, the workflow automatically notifies the affected user that his role assignment

has been rejected, and DirX Identity will remove the role assignment.

If Marcella does not respond to these tasks (for example, because she is on vacation), the workflow ends with a timeout error once the end date is reached, and DirX Identity removes the privilege assignments.This is the reason why the default re-approval period is 14 days; in a production environment, the workflow should also be configured to define an escalation path for re-approval to avoid workflow timeouts.

Recall that our **Trainer** role report listed Marcella Costello as an affected user.However, you'll notice that no re-approval workflow was generated for her.Let's look at her assigned roles to see why.

- Click **Self Service** → **Display summary**, and then click on the **Roles** button.You'll see that Marcella has the **Training Manager** role.Let's take a look at this role in more detail.
- Click **Roles** → **Select role**.
- Enter **T** in **Search for**, and then click **Search** to get a list of roles that begin with "T".
- Select the **Training Manager** role and look at **Privileges**.You can see that the **Trainer** role is listed in **Assigned junior roles**, which means that **Training Manager** automatically inherits the **Trainer** role.This is the reason why Marcella appears in the role report as an affected user but does not need her **Trainer** role re-approved: since she has the **Training Manager** role assigned to her, she automatically gets the **Trainer** role.
- Logout and close the Internet browser.

## 2.7. Certifying a Role

Another method for protecting critical privileges from potential misuse or abuse is to certify the privilege itself.In this case, you have to certify the privilege regarding all assigned users.It is imperative that this kind of privilege is periodically checked by authorized personnel.

DirX Identity's certification feature allows sensitive privileges to be checked on a regular schedule or during compliance campaigns.

In this section, we'll show you how a certification campaign works using the **Trainer** role that is assigned to some of My-Company's employees.We'll demonstrate how to:

- Set up a certification campaign for this privilege
- Start the campaign
- Certify the privilege with DirX Identity Web Center or Business User Interface
- Monitor the campaign
- Finish the campaign

You can find more information about privilege certification in the DirX Identity Use Case Document *Certification Campaigns*.

## 2.7.1. Understanding How a Privilege Certification Campaign Works

A privilege certification campaign is configured in both the Connectivity and Provisioning views and consists of the following steps:

- Creating and configuring a certification campaign in the Provisioning view by providing a start date, an approval period, a search criterion for users to be certified and a search criterion for privileges to be certified.
- Enabling the Certification Campaign Controller workflow in the Connectivity view.
- Setting up a schedule for the Certification Campaign Controller workflow as a regular task.

In this exercise, we assume that the My-Company Certification Campaign Controller workflow runs every night. A compliance administrator needs to create, configure and run a certification campaign on selected privileges and selected users. In our scenario, a compliance administrator selects the **Trainer** role and starts a certification campaign. **Marcella Costello**, the role owner, must certify this role.

## 2.7.2. Configuring the Certification Campaign

The first step is to configure the certification campaign feature so that you can run the privilege certification campaign. Configuration tasks include:

- Configuring the SMTP service
- Setting up a schedule for the Certification Campaign Controller workflow
- Creating a thread for the Certification Campaign Controller workflow in the Java-based Server
- Configuring the Certification Campaign Controller workflow

### 2.7.2.1. Configuring the SMTP Service

You need to configure the SMTP service because it's required for sending the campaign notifications. To configure it:

- In the Provisioning view, browse to **Workflows** → **Configuration** → **Services** →**SMTP**.
- In **SMTP host**, provide an appropriate SMTP host address (for example, **localhost** if you have a local SMTP server). For the purpose of this tutorial, you can update the **Map mail address** field. Enter **dummy** to suppress all notification emails or enter a specific email address; for example, your own address to send all the notifications to your mailbox rather than to the calculated ones.

You should clear this field in a production environment.

### 2.7.2.2. Scheduling the Certification Campaign Controller

In a production environment, you should configure a schedule for the workflow and make sure that it runs at least once a day. For this tutorial, we don't need to set up a schedule for this workflow because we'll start it manually.

#### 2.7.2.3. Configuring the Java-based Server for Certification Campaigns

The Java-based Server must start one thread for the Certification Campaign Controller workflow. To configure it to do so:

- In the Connectivity Expert View, browse to **Configuration** → **DirX Identity Servers** → **Java Servers** → **My-Company** → **My-Company-S1-** *yourhost*.
- Click **Edit**.
- In the Resource Families tab, move **CertificationCampaign** from **Available** down to **Selected**. Change the number of threads to **1**.
- Click **Save**.

#### 2.7.2.4. Configuring the Certification Campaign Controller Workflow

Now we'll configure and enable the Certification Campaign Controller workflow:

- In the Connectivity Global View, select the **My-Company** → **Main** scenario.
- Select the workflow line between the two Identity Store connected directories and then select **New** from the context menu. Select the **CertificationCampaignController** workflow.
- The workflow wizard starts and guides you through the configuration options. In the first step, make sure you check **Is Active**. You also need to set the **Cluster** and **Domain** fields. For this tutorial, it's sufficient to specify ***** for both fields. After the last wizard step, click **Finish** to save the new workflow in the My-Company scenario.
- Now restart the Java-based Server so that it loads the configuration for the new workflow and is prepared to start it.

### 2.7.3. Creating the Privilege Certification Campaign

As described in the DirX Identity Use Case Document *Certification Campaigns,* there are many ways to set up and execute certification campaigns. In this tutorial exercise, we use a simple approach: we select the privileges to be certified with search criteria and then set a corresponding filter in a newly created Certification Campaign entry.

- Start DirX Identity Manager and log in to the **Provisioning** view group.
- To create the certification campaign, select **Provisioning** → **Certification Campaigns**. Right click the **Certification Campaign** folder and the select **New** → **Certification Campaign**.
- In the **General** tab:

In **Name**, enter **Tutorial Privilege Certification**.

In **Type**, select **Privilege certification** from the list.

In **Owner**, select user **Hungs Olivier**.

In **Description**, enter **Campaign Certification for Trainer privilege**.

In **Apply Changes**, select **Revoke all manually assigned privileges that are rejected or left uncertified**.

- In the **Status** tab:

In **State**, select **Campaign is in preparation (PREPARING)**.

In **Start Date**, select the yesterdays date from the calendar control. This selection ensures that the Campaign Controller workflow will start the campaign immediately after you start the workflow.

In **Approval Period**, set **Year(s)** and **Month(s)** to **0** and set **Day(s)** to **2**. These fields define the duration of the campaign's approval period; in this case, one day.

Delete any value from **Due Date** if one is present. **Due Date** is automatically calculated and set when the campaign starts.

- In the **User filter** tab:

In **Filter Base**, enter **cn=Users,cn=My-Company**.

In **User Filter**, enter **objectClass="dxrUser"**. This query will return all users in the My-Company domain.

If risk management is enabled in the domain configuration, the certification campaign can use this information to create a certification campaign for users with High Risk. To use risk assessment values, we can add another search attribute to our User Filter that will check the **dxrRskLevel** attribute value for the following values: **0** (Normal Risk), **1** (Low Risk), **2** (Medium Risk) and **3** (High Risk). The following query will return all users with **2** (Medium Risk) or **3** (High Risk):

(&(objectClass=dxrUser)(dxrRskLevel>=2))

In this case, information about each user's risk is available in Web Center pages during the approval period.

- In the **Privilege filter** tab:

In **Filter Base**, enter **cn=Trainer,cn=General,cn=Corporate Roles, cn=RoleCatalogue,cn=My-Company**.

In **Privileges Filter**, enter **objectClass="dxrRole"**.

Click **OK** to commit the changes.

We have selected the Trainer role for the privilege certification campaign. Now the certification campaign is ready to be run.

Yyou can find a demonstration of how to use the Notifications feature in the section "Certifying a User" → "Configuring the Certification Campaign". We won't be using this feature in this tutorial exercise.

## 2.7.4. Starting the Certification Campaign

Because the campaign start date is set in the past, the campaign is ready to start:

- In the Global View of the Connectivity view group, select the workflow line between the two Identity Store connected directories.
- Select the **CertificationCampaignController** workflow and then select **Run** from the context menu. The workflow starts the campaign and creates the necessary certification tasks in the campaign folder.

Now the privilege certification is running and we can perform the certification.

## 2.7.5. Certifying the Privilege

There are two options for performing the certification: with DirX Identity Web Center or with DirX Identity Business User Interface. We'll demonstrate both options in the next sections.

### 2.7.5.1. Certifying the Privilege with DirX Identity Web Center

We act as Marcella Costello and perform the certification task in the Web Center. Marcella is informed about this task via email if the **Approval Start** notification is set to active inside the certification campaign.

- Log in to Web Center as **Costello Marcella**.
- Select the campaign **Tutorial Privilege Certification** from the **My Certification Campaigns** list.
- From **Certification Campaign - Details**, select the **Trainer** entry.
- For **Blander Dyan**, select **Accept**.
- For **Kubalke Leo** and **Straub Hatty**, reject the role by selecting **Reject**.
- For **Telfer Laura**, do not select anything.
- Click **Save changes**.

Now this role is ready to apply changes.

### 2.7.5.2. Certifying the Privileges with DirX Identity Business User Interface

We act as Marcella Costello and perform the certification tasks in the DirX Identity Business User Interface. Marcella is informed about this task via email if the Approval Start notification is set to active inside the certification campaign.

- Log in to the Business User Interface as **Costello Marcella**.
- Click the **My Certifications** widget on the Business User Interface home page.
- Select the campaign **Tutorial Privilege Certification** from the certification campaigns list.
- From the **Certification Campaign** – **Tutorial Privilege Certification** list, select the **Trainer** entry.

- For **Blander Dyan**, check **Approve**.
- For **Kubalke Leo** and **Straub Hatty**, reject the role by checking **Reject**.
- For **Telfer Laura**, do not select anything.
- Click **Save changes**.

Now this role is ready to apply changes.

### 2.7.6. Monitoring the Campaign

We can check the certification campaign status in DirX Identity Manager; the **Trainer** entry is still in **RUNNING** state because user **Telfer Laura** was not certified.

Return to Web Center and accept the role for this user and set **End date** for this role to the current date + 3 days. This time click **Save changes and finish certification** to finish this certification entry. Back in DirX Identity Manager, the **Trainer** entry is in state **APPROVAL .FINISHED**.

### 2.7.7. Finishing the Campaign

Now you can finish the campaign by changing the due date in the campaign entry (we do this because we want to finish the tutorial on the same day):

- Go to the campaign entry **Tutorial Privilege Certification** and select the **Status** tab.
- Click **Edit** and change **Due Date** to today.
- Click **Save** to commit the changes.
- Run the **CertificationCampaignController** workflow in the Connectivity view.

After the workflow has finished, check the campaign status.The Privilege Certification task for the Trainer role should be in the state **FINISHED** and the certification campaign should be in the state **Campaign finished successfully (SUCCEEDED)**.

We can use DirX Identity to check the campaign results:

- For users **Kubalke Leo** (Product Testing) and **Straub Hatty** (Sales), the role **Trainer** was removed.
- For users **Blander Dyan** (Professional Services) and **Telfer Laura** (Human Resources), the role Trainer is assigned.For user **Telfer Laura**, the role is set to be removed after **current date + 3 days**.

## 2.8. Certifying a User

As an alternative to certifying privileges, certification can be based on users.This section shows how to set up a certification campaign for all users working in the Finance department and one other user.

Certification campaigns for users are completely different from the campaigns for privileges.

In this tutorial, we'll demonstrate how to:

- Configure the certification campaign function
- Create a certification campaign
- Start the certification campaign
- Monitor a running certification campaign
- Analyze the result of a finished certification campaign
- Produce reports for a certification campaign

For more details about user certification campaigns and more complex scenarios, see Certification Campaigns.

### 2.8.1. Understanding How a Certification Campaign Works

A certification campaign consists of the following phases:

- Creating the certification campaign. A campaign entry is created and configured with all mandatory attributes.
- Running the certification campaign. After the campaign is started, the certification tasks are created. The approvers are notified and they must certify - approve or reject - privilege assignments.
- Applying the results after the campaign is finished. Depending on the configuration, the rejected and/or ignored privilege assignments are deleted. The certification owner can run a report to document the certification result.
- Physically deleting all the campaign entries from the Identity domain after the configurable status expiration date is reached.

### 2.8.2. Configuring the Certification Campaign

First, you need to configure the Certification Campaign feature and perform the following tasks:

- Make sure the SMTP service is configured (it's required for sending the campaign notifications): In the Provisioning view, browse to **Workflows** → **Configuration** → **Services** →**SMTP**. In **SMTP host**, provide an appropriate SMTP host address (for example, **localhost** if you have a local SMTP server). For the purpose of this tutorial, you can update the **Map mail address** field. Enter **dummy** to suppress all notification emails or enter a specific email address; for example, your own address to send all the notifications to your mailbox rather than to the calculated ones.

  You should clear this field in a production environment.

- Set up a schedule for the Certification Campaign Controller workflow. For the purpose of this tutorial, we do not need a schedule for this workflow, because we'll start it manually. For production, you should configure a schedule for the workflow and make sure that it runs at least once a day.

- Make sure the Java-based Server starts one thread for the Certification Campaign Controller workflow: In the Connectivity view, browse to **Configuration** → **DirX Identity Servers** → **Java Servers** → **My-Company** → **My-Company-S1-** *yourhost*. Click **Edit**. In the Resource Families tab, move **CertificationCampaign** from **Available** down to **Selected**. Change the number of threads to **1**. Click **Save**.
- Configure and enable the Certification Campaign Controller workflow: In the Connectivity Global View, select the **My-Company** scenario. Select the workflow line between the two Identity Store connected directories and then select **New** from the context menu. After you have selected the Certification Campaign Controller workflow, the workflow wizard starts and guides you through the configuration options. In the first step, make sure you check **Is Active**. You also need to set the **Cluster** and **Domain** fields. For this tutorial, it's sufficient to specify ***** for both fields. After the last step, click **Finish** to save the new workflow in the **My-Company** scenario.

Restart the Java-based Server so that it loads the configuration for the new workflow and is prepared to start it.

### 2.8.3. Creating the User Certification Campaign

The goal of this certification campaign is to certify the manual assignments for the Finances department and for two users from Customers: Bellosa Marco and Maskery Guy.

To create the certification campaign, select **Provisioning** → **Certification Campaigns**. Right click the Certification Campaigns folder and then select **New** → **CertificationCampaign**.

A wizard opens and lets you enter the necessary data:

- In the **General** tab:

In **Name**, enter **Tutorial User Certification**.

In **Type**, select **User certification** from the list.

In **Owner**, select user **Hungs Oliver**.

In **Description**, enter **Campaign Certification for Users in department Finances and some selected customers**.

In **Apply Changes**, select **Revoke only rejected privileges that were manually assigned**.

- In the **Status** tab:

In **State**, select **Campaign is in preparation (PREPARING)**.

In **Start Date**, select **yesterday** from the calendar control. This selection ensures that the campaign controller workflow will start the campaign immediately after you start the workflow.

In **Approval Period**, set **Year(s)** and **Month(s)** to **0** and set **Day(s)** to **2**. These fields establish the duration of the campaign's approval period; in this case, one day.

Delete any value from **Due Date**, if one is present. **Due Date** is automatically calculated and set when the campaign starts.

- In the **User filter** tab:

In **Filter Base**, enter **cn=Users,cn=My-Company**.

In **User Filter**, enter **(&(objectClass=dxrUser)(|(ou=Finances)(cn=Bellosa*)(cn=Maskery*)))**. This query will return all users in the Finances department and the users Bellosa and Maskery.

If risk management is enabled in the domain configuration, the certification campaign can use this information to create a certification campaign for users with High Risk. To use risk management values, we can add another search attribute to our User Filter that will check the **dxrRskLevel** attribute value for following values: **0** (Normal Risk), **1** (Low Risk), **2** (Medium Risk) and **3** (High Risk). The following query will return all users with **2** (Medium Risk) or **3** (High Risk):

(&(objectClass=dxrUser)(|(ou=Finances)(cn=Bellosa*)(cn=Maskery*))(dxrRskLevel>=2))

In this case, information about each user's risk is available in Web Center pages during the approval period.

- In the **Privilege filter** tab:

In **Filter Base**, enter **cn=RoleCatalogue,cn=My-Company**.

In **Privileges Filter**, enter **(objectClass=dxrRole)**.

- Click **OK** to commit the changes.

Next, we'll enable several notifications for this campaign. In the newly created Tutorial User Certification campaign, go to the Notifications container and check **Is Active** in the General tab for the following entries:

- Campaign Start - this notification will be sent to the Certification Campaign owner, who is the user Hungs Oliver, when the Certification Campaign starts. The email contains information about campaign such: name, description, start date, due date, and other information.
- No Approver - this notification will be sent to the Certification Campaign owner (by default) right after the campaign starts and all certification entries are created. The email contains a list of certifications which are not started (RUNNING state), and are moved to failed prepare (FAILED.PREPARE state). The reason for this notification is the missing manager field for the subject of the certification (the manager attribute in LDAP). The email recipient can use this content to fix the failed prepared certification and then restart them. We'll describe how to do this in the next sections.
- Approval Start - this notification will be sent to all Certification Campaign participants with the approver role. The email is sent at the start of campaign and contains the list with certifications to be approved (simple and attributed assignments). This email is sent once; during the approval period, the Approval Remind notification is used.

The content of these default notifications will provide sufficient information for the tutorial emails.

### 2.8.4. Starting the Certification Campaign

Before you start the campaign:

- Perform a manual privilege assignment that can easily be rejected in the campaign: In **Provisioning** → **Users**, assign the role **Test Tasks** to the user **Bellosa Marco** in the folder **Customers** → **Mercato Aurum**.
- Modify the preferred language of **Hungs Olivier** to English: in **Provisioning** → **Users** → **My -Company** → **Hungs Olivier** → **Communication**, set **Preferred Language** from **German** to **English** so that notifications regarding the certification campaign are sent in English. Make sure you record the previous value for this field before you modify it. You will need to reset **Hungs Olivier**'s preferred language to this value at the end of this tutorial to avoid side-effects when running other tutorials.

Because the campaign start date is set in the past, the campaign is ready to start: in the Global View of the Connectivity view group, select the workflow line between the two Identity Store connected directories. Select the **CertificationCampaignController** workflow and then select **Run** from the context menu. The workflow will then start the campaign and create the necessary certification tasks in the campaign folder.

### 2.8.5. Monitoring the Certification Campaign

After the Certification Campaign Controller workflow starts the campaign, the folder for the certification campaign has the following structure:

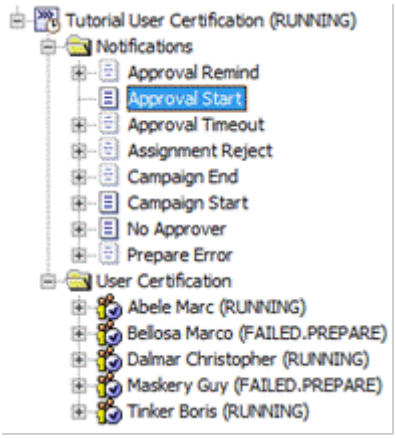



*Figure 8. Tutorial User Certification Folder Structure*

The workflow creates a new folder **User Certification** with a list of certification task entries: one for each user that matches the user filter configured in the campaign entry.

The **Tutorial User Certification** campaign entry has now changed:

In the Status tab, the **State** has been changed to **Campaign is running (RUNNING)**; the campaign has been started and is not yet finished.

In the Logs tab, we see some log messages:

- A warning for user **Bellosa Marco**. The Certification Campaign Controller workflow didn't find an approver for this user and the state of the certification task is set to **FAILED.PREPARE**.

  The default implementation selects the manager of a user as the approver.

- A warning for user **Maskery Guy** for the same reason.
- Because **Status Expiration Date** was not set at the start of campaign, a default Status Expiration Date will be calculated at the end of the campaign and will be **End Date** plus 30 days.

For each user, a certification task is created. By default, the workflow selects the manager of a user as the approver.

- The certification for **Abele Marc** is running correctly with **Dalmar Christopher** as approver, with **DXR Auditors** as Simple Assignments.
- The certification for **Bellosa Marco** is in status **FAILED.PREPARE**. The reason is explained in the Log entry of the Status tab: this user does not have a manager and therefore no approver was found. To fix this problem, click **Edit**. In the Approvers list in the Approvers tab, select **Hungs Olivier**. In the Status tab, change the **Status** to **RETRY.PREPARE** and then click **Save**. Now run the Certification Campaign controller again. Certification changes to **RUNNING** status.
- The certification for **Dalmar Christopher** is running correctly with **Hungs Olivier** as approver.
- The certification for **Maskery Guy** is in status **FAILED.PREPARE**. The reason is explained in the Logs entry of the Status tab: this user does not have any manual assignments. Automatic assignments, such as inherited or obtained by provisioning rules, are not certified.

Because Notifications templates are available for this campaign, the following email notifications are sent:

- A **Campaign Start** notification for the campaign owner; in our tutorial, this is **Hungs Olivier**. This email contains details about the campaign such as: User Filter Base, User Filter, Start Date, Due Date, and so on.
- A **No Approver** notification for campaign owner; in our tutorial, to **Hungs Olivier**. This email informs about users without approvers; in our tutorial, **Bellosa Marco** and **Maskery Guy**.
- Two **Approval Start** notifications for approvers **Hungs Olivier** and **Dalmar Christopher**. These emails contain details about the campaign and about certification tasks subjects (the users to be certified), their Attributed Assignments and Simple Assignments.

The following figure lists these notifications for **Hungs Olivier**:

| Received | Subject | To |
|---|---|---|
| 15.10.2015 16:04 | Certification campaign 'Tutorial User Certification' started | Olivier.Hungs@My-Company.com |
| 15.10.2015 16:04 | No approvers found for certification campaign 'Tutorial User Certification' | Olivier.Hungs@My-Company.com |
| 15.10.2015 16:04 | Please certify IT users in certification campaign 'Tutorial User Certification'. | Olivier.Hungs@My-Company.com |
| 15.10.2015 16:04 | Bitte zertifizieren Sie die IT Benutzer in der Zertifizierungskampagne 'Tutorial User Ce... | Christopher.Dalmar@My-Company |

*Figure 9. Start Campaign Notifications*

Because user **Hungs Olivier** has English as his preferred language (**Provisioning**→ **Users** → **My-Company** → **Hungs Olivier** → **Communication** → **Preferred Language**), notifications will be sent in English. For user **Dalmar Christopher**, the preferred language is set to German.

## 2.8.6. Certifying the Users

Because the campaign is in the state **Campaign is running (RUNNING)**, you can now certify the selected users, which is to accept or reject their manual assignments. You can do this with DirX Identity Web Center or with DirX Identity Business User Interface. In the next sections, we'll demonstrate how to use both options to perform this task. You can choose either option, but don't perform them both.

### 2.8.6.1. Certifying the Users with DirX Identity Web Center

Because the campaign is in the state **Campaign is running (RUNNING)**, you can now open Web Center to certify the selected users, which is to accept or reject their manual assignments.

One approver is **Dalmar Christopher**. Log in to Web Center as **Dalmar Christopher** with the default password. The home page shows the currently running certification campaigns where **Dalmar Christopher** has at least one task.

- Click **Tutorial User Certification**. The list of users that **Dalmar Christopher** needs to certify is displayed.

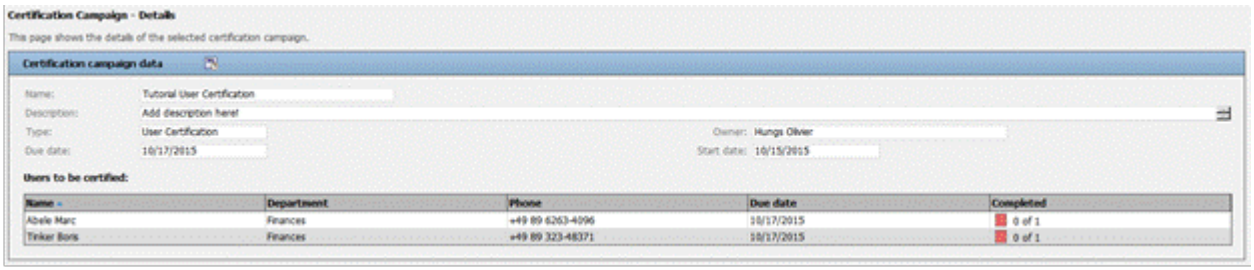



*Figure 10. Dalmar Christopher Certification Campaign Details*

- **Dalmar Christopher** has two certification tasks: one for user **Abele Marc** and one for **Tinker Boris**.
- Click the **Abele Marc** entry. The following figure shows the result:

*Figure 11. Abele Marc Certification Details*

- Check **Accept** and then click **Save changes and finish certification**.
- For user **Tinker Boris**, apply the same actions: accept the current role as **DXR Audit Administrator**.
- Select **Accept** and then click **Save changes and finish certification**.

Now log out from Web Center and then log in as **Hungs Olivier** with the default password.

- Follow the same steps as you did for **Dalmar Christopher**. **Hungs Olivier** has two users to certify: **Bellosa Marco** and **Dalmar Christopher**.
- Ignore all the privileges for **Dalmar Christopher** and leave them uncertified.
- For **Bellosa Marco**, reject the **Test Tasks** privilege and then click **Save changes and finish certification**.

*Figure 12. Bellosa Marco Certification Details*

In DirX Identity Manager, check the campaign **Tutorial User Certification**:

- **Tutorial User Certification** is in the state **Campaign is running (RUNNING)**.
- All approved certifications are now in the state **APPROVAL.FINISHED**.
- The **Maskery Guy** certification is still in the state **FAILED.PREPARE** because there were no manual assignments to approve, and no approver assigned.
- The **Dalmar Christopher** certification is in the state **RUNNING** because no approver performed any action for certifying this user.

#### 2.8.6.2. Certifying the Users with DirX Identity Business User Interface

Because the campaign is in the state **Campaign is running (RUNNING)**, you can now open Business User Interface to certify selected users, which is to accept or reject their manual assignments.

One approver is Dalmar Christopher. Log in to the Business User Interface as **Dalmar Christopher** with the default password. The home page shows the current running certification campaigns where Dalmar Christopher has at least one task.

- Click the My Certification widget.
- Select **Tutorial User Certification** from the certification campaign lists. The list of users that Dalmar Christopher needs to certify is displayed.
- Dalmar Christopher has two certification tasks: one for user **Abele Marc** and one for **Tinker Boris**.

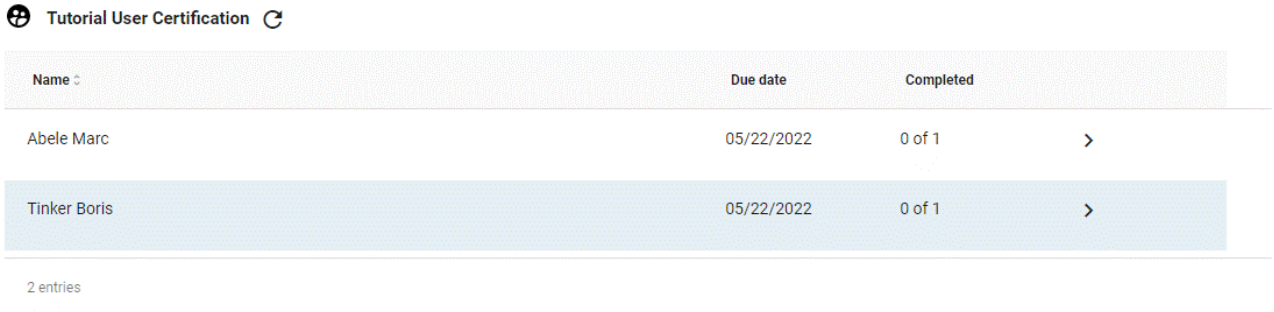



*Figure 13. Dalmar Christopher Certification Campaing Details*

- Click the **Abele Marc** entry. The following figure shows the result:

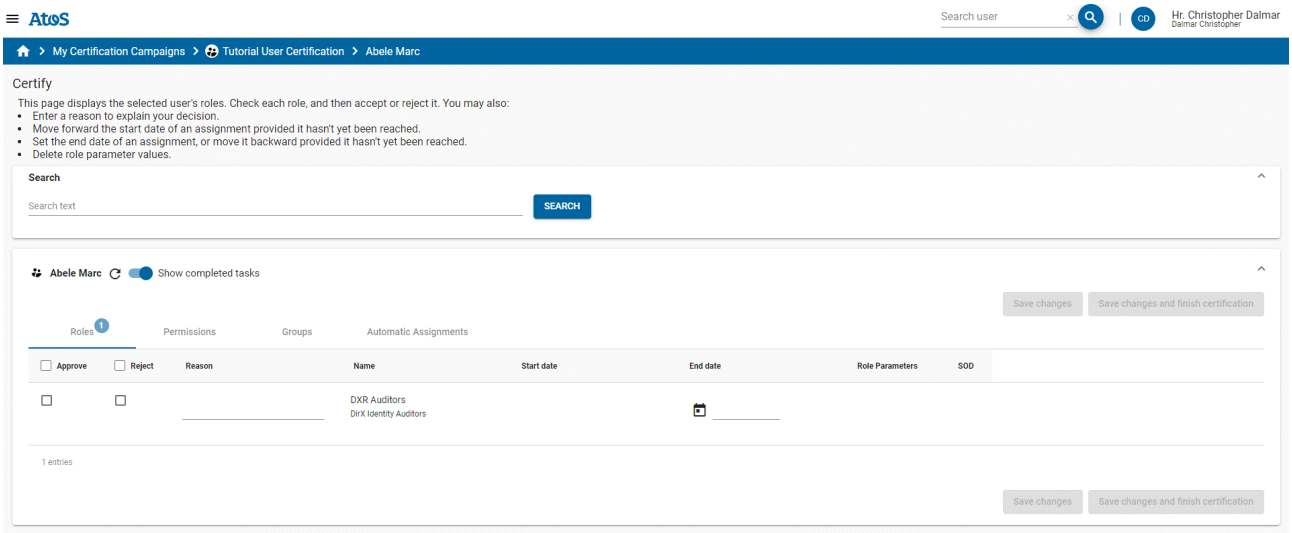



*Figure 14. Abele Marc Certification Details*

- Check **Approve** and then **click Save** changes and finish certification.
- For user **Tinker Boris**, apply the same actions: approve the current role as DXR Audit Administrator.
- Check **Approve** and then click **Save changes** and finish certification.
- Now log out from Business User Interface and log in as **Hungs Olivier** with the default password.
- Follow the same steps as you did for **Dalmar Christopher**. **Hungs Olivier** has two users to certify: **Bellosa Marco** and **Dalmar Christopher**.
- Ignore all the privileges for **Dalmar Christopher** and leave them uncertified.
- For **Bellosa Marco**, reject the **Test Tasks** privilege and then click **Save changes** and finish certification.

In DirX Identity Manager, check the campaign **Tutorial User Certification**:

- **Tutorial User Certification** is in the state Campaign is running (**RUNNING**).
- All approved certifications are now in the state **APPROVAL.FINISHED**.
- The **Maskery Guy** certification is still in the state **FAILED.PREPARE** because there no manual assignments to approve, and no approver assigned.
- The **Dalmar Christopher** certification is in the state **RUNNING** because no approver performed any action for certifying this user.

### 2.8.7. Finishing the Campaign

Now you can finish the campaign by changing the Due Date in the campaign entry (we do this because we want to finish the tutorial on the same day):

- Go to the campaign entry **Tutorial User Certification** and select the Status tab.
- Click **Edit** and change **Due Date** to today.
- Click **Save** to commit the changes.
- Run the **CertificationCampaignController** workflow from the **Connectivity** view group.

After the workflow has finished, check the campaign status. All the certification tasks should be in the state **FINISHED** except for the **FAILED** one and the certification campaign should be in the state **Campaign finished successfully (SUCCEEDED)**.

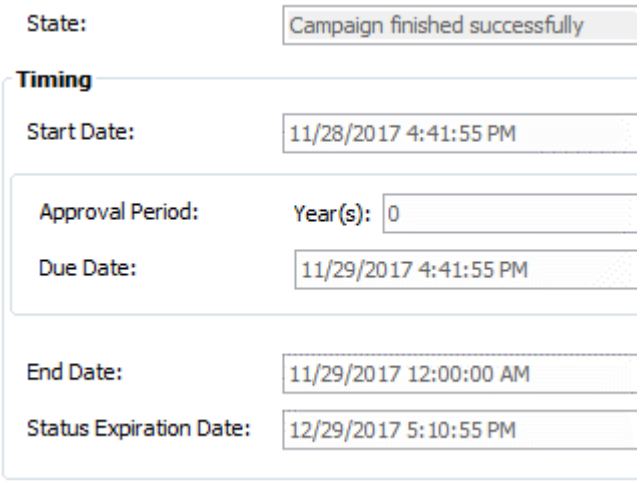



*Figure 15. Finished Campaign Details*

- In the **Provisioning** view group, select the Certification Campaigns view and then select the campaign **Tutorial User Certification**.

The **End Date** is set to the current date.

Because we did not explicitly set the **Status Expiration Date** field during the campaign, the value was set by the Certification Campaign controller as **End Date** plus 30 days. After the campaign status expires, the campaign will be automatically moved to the **Campaign is marked for deletion (DELETED)** state and deleted later on by the Cleanup Objects

workflow.

- Underneath the campaign container, browse to the certification for **Bellosa Marco**. The **Approvers** tab shows **Hungs Olivier**.
- The **Attributed Assignments** tab is empty. **Bellosa Marco** had no assignments with either role parameters or end date.
- The **Simple Assignments** tab shows that the **Test Tasks** role has been rejected.
- Navigate to the user **Bellosa Marco** in the folder **Customers** → **Mercato Aurum** and check that the role **Test Tasks** has been removed.
- Return to the **Tutorial User Certification** campaign and select the certification for **Dalmar Christopher**.
- All assignments in the tabs Attributed Assignments and Simple Assignments are in the **Not Certified** list. If you check the user entry in the folder **My-Company** → **Finances**, you'll see that all these assignments still exist. This is correct because the **Revoke privileges settings** field in the campaign requested to remove only the rejected assignments and leave the assignments that were not certified as they are.

### 2.8.8. Generating a Certification Campaign Report

You can produce a report at any time during a running campaign:

- In the Provisioning view group, select the Certification Campaigns view and then browse to **Tutorial User Certification**.
- Right click the campaign entry and then select **Report** from the context menu.
- Select the report **Campaign with all properties**.
- Set the following fields:

In **Search base**, select **Tutorial User Certification**.

In **Type**, select **HTML**.

Uncheck **Output to viewer**.

In **Output file**, select a location in which to save the report.

- Click **Run report**.
- Browse to the location where the report was saved and open the generated HTML report with your favorite web browser. You will see something like this:

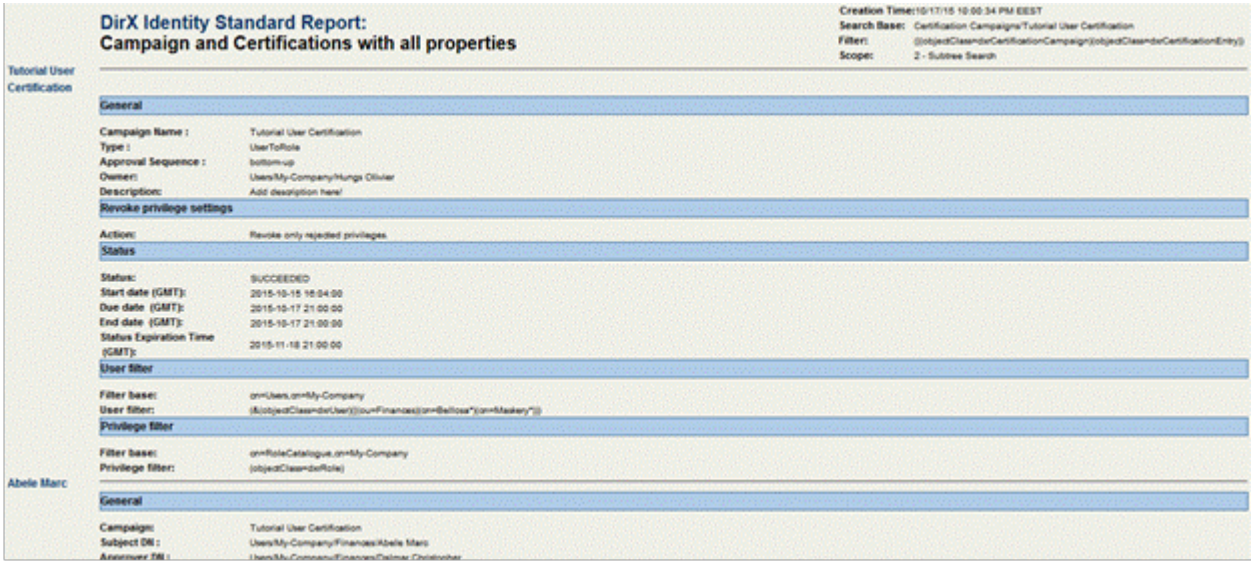



*Figure 16. Certification Campaign Report*

### 2.8.9. Deleting a Certification Campaign

If you do not want to wait 30 days until the campaign has expired, you can change the **Status Expiration Date** and let the standard workflow delete the campaign tree:

- In the Provisioning view group, select the Certification Campaigns view and then browse to **Tutorial User Certification**.
- Select the Status tab and then set **Status Expiration Date** to a date in the past (for example, **yesterday**).
- In the Connectivity view group, navigate to the workflow **CertificationCampaignController** (**Workflows** → **My-Company** → **Main** → **Identity Store**) and then run it via the context menu.
- Return to the campaign in the Provisioning view group.You'll see that the campaign **Tutorial User Certification** is now in the state **Campaign is marked for deletion (DELETED)**.This state is handled by the DirX Identity cleanupObjects workflow, which also evaluates the **cleanupDeletedCertificationCampaigns** rule from **Provisioning** → **Policies** → **Rules** → **Default**→ **Consistency**.

You have now completed the certification campaign tutorial.The last step is to reset the preferred language for user **Hungs Olivier** to the value you recorded before you changed it to English to avoid side-effects when running other tutorials.See the section "Starting the Certification Campaign" for instructions.

## 2.9. Applying Attribute Modification Approval

In this section, we'll demonstrate how attribute modification approval works.We'll show you how to:

- Activate attribute modification approval
- Modify and activate a user attribute modification policy

- Modify user attributes that require approval and examine the effects
- Approve the attribute modification request and check the result

### 2.9.1. Activating Attribute Modification Approval Checking

First, we'll activate attribute modification approval checking, if necessary. For performance reasons, this feature is deactivated by default in all customer domains. We'll activate it using DirX Identity Manager's **Provisioning** → **Domain Configuration** view:

- Log in to DirX Identity Manager's **Provisioning** view group.
- Click **Domain Configuration** → **My-Company**.
- In the **RequestWorkflows** tab, click **Edit**.
- Check **Attribute modification approval**, if it's not already checked.
- Click **Save**.

To make this change effective within DirX Identity Manager itself, you must stop and then restart all of your running Manager instances:

- Click **File** → **Exit** to close your Manager instance(s).
- Start Manager again and log in to the **Provisioning** view.

### 2.9.2. Modifying an Attribute Modification Approval Policy

Next, we'll configure an existing attribute modification approval policy provided with DirX Identity in the sample domain.

The My-Company sample domain has a pre-configured attribute modification approval policy for user attributes that requires approval on changes to the attributes Location and Organizational Unit. We'll configure and activate this policy now with DirX Identity Manager if necessary:

- In **Provisioning** → **Policies**, click **Attribute Policies** → **My-Company** → **User - Location and Organization**.
- In the **General** tab, click **Edit**, and then check **Is active** if it's not already checked.
- Click **Save**.
- Click the **Configuration** tab. You can see the attributes that require approval listed in the **Selected** box. The **Available** box lists additional attributes you can select to require approval.

### 2.9.3. Modifying the User Attributes

Now we're ready to test the user attribute modification policy. We'll use Franca Baretti in Marketing as our example, and use Web Center to change her Locality and Country attributes. We'll log in as Nik Taspatch, since he is the user administrator for the Marketing Department:

- In an Internet browser, start DirX Identity Web Center.

- Log in as **Taspatch Nik** with the password.
- In **Users** → **Select user**, enter **B** in **Search for**, and then click **Search** to return a list of users whose names begin with "B".
- Select **Baretti Franca** from the list. Web Center displays a user summary for Franca.
- In the **Users** menu, click **Modify user data**.
- Change the location to **My-Company San Jose** in **Location**.
- Click **Save**. Web Center displays Franca Baretti's entry again.

### 2.9.4. Monitoring the Attribute Approval Process

We can use the Web Center to check the results of Nik Taspatch's changes to Franca's user attributes.

- First note that the location in Franca Baretti's summary page is still **My-Company Rome**. But the page now includes a modification order list which shows that the modification is in approval (Since it may take some time to start the approval workflow, you may have to click the **Refresh summary** icon in the page repeatedly before you see the item.)
- Now click **Show subscription status** in the **Users** menu.
- The **Select a Workflow** page lists a workflow for Franca Baretti with the status **Running**. Click it.
- Franca Baretti's attribute modifications require approval from a member of My-Company's Human Resources department. In **Running activities**, you can see the members of this department. Only one of these members needs to approve the request.
- There is nothing more for Nik Taspatch to do, so click **Logout**.

We can also use DirX Identity Manager's Monitor View to see the progress of Franca Baretti's attribute modification approval:

- Log in to DirX Identity Manager's **Provisioning** view.
- Click **Workflow**, and then click **Workflow** → **Monitor** → **My-Company** → **Users** → **Modify Location and Organization**. In a production environment, this folder contains sub-folders named with dates; for example, **2016-10-04**. Each sub-folder contains status information about the request workflows that have executed on these dates.
- In our case, we have one folder with today's date. Click this folder, and then look for **Baretti Franca** in the list of workflows (it should be the only one there). Click it. In the General tab, Manager displays a graphical representation of the request workflow that is processing Franca's attribute modification approval. The step **Approval of Attribute Modifications** is highlighted in grey to indicate that it is the activity that is currently being processed. You can double-click this step, and then click the Status Information tab. You can see the same participant list of Human Resources members who are allowed to approve the request.

### 2.9.5. Approving the Attribute Modification Request

As mentioned before, Franca Baretti's attribute modifications require approval from a member of My-Company's Human Resources department. The request workflow's approval activity will automatically notify each of these people by email about Franca Baretti's attribute modification request. Only one of these people needs to approve the request in order for it to be successfully processed. In this topic, we'll show how one of these people - Hans Berner - uses the Web Center to view and approve Nik Taspatch's request. We can do this with or without email notification; it depends on how you set up email notification when you prepared to use the quick start. For this exercise, we'll assume that email notification is off (the default). See the first exercise in this tutorial ("User Self-Registration") for more information about this feature.

To approve the request from Hans Berner's **Task list** dialog:

- Start the Web Center.
- Enter **Berner Hans** in **Name** and enter the password. (Remember that all persons in the sample domain are set up to have the same password.)
- Click **Log in** (or press RETURN) to log in.
- In **Work List**, click **Task list** and select the approval task. Web Center displays Franca Baretti's data in the "Approval of Attribute Modification" dialog.
- Enter **F. Baretti transfer to San Jose, U.S. of America** in **Reason**. (It's good practice to provide a reason for your decision, especially if you reject a request.)
- Click **Accept** to grant the request.

**Task list** is now empty.

- There is nothing more to do, so click **Logout**.

### 2.9.6. Checking the Result of Attribute Modification Approval

Next, we use the Web Center to check the results of the approval on Franca Baretti's user data:

- Enter **Taspatch Nik** in **Name** and enter the password. Click **Log in** or press RETURN.
- In **Users** → **Select user**, enter **B** in **Search for**, and then click **Search** to return a list of users whose names begin with "B".
- Select **Baretti Franca** from the list. The **Display summary** page is shown. Check that the location has now changed to **My-Company San Jose**.

Country and postal address have been changed accordingly since these attributes are mastered by location.

- Select the **Groups** button near the bottom of the page.
- Check that the Windows groups and accounts have moved from **Windows Domain Europe** to **Windows Domain US** (the assignments still exist, but are in the state

DELETED or DISABLED).This shows that DirX Identity automatically re-configures the IT resources that Franca Baretti requires.

- Click the **Users** → **Show subscription** status icon in the toolbar and then click the button **Succeeded Workflows**.The **Select a Workflow** list shows the workflow for Franca Baretti with the status **Succeeded**.Click the workflow.The **Finished activities** section shows the results of the approval.
- There is nothing more to do, so click **Logout**.

You can also check the results with DirX Identity Manager:

- In the Provisioning view group, click **Workflows**, and then click **Workflows** → **Monitor** → **My-Company** → **Users** → **Modify Location and Organization** → *today's date* → **Baretti Franca**.
- Double-click the **Approval of Attribute Modifications-0** step, and then click the Status Information tab.Here you can see that the approval request workflows for the other Human Resources members were successfully finished, since one member accepted the attribute modification.

# 2.10. Scheduled Privilege Assignment

Event-based privilege assignment changes access rights almost in real time and reduces time-consuming policy executions and privilege resolutions.Nevertheless, there may be reasons when these concepts are not sufficient; for example:

- You want to run a cleanup policy execution and privilege resolution over all users from time to time to be sure that all users have the correct access rights.
- You allow pure LDAP clients to change user entries.In this case, the service layer cannot produce events.The only way to solve this problem is to indicate the changed users by setting the dxrTBA flag (**To Be Analyzed** in the user interface) to TRUE and then subsequently run a scheduled or manually-triggered policy execution and privilege resolution.

In the next sections, we demonstrate the second scenario.

## 2.10.1. Disabling Event-based Privilege Resolution

Before we proceed with this exercise, we need to stop the Java-based server to disable event-based privilege resolution, which runs as a part of the event-based maintenance workflows. See Managing the Java-based Server.

We must also disable the SoD flag at the domain object, otherwise privilege resolution will run automatically with policy execution:

- Log in to DirX Identity Manager's Provisioning view group.
- Click **Domain Configuration** → **My-Company**.
- In the Compliance tab, click **Edit**.
- Uncheck **Segregation of Duties** (**SoD**) **checks**.

- Click **Save**.

### 2.10.2. Performing a Pure LDAP Change

Use either a pure LDAP client or DirX Identity Manager's Data View:

- Click **Provisioning**.
- Select the user **cn=Briner Ruben,ou=Sales,o=My-Company,cn=Users,cn=My-Company**.
- Click **Edit**.
- Scroll down to the **employeeType** attribute and change its value to **Contractor**.
- Set the **dxrTBA** attribute to **true**. You can use this flag later on to indicate to DirX Identity services (policy execution and privilege resolution) that only users with this flag need to be managed (after working on these objects, the services reset the flag). If you do not use this flag, you must run the services on the entire user population, which might be a time-consuming procedure with a high load on the Provisioning configuration server.
- Save the user object.

Check the object from the Provisioning view group:

- Select the user **Briner Ruben** from **Provisioning** → **Users** → **My-Company** → **Sales**.
- Click the **General** tab and then refresh the entry. The **Employee Type** field displays **Contractor**.
- Verify that the **To be analyzed** flag in the **Operational** tab is set (this is the **dxrTBA** flag).
- Check that the user has the **Internal Employee** role assigned; this is inconsistent with the changed employee type. To correct this problem, we run the policy execution service and a subsequent privilege resolution.

### 2.10.3. Configuring the Policy Execution Service

First, we need to configure the policy execution service with DirX Identity Manager:

- Click **Provisioning** and then click **Target Systems**.
- Right-click **Target Systems** (the top-level object in the tree) and then select **Connectivity** → **Workflows** → **PolicyExecution** → **Configure Workflow**. The workflow wizard opens.
- Click **Next**. You can see that the policy execution service does not perform a privilege resolution automatically (**Provisioning Mode** is set to **Assign Privilege only**). Do not change this step. Click **Next** again.
- Change **Base Object** to **cn=Role based scenario,cn=My-Company,cn=Rules,cn=Policies,cn=My-Company** if it is not already set to this value. This setting indicates that we run all policies below this node.
- Change **Search Filter** to **(objectClass=dxrProvisionRule)** if it is not already set to this value to specify that we use only provisioning rules in this run.
- Click **Finish** to close the wizard.

### 2.10.4. Running the Policy Execution Service

To run the policy execution service:

- Right-click **Target Systems** and select **Connectivity** → **Workflows** → **PolicyExecution** → **Run Workflow** to run the workflow. Wait until the workflow has completed.

### 2.10.5. Using the Structure Tab to Check the Results

First we'll use the policy execution workflow's open window to check the results:

- Click the **Structure** tab and then the activity in the middle. A new window opens. Click the **Trace** tab and then check the trace file (click the line and then the button to the right).
- At the tracing level set for the workflow, the trace file contains only brief statistics. Each privilege that has been processed is listed in a table with the number of subjects processed and the corresponding errors and warnings. In this case, the privileges **Contractors**, **Internal Employee** and **Signature Level 1** each have processed one subject (user). This must be our Ruben Briner, because all other users in the database were not changed, and so the policy execution service had nothing else to do.

Assignment of permission **Signature Level 1** failed since we've stopped the Java-based server. Let's ignore this for this tutorial chapter.

- Close the trace window and the Policy Execution and Run Workflow windows.

### 2.10.6. Checking Ruben Briner's Privileges

Use the DirX Identity Manager to check the result of the policy execution workflow run.

- Select the user **Briner Ruben** from **Provisioning** → **Users** → **My-Company** → **Sales**.
- Click the **Assigned Roles** tab and refresh the entry. The **Contractor** role should be visible.
- Click the **Assigned Permissions** tab and refresh the entry. The **Contractor** permission is not yet visible because the privilege resolution service did not run. For the same reason, the **Internal Employee** permission is still assigned.

### 2.10.7. Running the Privilege Resolution Service

Now we'll resolve the privileges that the policy execution service assigned to Ruben Briner:

- Select the **Target Systems** view.
- Right-click the **Target Systems** top-level node and select **Connectivity** → **Workflows** → **PrivilegeResolution** → **Run Workflow** to run the privilege resolution workflow. Wait until the workflow has completed.
- Click the **Structure** tab to check whether the workflow has performed successfully. The workflow activity **PrivilegeResolution** must be displayed in green.

- Close the Run Workflow window.

### 2.10.8. Re-Checking Ruben Briner's Privileges

Now we return to the DirX Identity Manager's **Provisioning** → **Users** view and examine Ruben Briner's privileges again.In DirX Identity Manager → **Provisioning** → **Users** → **My-Company** → **Sales**:

- Click the **Assigned Permissions** tab. Now the **Contractor** permission is visible (you may need to click the refresh button ↻ first).
- Click the **Assigned Groups** tab. Several new group assignments have been created.

A new **Contractor** group is visible (**Windows Domain USA**) as well as the **Contractor Portal** group in the **Intranet Portal** target system.

This exercise shows how you can resolve inconsistencies in the Identity Store.

## 2.11. Creating a Nested Workflow

In this exercise, we show you how to create a nested Tcl-based workflow.Before you try this exercise, make sure you've followed the exercise described in "Importing Identities" in this Guide.

We'll combine the run of the NewHR2Ident workflow, the policy execution service and the privilege resolution service in a nested workflow.To do this, we'll use the DirX Identity Manager's Expert View to copy a default workflow and move it to our NewCompany scenario and then reconfigure its activities from the defaults to the values for our scenario.Next, we'll run and monitor the workflow from the Expert View, then make it available for execution in the Global View.

### 2.11.1. Copying and Moving a Default Workflow

First, we'll use DirX Identity Manager's Expert View to copy a default workflow and move it to our NewCompany scenario:

- Click **Connectivity**, and then click **Expert View**.
- Open **Workflows** → **Default** → **Source Scheduled**. The workflow **ODBC2Ident+Maintenance** is exactly what we need.
- Select this object and then select **Copy Object**. Set the name to **NewHR2Ident+Maintenance** and click **OK**.
- The location of the copied object is not correct. Select **Move Object** and set **Workflows** → **My-Company** → **NewCompany** → **Source Scheduled** → **ODBC** as the target. Click **OK** and wait until the move is performed. DirX Identity checks and adapts all references pointing to the moved object to guarantee the integrity of the Connectivity configuration, so this action requires some time.

### 2.11.2. Reconfiguring the Default Workflow's Activities

Next, we'll reconfigure the activities in the default workflow to point to our scenario's objects:

- Open the previously selected target path for the move operation. The new workflow **NewHR2Ident+Maintenance** is visible there.
- Click it and open it. Three activities are visible but they point to the wrong (default) objects. Adjust them to make the workflow usable for the My-Company scenario.
- Click the **Creation Workflow** activity in the left pane and then **Edit**. Click the last icon behind **Run Object (...)**. A tree browser opens and you can see that this activity references the default **ODBC2Ident** workflow. Select **Workflows** → **My-Company** → **NewCompany** → **Source Scheduled** → **ODBC** → **NewHR2Ident** instead and click **Save**.
- Click the **Policy Execution** activity and link it to **Jobs** → **My-Company** → **Main** → **Identity Store** → **Policy Execution**. Click **Save**.
- Click the **Privilege Resolution** activity and link it to **Jobs** → **My-Company** → **Main** → **Identity Store** → **Privilege Resolution**. Click **Save**.

### 2.11.3. Running and Monitoring the Nested Workflow

Now we have a workflow that runs the ODBC creation workflow and then the policy execution and privilege resolution services. Run the new workflow from the Expert View and click the **Structure** tab in the run window. Watch the three activities running in sequence.

### 2.11.4. Making the Nested Workflow Available in the Global View

The last step is to make this workflow available in the Global View:

- Click **Global View** and select the scenario **My-Company** → **NewCompany**.
- Right-click the workflow line, then select **Assign**.
- Select the **NewHR2Ident+Maintenance** workflow from the list and click **OK**.

Now you can run the workflow from the Global View.

If you need a new nested workflow for another identity creation workflow, simply copy this workflow and link only the first activity to the new creation workflow.It makes sense to use the same second and third activity; for example, if you need to reconfigure one of these activities, the new configuration will take effect in all nested workflows at once.

## 2.12. Using Manual Provisioning

There are many reasons why it is difficult or why it does not make sense to tightly integrate a potential connected system.Setting up the system for manual provisioning provides you with a quick and easy solution.In this section, we'll show you how to do it with the example of a physical access system that is part of the DirX Identity sample domain.We'll describe how manual provisioning works and demonstrate how to:

- View the scenario in the sample domain
- Request physical access to secured rooms
- Check the approval result
- Perform manual provisioning as an administrator
- View the result

You can read more about this topic in Service Management.

## 2.12.1. Understanding Manual Provisioning

The My-Company sample domain contains the Physical Access target system, which is an example of an offline target system. With an offline system, there is no direct connection for provisioning. Instead, the target system administrator is notified by request workflow activities that he needs to synchronize the external offline system with the target system's content.

Manual provisioning requires configuration in both the Connectivity and Provisioning views and consists of the following steps:

- Creating a new target system of type **RequestWorkflow** in the **Target Systems** view that reflects your offline system that is to be manually provisioned. This step adds the necessary Java-based workflow that starts the corresponding request workflow for each change to the target system.
- Setting up the necessary groups in your target system or loading them from the offline system.
- Copying the provided request workflow template, modifying it to your requirements and then configuring the Java-based workflow you copied to use this request workflow definition.

These steps have been completed for the **Physical Access** target system. You can use this prepared example to run this exercise.

## 2.12.2. Viewing the Scenario

In our scenario, the Physical Access target system controls access to protected rooms in the following locations:

- A documentation archive in Munich
- Three data center rooms in Berlin, Frankfurt and Munich.

Roles and rules protect the access to these rooms.

### 2.12.2.1. Viewing the Target System

First we view the target system Physical Access.

- Log in to the **Provisioning** view group of the DirX Identity Manager.

- Select the **Target Systems** view and navigate to the **Physical Access** target system.
- In the **General** tab, you can see that the target system type is **RequestWorkflow**. This assignment means that it is configured and tailored to be used with real-time workflows that start request workflows. These request workflows can be used for manual provisioning. To create another instance of this type of target system, select the type **Service Management** in the target system wizard.
- Click the **Advanced** tab. This target system works with assignment states and references groups from the account side. This is important because then all changes for attributes and group memberships are stored at the account. Real-time provisioning is enabled and password synchronization is disabled. Synchronizing passwords does not make sense because the administrator would potentially know all passwords of all users.
- Open the tree. You can see a folder with a set of accounts and a folder with four groups: **Munich - Archive** protects the documentation archive and the three **Data Center** groups protect the data centers in Berlin, Frankfurt and Munich.

All these groups are not user assignable.

#### 2.12.2.2. Viewing the Privilege Structure

Next, we evaluate the privilege structure.

- Select the **Privileges** view.
- Navigate to **Roles** → **Corporate Roles** → **Physical Access**. This folder contains four roles that are each linked to the corresponding permission (view the **Assigned Permissions** tab). These roles are all user assignable.
- Navigate to **Permissions** → **Corporate Permissions** → **Physical Access**. This folder contains four permissions that are linked to the four groups in the target system (view the **Assigned Groups** tab).

The permissions are not user assignable.

The role administrator has decided that only roles can be assigned to users. Assignment of permissions and groups is not possible.

#### 2.12.2.3. Viewing the Rules

The sample scenario assumes that users located in Berlin, Frankfurt and Munich get automatic access to the data center rooms. Access to the archive requires explicit role assignment as well as access to the data center rooms for persons outside the three mentioned locations.

- Select the **Policies** view.
- Navigate to **Rules** → **My-Company** → **Role based scenario** → **Corporate**.
- View the three rules for access to **Data Center**. They assign all internal employees of the specific location to the corresponding group.

These rules are either executed during a policy execution service run or during an event-based user processing workflow.

## 2.12.3. Requesting Physical Access

In our example scenario for manual provisioning, Bill Sedran, the head of software development in San Jose, frequently visits the My-Company's Munich branch and needs access to the Munich Archive when he's there. He requests the necessary role via Web Center.

### 2.12.3.1. Requesting Access to the Munich Archive

To launch the access request for the Munich Archive:

- Log in as **Sedran Bill** into Web Center.
- Select **Self Service** → **Subscribe privileges** and click **Search**.
- Click the **Munich** - **Archive** role, move it to **Assigned roles** and then click **Save**.
- Select **Self Service** → **Show Subscription Status**. A new line with the pending approval is shown.
- Click the line to open the details. You can see that Olivier Hungs, Gabriela Morton (the role's owners) or Christopher Dalmar (Olivier's representative) are required to approve the assignment in **Approval by Privilege Managers** activity. The **Approval by User Manager** activity must be approved by Frederic Duplan (Bill's manager) or by Veronique Cohu (Frederic's representative).
- Log out from Web Center.

### 2.12.3.2. Approving the Access Request

All of the approvers have received an email message that requests that they approve the role assignment.

- Log in to Web Center as **Hungs Olivier**.
- Select **Work List** → **Task list** and approve the subscription with the reason "As discussed with Dalmar Christopher".
- Log out from Web Center.
- Log in as **Duplan Frederic** into Web Center.
- Select **Work List** → **Task list** and approve the subscription with the reason "Necessary according to Bill Sedran's work profile".
- Log out from Web Center.

## 2.12.4. Checking the Approval Result

Now that approval is complete, we can check the result in various places.

#### 2.12.4.1. Checking the Request Workflow Result

First, we'll check the approval result in the **Workflows** view:

- Open DirX Identity Manager (**Provisioning** view group).
- Click the **Workflows** view.
- Navigate to **Monitor** → **My-Company** → **Approval** → **4-Eye Approval** → *date* → **Sedran Bill** → **Munich - Archive**.
- Click the **General** tab. In the structure view, you can see that both approvers have accepted and the Apply Approved Privilege Change activity has succeeded. In this step, privilege resolution was performed that created a new account for Bill Sedran and that was assigned to the Munich - Archive group in the Physical Access target system.

#### 2.12.4.2. Viewing the New Account and Group Membership

Now let's view the new account and group membership information in the Physical Access target system:

- Click the **Target Systems** view.
- Navigate to **Physical Access** → **accounts** and open this folder.
- Click the **Bill Sedran** account.
- View the **Operational** tab. You can see that the account is in state ENABLED but the target system state is still in state NONE which means that it is not yet created in the connected system.
- Click the **Member of** tab. The assignment state for the Munich - Archive group is ADD which also means that the assignment is not yet provisioned.

#### 2.12.4.3. Viewing the Real Time Workflow

Next, we'll check the real time workflow that results from the approvals:

- Open DirX Identity Manager (**Connectivity** view group).
- Click the **Monitor View**.
- Navigate to **My-Company** → **Main** → **Target Realtime** → **Physical Access** → **Ident_PhysicalAccess_Realtime**. You can see one instance of the workflow.
- Click the instance and view the **Remark** field. There could be a message that indicates that a request workflow was started to request manual provisioning:
  Creation request workflow for object "cn=Bill Sedran 4365,cn=accounts,cn=Physical Access,cn=TargetSystems,cn=My-Company" successfully instantiated.

#### 2.12.4.4. Viewing the Request Workflow Instance

Now let's check the request workflow instance (created by the previous real time workflow):

- Open DirX Identity Manager (**Provisioning** view group).
- Click the **Workflows** view.

- Navigate to **Monitor** → **My-Company** → **Service Management** → **Physical Access** → *date*. One request workflow instance is present.
- Click the instance. You can see that one instance waits in the activity **Addition by Administrator**. Double-click the activity and view the approver in the **Status Information** tab. **Retha Wagner** - who is the **Local Admin** of the **Physical Access** target system - is responsible for acting on this workflow.
- Click the **Object** tab of the workflow. You can see all attributes of the new account (this is technically the order information). We will see that this information is sufficient for the administrator to create the account in the Physical Access offline system.

The order information in the **Object** and **Assignment** tabs is used to create the necessary information in the corresponding Web Center pages.

### 2.12.5. Performing Manual Provisioning

In our sample scenario, we assume that an administrator of an offline target system does not receive an email message when a new provisioning request is produced. As a result, the approval activities of the Physical Access request workflow are not configured for email. Look-up the definition of the workflow in the path **Definitions** → **My-Company** → **Service Management** → **Physical Access**. For our scenario, we assume that Retha Wagner works frequently in her work list.

- Log in to Web Center as **Wagner Retha**.
- Execute **Work List** → **Task list** from the menu bar.
- You can see a work item that requests the account creation.
- Click the item to view the details (**Addition by Administrator**). It shows that an account for Bill Sedran is to be created and added to the Munich - Archive group. All of the required attributes are shown.

These attributes are defined in the mapping of the corresponding real-time workflow. So you can add or remove attributes as they are needed. You can also define the sequence of the attributes in the mapping definition.

- Retha now must open the administrative interface to the Physical Access system and add the account as requested. After successful creation, she can add a reason and then click **Accept**. If she wants to do the task later, she can press **Do later**. Clicking **Reject** means that she cannot do the task. In this case, it makes sense to add a reason. This action only sends a notification to the user that the task failed. The DirX Identity target system and the external system are not in sync which is visible because the target system state is still NONE.
- We assume that Retha can add the account to the Physical Access system, so we acknowledge it by clicking **Accept**.
- The work list is now empty.

The manual provisioning scenario assumes that the administrator works carefully and correctly. Nevertheless, errors are possible. To solve this problem, build a mechanism such

as a file-based workflow that validates the content of the external system from time to time against the target system. Then the differences are visible and the administrator can address them.

## 2.12.6. Viewing the Result

In the previous sections, we learned how manual provisioning is to be done. This section completes the topic by viewing the result in various ways.

### 2.12.6.1. Viewing the Provisioned User

First, let's view Bill Sedran's account and group membership from his perspective:

- Log in to Web Center as **Sedran Bill**.
- Perform **Self Service → Display summary** from the menu bar.
- Click the **Accounts** button. The target system state of the account in the Physical Access target system is ENABLED.
- Click the image at the end of the account line to display the assigned groups. Verify that the assignment state is ENABLED.

### 2.12.6.2. Viewing as the Administrator

Next, we'll view the account and group membership from the target system administrator's perspective:

- Log in to Web Center as **Wagner Retha**.
- Perform **Accounts → Select account** from the menu bar.
- Type in "Bill" into the search field (begins with). Click **Search**.
- In the displayed account table, click the account for the **Physical Access** system. All attributes are displayed and also the group membership.
- Click the button at the end of the group line to display the group.
- Select **Groups → Show members** to display the group members.

### 2.12.6.3. Checking Statistics on Groups

Finally, we'll run a report on group statistics to check the effect:

- In Web Center, select **Tools → Reports**.
- Navigate in the tree to **Target Systems → Physical Access** and then click it.A set of possible reports is displayed.
- Click **Number of users per group**.After a few seconds, the list of groups is displayed.The second column displays all users that are directly assigned to the groups via the provisioning rules.The first column also shows the indirectly assigned users, which are users assigned via roles or permissions.There's only a difference for the **Munich - Archive** group.

# 2.13. Working with Internal Tickets

In most cases, you want changes in your identity solution to be effective immediately in real time.However, there are situations where it makes sense to delay changes to a later date.If you have no way of delaying these changes, you have to note the changes somewhere, wait until the time comes and then perform the changes.

Of course this procedure is very inconvenient.You can use DirX Identity's internal service management component to manage these situations.

In this section, we'll show you how to create and manage tickets in DirX Identity.First, we explain how you can work with internal tickets and then we demonstrate how to:

- Create tickets
- View tickets
- Process tickets
- View the processed ticket and its results

For more information about this topic, read Managing Tickets.

## 2.13.1. Understanding How to Work with Internal Tickets

Creating tickets in DirX Identity is easy. Perform your action as usual and then set a due date with Web Center or DirX Identity Manager. The action is not performed immediately, but at the required due date. DirX Identity supports the following types of actions:

- Creating objects
- Modifying objects
- Deleting objects
- Assigning privileges
- Removal of privileges

A ticket processor needs to run daily to process all maturing tickets. Depending on the action to be processed, corresponding approval workflows and their results must be handled.

In this follow-on tutorial, we modify a user while setting a due date, process the ticket and view the results.

## 2.13.2. Creating a Modification Ticket

We assume that Leo Kubalke from the Product Testing group is to receive his doctorate on the date that he defends his thesis. After that date, he'd like to add his new title to his company name information.

For purposes of this exercise, we need to set today's date as the due date, or the ticket processor won't run and we can't perform the whole tutorial on one day.

We use the DirX Identity Manager for this tutorial.

To prepare for the upcoming name change:

- Login to the **Provisioning** view group of the DirX Identity Manager.
- Click the **Users** view and then navigate to **Users** → **My-Company** → **Product Testing**.
- Select **Leo Kubalke** and then click the **General** tab.
- Click **Edit** and enter **Dr**. into the **Title** field.
- Enter today's date into the **Due Date** field (to the left of the **Save** button).
- Click **Save**.

After a short time, the **Title** field is cleared, which means that the change is not yet effective.

### 2.13.3. Viewing the Modification Ticket

Now we'll explore several ways to view the upcoming change.

First we'll check the information at the user entry:

- If not yet done, click the entry of **Leo Kubalke**.
- Click the **Orders** tab. If it is still empty, perform a refresh (on Windows: F5). Be patient until the data appears.
- You can see the order with the **due date** (the current date), the attribute name **title**, the empty old value and the new value 'Dr.'.

Additionally you can view the ticket:

- Select the **Tickets** view.
- Navigate to **Tickets** → **Internal** → **Users** → *date* → **Kubalke Leo** *time* and click this entry. DirX Identity stores all of its self-created tickets in the tree **Internal**. The next level divides the object types, in this case we have a **Users** folder. Below this folder you can find a folder for each date (this is the due date). This allows easily finding out all tickets that are valid for a specific date and you can also see that for a specific date no tickets exist if the corresponding folder is missing.
- In the **General** tab of the ticket, you can see the **Name** (Kubalke Leo *time*), the **Object type** (here dxrUser) and the O**peration** (here MODIFY). The creator of this ticket is shown as the **Owner** (in this case the Domain Admin). The **Subject** field shows the DN of the object the ticket is valid for. Because the Request workflow field is empty, there is no related approval in progress.
- Check the **Status Information** tab. The **Status** field shows Input.Completed, which means that the ticket is ready to process. You can also see the Due date here. The **Expiration date** and **Delete date** are still empty. If any errors occur, for example, during ticket processing, the **Error** field shows this information.
- In the **Object** tab, you can find the order (the change definition) for the object itself (for

the subject). It is identical to the previously viewed **Orders** tab at the user entry.

- Click the **Assignments** tab to see that this ticket does not have any assignment order.

Use query folders to explore and manage the ticket tree. Especially if you have many tickets, these queries help a lot. If the provided default queries are not specific enough, set up your own queries but first let's use some of the default queries:

- Open the **_Queries** folder. At the top level, you can find a set of general queries.
- Click **Active Tickets** to see our active ticket for Leo Kubalke.
- Because there are no error tickets, clicking **Error Tickets** does not show any entries.
- Click the **Variable Time Constraint** query. A dialog appears that shows by default a three-hour time period for modifications and creations. You could change these values, but in this case, we accept them and simply click **OK**. Again we can see the Leo Kubalke entry.
- There are some additional folders that help to evaluate specific types of tickets. Open **For Object Types** and click **Users**. Our entry is shown again.
- Opening **For Operation Types** and clicking **Modify** reveals the same entry.
- Now open **For Status** and click **Input.Complete** to see the same entry.

These examples show that there are many ways to view and explore tickets.

### 2.13.4. Processing the Modification Ticket

In this step, we'll process the previously created ticket.

- Login to the **Connectivity** view group of the DirX Identity Manager.
- Click **Global View** and then navigate to **My-Company → Main**.
- Click the line between the two **Identity Store** icons and select **Process Internal Tickets → Configure** from the context menu.
- Click through the two tabs. There is nothing interesting to see besides the fact that this workflow runs with **Resource Family** Event_Maintenance, which means it runs in the Java-based server and does not run as an external executable in the C++-based server.
- Now run the workflow (select **Process Internal Tickets → Run** from the context menu.

The workflow runs and we can view the result.

Typically you should set up a schedule for this workflow to run it overnight on a daily basis. This configuration guarantees that all mature tickets are processed. Each day, you should check to see if any errors occurred. Use the **Error Tickets** query to perform this task.

### 2.13.5. Viewing the Modification Ticket Result

Now we'll see if the user entry has changed:

- Log in to the **Provisioning** view group of the DirX Identity Manager.
- Click the **Users** view and then navigate to **Users** → **My-Company** → **Product Testing**.
- Select **Kubalke Leo** and click the **General** tab. You should see the value **Dr**. in the title field. This shows that the ticket was processed.
- Click the **Orders** tab. The tab is empty.

Now we check the ticket itself:

- Click the **Tickets** view.
- Navigate to **Tickets** → **Internal** → **_Queries** and then click the **Processed Tickets** query.
- The **Kubalke Leo** ticket is found.
- Click the **Status Information** tab.
- The **Ticket state** is now ApplyChange.completed and the **Delete Date** is set to one month later.

Where does this one month value come from?

- Click the **Domain Configuration** view and then the top-level object with the name of your domain.
- Select the **Timing** tab.
- In this tab, you can find global settings for the **Ticket life time** of successfully processed tickets and erroneous tickets.

## 2.14. Working with Source Tickets

Many customers have already service management systems (ticketing systems) in place to manage and control IT processes.Users open tickets that allow them to get access to specific resources.These processes are well-known and thus hard to change.When introducing an identity management system into this environment, it makes sense to connect it to these ticketing systems to allow for seamless integration with existing processes.

In this section, we'll show you how to integrate a ticketing system with DirX Identity's provisioning mechanism via a sample web service.First, we explain how you can work with source tickets and then we demonstrate how to:

- Prepare the web service environment.
- View the ticket request.
- Send the ticket.
- Watch the ticket workflow.
- Check the ticket status.
- Approve the Manager Analyst Relations role.
- View the request workflows.

- Check the final ticket state.

For more information about this topic, read Managing Tickets.

### 2.14.1. Understanding How to Use Source Tickets

Tickets from source ticketing systems (service management systems) that are related to identity management can comprise various issues:

- Creation of a new user optionally with already assigned privileges.
- Modification of a user, which means either attribute changes or changes of the assigned privileges.
- Deletion of users.
- Creation, modification and deletion also apply for other object types like roles, permissions or groups.

Configuration of this feature includes these issues:

- Setting up a Web service that is derived from the delivered sample web service that consumes tickets, converts them to Identity internal order representation and starts with that order the corresponding request workflow.
- Copying the provided ticket processing request workflow and adapting it as required. You can set up one or more of these request workflows for each type of object.

These steps have already been performed in the My-Company sample domain. You can use this prepared example to run the following tutorial. You can find more information about source ticketing in the *DirX Identity Use Case Document* "Service Management".

For purposes of this exercise, we assume that My-Company has a service management system in place that manages contractors. This system was connected to DirX Identity via web services. If a new contractor is hired, the hiring information goes through a process in the service management system and a ticket is then created that is consumed and processed by the DirX Identity sample Web service. In our example, the My-Company division in Ottawa, Canada needs help in the professional services area. So they hire the famous Irwin Dough as a contractor. Because the tasks are well-understood, the user is created together with two role assignments: **Manager Analyst Relations** and **My-Company Newsletter**. A ticket request workflow is started that implements the user and his privileges automatically.

### 2.14.2. Preparing the Web Service Environment

DirX Identity comes with a sample ticket web service server and client. To set up the sample web service:

- Copy the complete folder **Additions\ServiceMM\WorkingWithSourceTickets** on your DVD to any location on your machine (for example **C:\SampleTicketWS**).
- Make sure your PATH variable is set appropriately to find the correct version of the program java.exe.

- Start the sample ticketing Web Service. There are two possibilities:
- Starting the Web Service as stand-alone application
- Deploying the Web Service into Tomcat

The Web Service is then accessible via an URL of the form

**http://** *host*: *port* **/sourceTicketing/sourceTicketingService**.

#### 2.14.2.1. Starting the Web Service as Stand-alone Application

Run the file **runServer.bat** to start the sample ticket web service server. The server runs stand-alone and does not require any other environment. The default parameters assume that the web service is accessible via host=localhost and port=40099. It is also assumed that the Java-based server is accessible via host=localhost and port=40000.

If you want to use other parameters, read Service Management.

#### 2.14.2.2. Deploying the Web Service into Tomcat

Perform the following steps to deploy the Web Service into Tomcat:

- Copy the **sourceTicketing.war** file to *tomcat_install_path* **/webapps**.
  Tomcat will automatically detect it and deploy it to *tomcat_install_path* **/webapps/sourceTicketing**.
- Check the request workflow service connection parameters in the file **config.properties** of the folder *tomcat_install_path* **/webapps/sourceTicketing/WEB-INF/config**.
- If necessary, correct them and restart Tomcat. The web service is accessible through the Tomcat host and port. This is the recommended approach for productive environment.

### 2.14.3. Preparing the Sample Web Service Client

In this tutorial, we use a pre-configured Web Service client that is easy to use. It reads a request file and sends it to the sample Web Service server. To set up the sample Web Service:

- Check the setting of the variable **ENDPOINT** in the file **seturl4client.bat** regarding the port number. The file shipped with the product is correct if the related port is **8080**. If the correct port is not 8080, then the setting must be corrected so that it contains the correct port - either the port used when starting the service via **runServer.bat** or the actual Tomcat port, respectively.

### 2.14.4. Viewing the Ticket Request

The sample ticket web service that is delivered with DirX Identity can consume various ticket requests for different types of objects. It is similar to SPML but simplified and thus easier to handle. To learn more about the sample web service, read the corresponding chapter in Extending Web Services.

To understand the details of the request we use in this tutorial, open the file

**CreateUserWithRoles.xml** from the folder **C:\SampleTicketWS\sampleTickets** and view it:

- The request contains a **requestID** (value Request-1) that is important because you need it later on to refer to the sent request if you request the status of it. Typically this ID is generated by the generating service management system or it is equal to the ticket number in that system.
- The **spml:identifier** is of type DN and defines the object to create with its path: **cn=Irwin Dough,ou=Professional Services,o=My-Company,cn=Users,cn=My-Company**
- The **spml:operationalAttributes** section defines operational attributes:
  - **directoryType=dxrUser** - defines the object type to create. This information is used to find the correct ticket workflow via the **When Applicable** section.
  - **operation=CREATE** - specifies the operation to perform on this object.
  - **creator=cn=DomainAdmin,cn=My-Company** - sets the initiator of the workflow. You can use it in the **When Applicable** section.
- The **spml:attributes** section allows specifying all attributes of the user to create as there are for example the **cn**, **objectClass**, **sn**, **givenName**. You can also set links to other objects, for example **manager**, **dxrLocationLink**, **dxrOrganizationLink**, **dxrOULink**.

We cannot set the **dxrOULink** to Professional Services because this would assign the Sales Task role to the user. This role includes SAP relevant groups that cannot be synchronized because the SAP R/3 system is not physically connected. This would cause ticket workflow that cannot end successfully.

- Use the **spml:addPrivilegeAssignment** section to specify the privileges to add together with the user creation. In this case, we add two roles: **Manager Analyst Relations** and **My-Company Newsletter**. The first role must be approved later on, which is not visible here. Of course we could specify time restrictions or - if the role requires it - role parameters. But we keep it simple here. Check the other sample requests delivered with the product to see all features of this sample web service.

Next, we will send this request to the sample web service server.

## 2.14.5. Sending the Ticket

To send the ticket, use the prepared batch file and perform these steps:

- Double click the batch file **CreateUserWithRoles.bat** to run it. The client sends the XML request to the sample web service which starts a request workflow. This requires some seconds. Be patient.
- A log file **CreateUserWithRoles.log** is written. Open it to see the result:

```
CreateUserWithRoles.id=Request-1
Connecting to http://localhost:8080/sourceTicketing/sourceTicketingService
CreateUserWithRoles.result=URN_OASIS_NAMES_TC_SPML_1_0_SUCCESS
CreateUserWithRoles.correlationId=141cb8e7b72$7bcc
```

The third line shows that the request ended with SUCCESS.

The fourth line delivers the internal correlation ID that can be used for status requests alternatively to the request ID.

If everything worked as expected, we can view the resulting ticket workflow.

### 2.14.6. Watching the Ticket Workflow

The sample ticket web service evaluates the incoming tickets and starts the request workflow that fits best. Perform these steps to view and monitor the workflow:

- Log in to the DirX Identity Manager's **Provisioning** view group.
- Select the **Workflows** view and navigate to **Monitor** → **My-Company** → **Service Management** → **Process Ticket** → *date* → **Irwin Dough**.
- Click the **General** tab to view the workflow progress.
- You can see **Add attributes** and **Apply order** steps that are already completed. The sample web service converted the ticket into an Identity order that was passed to the ticket request workflow.
- The first activity **Add attributes** added two fixed value attributes. To understand this, navigate to the workflow definition: **Definitions → My-Company → Service Management → Process Ticket**.
- Open this entry, click the **Add attributes** activity and then click the **Parameters** tab. You can see that the attribute **employeeType** is set to "Contractor" and the **dxmOprMaster** attribute to "Service Management".
- Check the result in the workflow instance: click the **Object** tab. You should see all attributes from your ticket definition and additionally the two attributes that were set by the **Add attributes** step.
- Click the **Assignments** tab. Here you should see the two requested assignments from the ticket definition.
- The second activity **Apply order** performed a sequence of steps:
- It created the user in the Identity Store at the requested location. Check that the user exists (**Users → My-Company → Professional Services → Irwin Dough**). Step through the tabs to verify that the attributes were correctly set. You can see that many attributes were set automatically through attribute mastering from the business object **Location**.
- It performed a privilege resolution. Check the **Assigned Roles**, **Assigned Permissions**, **Assigned Groups** and **Accounts** tabs to see the result.
- One role was assigned automatically and is already resolved (**Contractor** role via rule).
- The role **My-Company Newsletter** was resolved immediately because it is not flagged for approval.
- The role **Manager Analyst Relations** requires approval. Thus an approval child workflow was started from the parent (ticket) workflow.
- You can check this workflow either directly under **Monitor → My-Company** → **Approval** → **4-Eye Approval** → *date* → **Irwin Dough**→ **Manager Analyst Relations** or you can use the

parent workflow as starting point.

- Click the parent workflow instance again and then select the **Child Workflows** tab. You can see the started child workflow as a line in the table.

**State** and **ApplicationState** are empty for child workflows not yet finished. Click the line and then the icon right beside the table. The Manager navigates to the child workflow instance. You can view it and then return to the parent workflow (use the arrow button of the manager).

- We check the configuration of the **Apply order** activity to understand the settings in more detail. Click this activity in the workflow definition and select the **Parameters** tab. The meaning of the flags is:
- If **Apply Subject Order** is flagged, the activity analyzes the order and creates or modifies the object.
- If **Track Changes** is checked, all account and group relevant changes that are caused by this activity are stored in a list of provisioning changes that can be used later on by the **Wait for completed provisioning** activity.
- If **Track Changes in Child Workflows** is checked, then child workflows get the Track Changes flag checked. This setting forces the child workflow to propagate the list of provisioning actions to its parent. This means that the parent has a consolidated list of all changes that a **Wait for completed provisioning** activity can use.

Before we perform the pending approval task, we check the ticket status via the web service.

### 2.14.7. Checking the Ticket Status

Meanwhile, a user in the ticket system wants to know whether the ticket is processed. Regardless of how this is done, the ticket system has to send a status request to DirX Identity. You can simulate this procedure.

First, we view the status request:

- Open the file **sampleTickets/CheckStatus.xml** for editing.
- Change the status request so that the operational attribute **correlationID** is exactly the correlation ID of the response that was received from **CreateUserWithRoles.xml** request. The correct value is the value of **CreateUserWithRoles.correlationId** from the file **CreateUserWithRoles.log**.The **correlationID** is the identifier of the workflow.
- Remove the xml-comment characters which enclose the operationalAttributes section. This ensures that the status is checked first via correlationID and then via requestID as fallback.
- If not yet done, change the requestID value so that it is **checkStatus-1** instead of **Request-1**. This way, the status request will only be successful in case of a correct correlationID in the request.
- Save the file.

Now we can start the status request.

- Run the prepared **CheckStatus.bat** batch file.

Check the result:

- View the **CheckStatus.log** file.

```
CheckStatus.id=Request-1
Connecting to
http://localhost:8080/sourceTicketing/sourceTicketingService
CheckStatus.result=URN_OASIS_NAMES_TC_SPML_1_0_PENDING
CheckStatus.correlationId=141cb8e7b72$7bcc
```

- You can see the request ID in the first line. The third line shows the status: **PENDING**. This indicates that the ticket processing in DirX Identity has not yet completed.
- The correlation ID in the fourth line is the internal ID of the ticket in DirX Identity.

The response result can be used by the ticket system to present the associated information to the end user.

### 2.14.8. Checking the Ticket Status by Request Identifier

In a scenario where a ticket client is able to send tickets with unique identifiers for the **requestID** parameter, the status can be checked by this request identifier. You can simulate this procedure.

First, we view the status request:

- Open the file **sampleTickets/CheckStatusByRequestID.xml**.

The value for **requestID** is the same as for **sampleTickets/CreateUserWithRoles.xml**. Close the file.

Now we can start the status request:

- Run the prepared **CheckStatusByRequestID.bat** batch file.

Check the result:

- View the **CheckStatusByRequestID.log** file:

```
CheckStatus.id=Request-1
Connecting to
http://localhost:8080/sourceTicketing/sourceTicketingService
CheckStatus.result=URN_OASIS_NAMES_TC_SPML_1_0_PENDING
```

```
CheckStatus.correlationId=141cb8e7b72$7bcc
```

- You can see the request ID in the first line. The third line shows the status: **PENDING**. This indicates that the ticket processing in DirX Identity has not yet completed.
- The correlation ID in the fourth line is the internal ID of the ticket in DirX Identity. You could use it alternatively to request the status, see the section "Checking the Ticket Status" above.

The response result can be used by the ticket system to present the associated information to the end user.

### 2.14.9. Approving the Manager Analyst Relations Role

Now we have to approve the pending Manager Analyst Relations approval workflow.

- Log in to Web Center as **Benetton Gianfranco**.
- Click **Work List** → **Task list** and click the relevant item.
- Enter a reason and accept the request.
- Log out and log in again as **Bellanger Lionel**.
- Click **Work List** → **Task list** and click the relevant item.
- Enter a reason and accept the request.
- Log out of Web Center.

Next, we'll view the status of the request workflows.

### 2.14.10. Viewing the Request Workflows

First we check the completed approval workflow:

- Log in to DirX Identity Manager.
- Select the **Workflows** view.
- Navigate to **Monitor** → **My-Company** → **Approval** → **4-Eye Approval** → *date* → **Irwin Dough** → **Manager Analyst Relations**.
- Click the **General** tab and verify that the workflow is completed. The **Apply Approved Privilege Change** activity is green.

Because the **Track Changes** flag was set for this child workflow from the parent workflow, the child workflow propagated the completion to the parent workflow. We can see the result in the ticket workflow:

- Select the **Request Workflows** view and navigate to **Monitor** → **My-Company** → **Service Management** → **Process Ticket** → *date* → **Irwin Dough**.
- Click the **General** tab. The **Wait for child workflows** activity is green (successfully completed) and the **Wait for completed provisioning** activity is either in progress or also green (successfully completed). If not, wait until the activity and the workflow are

complete.

The last step checked every minute that all states of all relevant accounts and groups have changed as expected. This means that provisioning was performed correctly.

### 2.14.11. Checking the Final Ticket State

Meanwhile, a user in the ticket system wants to know the status of the ticket, so we need to send another status request.

- Run the prepared **CheckStatus.bat** batch file again.

Check the result:

- View the **CheckStatus.log** file:

```
CheckStatus.id=Request-1
Connecting to http://localhost:8080/sourceTicketing/sourceTicketingService
CheckStatus.result=URN_OASIS_NAMES_TC_SPML_1_0_SUCCESS
CheckStatus.correlationId=141cb8e7b72$7bcc
```

- The third line shows that the ticket could be processed correctly.

The response result can be used by the ticket system to present the associated information to the end user.The ticket was correctly and completely implemented.

## 2.15. Managing Personas

Personas are special representations of one real identity.To understand the concept of personas you should be familiar with chapter "Managing Personas" in the *Provisioning Administration Guide*.

In this section, we show you how to:

- Enable persona management.
- Create and view personas in Web Center.
- Prepare the persona environment.
- Create a persona from a non-primary account.

The professional license is required to use persona management.

### 2.15.1. Enabling Persona Management

To enable persona management:

- In DirX Identity Manager, select **Domain Configuration** of the Provisioning domain.
- In the **General** tab, check **Enable Persona Handling** if it's not already checked.
- Save the update and then re-start the IdS-J service, Apache Tomcat, and DirX Identity

Manager so that enabling persona management takes effect.

## 2.15.2. Handling Personas in Web Center

This part of the exercise demonstrates how to handle personas in Web Center.

### 2.15.2.1. Creating a User's Main Identity

To create a user's main identity, you simply create a user. We create the user John Smith, who will act as the main identity in our tutorial:

- Log in to Web Center as **Taspatch Nik**.
- From the **Users** menu, select the operation **Create new user**.
- Select the workflow **Create a user stepwise without approval**. The **Enter Attributes** dialog appears.
- Select the folder **Users** → **My-Company** → **Sales** → **Sales Europe**.
- Specify the following data for John Smith in the fields provided:
  Last Name: **Smith**
  First name: **John**
  Description: **Main identity for persona tutorial**
  Employee type: **Internal**
  Employee number: **EN-7716**
  Location: **My-Company London**
  Manager: **Richter Sven**
  Company: **My-Company**
  Organizational unit: **Sales**
  Phone: **+44 324 234-6944**
  E-Mail: **John.Smith@My-Company.com**
- Click **Save** to submit the data. The **Request Privileges** dialog is displayed.
- Assign the role **Internal Employee** and then click **Save** to submit the data.
- The workflow now creates the new user **John Smith**.

### 2.15.2.2. Verifying John Smith's Main Identity

To verify John Smith's main identity:

- In Web Center, search user **Smith John**.
- In the search result table, click on John Smith's name to display the overview page.
- Verify whether the data you entered are correct.

### 2.15.2.3. Creating a Persona for John Smith

John Smith will work for a specific time-period for My-Company in Houston, USA. Therefore Nik Taspatch creates a persona for John Smith. John Smith's user entry acts as template for the persona entry to be created:

- In Web Center, select user John Smith.

- From the **Users** menu, select the operation **Create new persona**.
- Select the workflow **Create a persona stepwise without approval**. The **Enter Attributes** dialog appears.
- The folder **Sales Europe, Sales, My-Company, Users** is pre-selected. Last name and first name are initialized with John Smith's data.
- Specify the following data for John Smith's persona in the fields provided:
  Description: **Persona for John Smith's location in USA**
  Employee type: **Internal**
  Employee number: **EN-7716**
  Location: **My-Company Houston**
  Manager: **Richter Sven**
  Company: **My-Company**
  Organizational unit: **Sales**
  Phone: **+1 214 324-46387**
  E-Mail: **John.Smith@My-Company.us**
- Click **Save** to submit the data. The **Request Privileges** dialog is displayed.
- Assign the role **Internal Employee** and then click **Save** to submit the data.
- The workflow now creates the new persona **John Smith**.

#### 2.15.2.4. Viewing John Smith's Main Identity and Persona

In Web Center, specify **Smith** in the quick search panel and press **Enter**.

In the search result table, the user **Smith John** and his persona **Smith John Psn** are displayed. The table displays the phone numbers of the user and his persona.

Double-click the persona entry in the table to display the persona overview page. It turns out that the user's email address is displayed instead of the specified value.

The main identity is displayed on the bottom of the overview page (in the references block). Clicking the name (Smith John) displays the main identity's overview page.

Clicking **Personas** shows the list of all personas assigned to the user. Clicking on the icon at the right in the table displays the persona's overview page.

Clicking **Accounts** on the persona overview page displays two accounts: one in the Intranet Portal and one in the Windows Domain USA.

So, with his identity and with his persona, John Smith now has accounts in Windows Domain Europe *and* in Windows Domain USA, and he is able to work at both locations.

The search result illustrates that there are the following problems:

- The persona's email address is overwritten with the user's data.
- The create workflow can be optimized.

Therefore you must change the configuration to prepare DirX Identity for using persona functionality. The following sections describe this task.

## 2.15.3. Preparing the Persona Environment

We assume that the main identity only has personas that have the same employeeType, employeeNumber and company as the main identity.

This assumption and the evaluation from the sections above result in the following requirements:

- The persona's email address can be different from the user's email address.
- The user's main identity serves as a template for the persona. It inherits the attributes employeeType, employeeNumber and company. In the create persona dialog of Web Center, these attributes must be initialized with the values of the main identity and be read-only.
- The persona's common name is built according the rule:

  cn=sn givenName employeeNumer P

  for example, **Smith John EN-7716 P**.

  Starting with the user's second persona, a counter is appended to the common name; for example, **Smith John EN-7716 P1** for John Smith's second persona.

To satisfy these requirements, you must modify object descriptions and Java scripts in the domain configuration with DirX Identity Manager and a properties file of Web Center. The following sections provide details about these modifications.

#### 2.15.3.1. Modifying the PersonaCommon.xml Object Description

Recall from "Preparing the Persona Environment" that:

- The user's main identity inherits the values of employeeNumber, emplyeeType and organization to the persona.
- The user's main identity does not inherit the email address value.

In the PersonaCommon.xml object description, the **master="owner"** attribute specifies the inheritance of property values. To remove this attribute for the email address and add this attribute for the attributes employeeType, employeeNumber and company:

- Log in to DirX Identity Manager → **Provisioning**.
- Select the **Domain Configuration** view.
- Edit the **PersonaCommon.xml** object description under **My-Company** → **Customer Extensions** → **Object Descriptions**.
- From **<property name="mail"** ... **/>**, remove **master="owner"** to disable the inheritance of the user's email address to the persona.
- To the property descriptions of employeeType, employeeNumber and company add **master="owner"**:

  **<property name="employeeType"** ... **master="owner" />**

**<property name="employeeNumber" … master="owner" />**
**<property name="dxrOrganizationLink" … master="owner" />**

to enable inheritance from the user.

You must add the entire property description for **employeeNumber** because this property is imported from the **Persona.xml** object description.

Now whenever the persona is saved, the properties marked with **master="owner"** are updated from the persona's owner, the user's main identity.

#### 2.15.3.2. Modifying the PersonaFromUser.xml Object Description

Recall from "Preparing the Persona Environment" that the user serves as a template when creating the persona.

The PersonaFromUser.xml object description specifies how the persona is created from its template, the user's main identity. Perform the following steps to modify this object description:

- You're already logged in to DirX Identity Manager → **Provisioning** → **Domain Configuration** view.
- Edit the **PersonaFromUser.xml** object description under **My-Company** → **Customer Extensions** → **Object Descriptions**.
- Add the following property description for company:

  **<property name="dxrOrganizationLink" type="java.lang.String" >**
  **<extension>**
  **<namingRule>**
  **<reference baseObject="SvcUser" attribute="dxrOrganizationLink" />**
  **</namingRule>**
  **</extension>**
  **</property>**

  As a result, the persona's organization is initialized with the user's organization value in the create persona dialog.

The corresponding property descriptions for employeeNumber and employeeType already exist.

Now the user's main identity is used as a template for the persona and the attributes employeeNumber, employeeType and company are inherited from the user's data. The last task in the DirX Identity Manager is to specify the building rule for the persona's common name. The following section describes this task.

#### 2.15.3.3. Modifying the Java Script CommonNameForPersona.js

Recall from "Preparing the Persona Environment" that the persona's common name is built

from the user's surname, given name, employeeNumber, an appended **P** and a counter. The existing script uses the surname and the given name to build the persona's common name. It appends the suffix **Psn** and a counter starting with the second persona.

The Java script CommonNameForPersona.js specifies the building rule for the persona's common name. Perform the following steps to modify this Java script:

- You're already logged in to DirX Identity Manager → **Provisioning** → **Domain Configuration** view.
- Edit the Java script **CommonNameForPersona.js My-Company** → **Customer Extensions** → **JavaScripts**.
- To include the employeeNumer to the persona's common name, you must first add the variable definition for emplyeeNumber. Insert the following definition after the definition for **sn**:

  **var employeeNumber = obj.getValue("employeeNumber");**
- The building rule for the persona's common name is specified in the variable definition for **cn**. Change this definition to:

  **var cn = sn + " " + givenName + " " + employeeNumber + " P";**
- Finally, you must change the instruction in the while statement below to:

  **cn = sn + " " + givenName + " " + employeeNumber + " P"+i;**

Now you have completed all of the tasks that you must perform with DirX Identity Manager:

- The user's main identity is used as a template for creating the persona.
- The persona's e-mail address can be different from the user's.
- The employeeNumer, the employeeType and the organization are inherited from the user's data.
- The persona's common name is built from the user's surname, givenname, employee number and the suffix **P** followed by a counter starting with the second persona.

The final configuration task now is to set the fields **Employee number**, **Employee type** and **Company** to read-only in Web Center to prevent the user from changing these values when creating the persona in Web Center. The following section describes this task.

#### 2.15.3.4. Modifying defaultRenderer.properties

A convenient way to create a persona is to create it in Web Center. In this tutorial, we don't permit changing the persona's employeeType, employeeNumber and company (dxrOrganizationLink). Therefore, the fields for these attributes should be set to read-only in the create persona dialog of Web Center.

To set the input fields for these attributes to read-only in the create persona dialog of Web Center, edit the Web Center configuration file **defaultRenderer.Properties** under *install_path***web\webCenter-My-Company\webCenter\WEB-INF\config** and add the

following instructions:

**employeetype@dxrpersona = roText**
**employeenumber@dxrpersona = roText**
**dxrorganizationlink@dxrpersona = roOrganizationSearch**

Before you can verify your modifications, you must re-start the IdS-J server and the Apache Tomcat server hosting your Web Center so that the modifications take effect. The following section describes how you can verify your modifications.

#### 2.15.3.5. Verifying the Updated Persona Environment

To verify your modifications:

- Log in to Web center as **Taspatch Nik**.
- Search for **Smith John**.
- Select the user entry to display John Smith's overview page.
- Select the **Create new persona** operation from the **Users** menu.
- Select the workflow **Create a persona stepwise without approval**. The **Enter Attributes** dialog is displayed.
- Check that the fields **Employee number**, **Employee Type** and **Company** are initialized with the correct values and are read-only.
- Specify values in the fields **Description**, **Phone** and **E-Mail**. and then click **Save**.
- Assign the role **Internal Employee** to the persona and then click **Save**. Now the new persona is created.
- Search again for **Smith John** to display his user and persona entries.
- Check that the new persona was created with the correct common name **Smith John EN7716 P**[*counter*] and the correct attribute values for **Employee number**, **Employee Type**, **Company**, **Description**, **Phone** and **E-Mail**.

### 2.15.4. Creating a New Persona for an Imported Non-Primary Account

This tutorial illustrates how to handle the case where a user has more than one account in a target system. This situation can occur, for example, if a user has his personal Active Directory (AD) account and an imported administrative account. The personal AD account is the user's primary account. He uses this account for his usual tasks. Sometimes the user must act as the system administrator. He uses the second imported administrative account for this purpose.

DirX Identity only manages one account per user in a target system, called the "primary" account.

In this tutorial, you create a persona for the non-primary account. Once the persona is created, the account is unassigned from the user and assigned to the persona. As a result of this procedure, the account is then the primary account for the persona and you can manage it with DirX Identity in the usual way.

#### 2.15.4.1. Preparing the Tutorial

In this tutorial, we use the account that we created for John Smith in the previous sections as the primary account. In this step, we create the account **Administrator** as a second account for John Smith because he must take the place of the system administrator while he is on vacation.

To create the data used in this tutorial:

- In DirX Identity Manager → **Provisioning** → **Target Systems** view, select the target system **Windows Domain Europe** → **Accounts and Groups**.
- Select **New** → **Account** to create a new account **Administrator**.
- In the **General** tab, specify the following data:
  Name: **Administrator**
  Type: Uncheck **Primary account**.
  User Data: Select the test user **Smith John** created in the previous sections (in **Users** → **My-Company** → **Sales** → **Sales Europe**). His data are displayed in the remaining fields. Leave the remaining data as they are.
- In the **Active Directory** tab, specify the value **admin** for the mandatory field **Account Name**. This value is re-created according to the naming rule when the account is saved.

  Copy the **PrimaryKey (DN in TS)** value **CN=Administrator,cn=Accounts** for later use.
- Click **OK** to create the new account.

The next steps must be performed in the **Data View**:

- Right-click on the new account **Administrator** and select **Goto DataView** from the context menu. The **Data View** opens and displays the attributes of the account **Administrator**.
- Edit the **All Attributes** tab and specify the value **IMPORTED** for the **dxrState** attribute.
- Select the group **cn=Administrator** under **My-Company cn=My-Company** → **cn=TargetSystems** → **cn=Windows Domain Europe** → **cn=Accounts and Groups** → **cn=General**.
- Edit the **All Attributes** tab and then specify the account's primary key value **CN=Administrator,cn=Accounts** (which you copied when you created the new account) for the **dxrGroupMemberImported** attribute.

Now the user has two accounts in the same target system: the primary account **John Smith** account and the **Administrator** account.

The state of the non-primary Administrator account and the group membership of the Administrator group are IMPORTED.

#### 2.15.4.2. Verifying John Smith's Accounts

Verify John Smith's data:

- In DirX Identity Manager → **Provisioning** → **Users** → **My-Company** → **Sales** → **Sales Europe**,

select the user **Smith John**.

- Click the **Accounts** tab. For the target system **Windows Domain Europe**, two accounts are displayed: **John Smith EN-7716** and **Administrator**. The Administrator's state is **IMPORTED**.
- Click the **Assigned Groups** tab. John Smith has the assigned group **Administrator**. Its state is **IMPORTED**.

Now we have prepared and verified the test data for the tutorial.

You can create a persona manually for the imported Administrator account as follows:

- Create a new persona for user John Smith. (See "Creating a Persona for John Smith" for details.)
- In DirX Identity Manager, edit the **Administrator** account and then assign the new created persona to this account in the **General** tab.
- In Web Center or DirX Identity Manager, perform a direct group assignment of group **Administrator** to the new persona.

Instead of creating the persona manually, you can use rules and scripts in DirX Identity to create personas automatically. The following sections describe how to create a persona for John Smith's non-primary **Administrator** account using rules and scripts.

#### 2.15.4.3. Modifying the Consistency Rule CreatePersonasForNonPrimaryAccounts

We'll use the consistency rule **CreatePersonasForNonPrimaryAccounts** to create the persona for John Smith's **Administrator** account. This rule is intended to automate persona creation for non-primary accounts. In DirX Identity Manager's **Provisioning** → **Policies** view, you'll find this rule under **Policies** → **Rules**→ **Default** → **Consistency** → **Personas**.

Since the rule and the algorithm are widely configurable, we'll need to adapt it to our requirements before we run it in a later step:

- In the **General** tab → **nameOfJavaScript**, we check the name of the Java script.
  We use the script **AccountToPersona.js** and do need not change the specified name here.
- In the **Filter** tab, the **Search base** results in processing the accounts of all target systems. We want to limit the processing to the target system containing our test data, so we specify the value **cn=Windows Domain Europe,cn=TargetSystems,cn=My-Company**.
- In the **Filter** tab, the initialized **Search filter** is **( objectclass="dxrTargetSystemAccount" and dxruserlink=* and ( not ( dxrisprimary=* ) or dxrisprimary="false" ) )**. That is, that the rule is applied only to non-primary accounts that are already assigned to a user. Since this configuration matches our needs, we leave the filter as it is.

In the next step, we modify the Java script **AccountToPersonas.js**.

#### 2.15.4.4. Modifying the Java Script AccountToPersonas.js

The Java script **AccountToPersonas.js** performs the assignment of accounts to personas. A default script is provided under DirX Identity Manager → **Provisioning** → **Domain Configuration** view → **My-Company** folder → **JavaScripts**.

When running the script, the application searches this script under the following locations:

- In the **JavaScript** folder of the target system (for target system-specific configuration).
- In the **JavaScript** folder of the customer extensions.
- In the **JavaScript** folder for system configuration (the location of the default script).

We don't want to change the original default script, so we copy it to the target system-specific folder **Target_Systems** → **Windows Domain Europe** → **Configuration** → **JavaScripts**:

- Right-click the default script and then select **Copy** from the context menu.
- Browse to the target system-specific folder.
- Right-click the target system specific folder and select **Paste** from the context menu.

Next, we'll edit our copied script:

- We add the attributes employeeType, employeeNumber and telephoneNumber to the user attributes to copy:

  var userAttributesToCopy = new Array("dxrOrganizationLink", "dxrOULink", "dxrLocationLink"**, "employeeType", "employeeNumber", "telephoneNumber"**);
- The common name of the persona should consist of the main identity's surname and given name, the employee number, the common name of the account, and an appended **P** to indicate the persona entry. So we change the rule for creating the persona to:

  **var cnOfPersona = user.getValue("sn") + " " +user.getValue("givenName") + " " + user.getValue("employeeNumber") + " - " + account.getValue("cn") + " P";**

Now we are ready to run the consistency rule for creating a persona for a non-primary account.

#### 2.15.4.5. Running the Consistency Rule CreatePersonasForNonPrimaryAccounts

To run the consistency rule **CreatePersonasForNonPrimaryAccounts**:

- Log in to DirX Identity Manager → **Connectivity**.
- In **Global View** → **Scenarios** → **Default**→ **Identity Store**, select the workflow line connecting the Identity Store with itself.
- Right-click the workflow line and then select the **Run...** for the **CreatePersonasForNonPrimaryAccounts** workflow from the workflow list provided in the context menu.

The rule starts a persona create workflow for user John Smith. The workflow requires that

John Smith approve creating the persona. This task is described in the next section.

#### 2.15.4.6. Approving the Persona Create Workflow

To approve the persona create workflow:

- John Smith must approve creating the persona for his own main identity. We first must specify a password for John Smith so that he can log in to Web Center. Log in to DirX Identity Manager → **Data View**, select the user John Smith (cn=Smith John,ou=Sales Europe,ou=Sales,o=My-Company,cn=Users,cn=My-Company), edit the **All Attributes** tab, and specify a **userPassword**.
- Now log in to Web Center as **Smith John**.
- The **My Tasks** column displays the tasks John Smith must approve. Verify that it contains an Approve Creation task for **Smith John EN-7716 - Administrator P.** This is the new persona that the **CreatePersonaForNonPrimaryAccounts** workflow wants to create.
- Click on this task to display the **Approval** page. The page displays the new persona's data and the direct group assignment for the **Administrator** group.
- Click **Accept** to create the new persona.

Do not log out of Web Center yet: we want to verify the new persona and the main identity as the last step.

#### 2.15.4.7. Verifying the New Persona and the Main Identity

In Web Center, perform the following steps to verify the new persona and the main identity of John Smith:

- From the **Self Service** menu, select the **Display Summary** operation.
- Select the **Accounts** tab to verify that the **Administrator** account is no longer assigned to the main identity.
- Select the **Groups** tab to verify that the **Administrator** group is no longer assigned to the main identity.
- Select the **Personas** tab and then select **Smith John EN-7716** - **Administrator P**.The persona's overview page is displayed.If you are not allowed to view your persona, you can enable the access policy **Users can handle their personas** in DirX Identity Manager, or you can log in as **Taspatch Nik** in Web Center, search for user **Smith** and access the persona **Smith John EN-7716 - Administrator P**.
- Select the **Accounts** tab to verify that the **Administrator** account is assigned to the new persona and that its state is **ENABLED**.
- Select the **Groups** tab to verify that the **Administrator** group is assigned to the new persona and that its state is **ENABLED**.

## 2.16. Managing Functional Users

Functional users are used to model resources like meeting rooms, team mailboxes, or

trainee accounts.To understand the concept of functional users, you should be familiar with chapter "Managing Functional Users" in the *Provisioning Administration Guide*.

In the following sections, we show you how to:

- Enable functional user management.
- Prepare the functional users environment.
- Create and view functional users in Web Center.

The Professional Suite (Pro Suite or Pro Upgrade license) is required to use functional user management.

## 2.16.1. Enabling Functional User Management

To enable functional user management:

- In DirX Identity Manager → **Provisioning** → **Domain Configuration** view, select the domain object; for example, **My-Company**.
- Click **Edit** at the domain object.
- In the **General** tab, check **Enable Functional User Handling**.
- Save the update and then re-start the IdS-J service, Apache Tomcat, and DirX Identity Manager so that enabling functional user management takes effect.

## 2.16.2. Preparing the Functional Users Environment

Handling functional users is similar to handling personas. Both processes require a create workflow that can be adapted to the customer's requirements. However, although the same mechanism applies, personas and functional users have separate configuration files: in the sample domain, functional users are created in a common folder **Users** → **My-Company** → **Resources,** whereas personas are created in the same folder as their main identities.

In this tutorial, we assume that the customer requests the creation of department-specific trainee users with the following requirements:

- The common name (cn) of the department-specific trainees must be **Trainee for** *department*.
- The trainees' sponsors are the department managers. They create the trainees' functional users.
- The new trainees must be located in the same folder as their sponsor, in the department folder.

The following sections describe how to create a new workflow and a new object description in DirX Identity Manager that satisfies these requirements.

#### 2.16.2.1. Creating the Workflow Create Trainee for Department

In DirX Identity Manager → **Provisioning** → **Workflows** view, create the new workflow as a copy of the workflow **Create Functional User Without Approval**:

- Browse to **Workflows** → **Definitions** → **My-Company** → **Functional Users**.
- Right-click **Create Functional User Without Approval** and then select **Copy Object** from the context menu. The **Copy** dialog box opens.
- In the **Name** field, specify the new name **Create a trainee for the department** and then click **OK**. The copy is created in the **Functional Users** folder.

Now we'll adapt the workflow to the requirements described in "Preparing the Functional Users Environment":

- In the properties window of the workflow, click **Edit**.
- In the **General** tab, double-click the first activity **Functional User from User** to edit its parameters:
- Select the **Parameters** tab.
- Specify **TraineeFromUser** in the **Name of Object Description** field because we use our own object description **TraineeFromUser** to create the data from the template user.
- Delete the information in the **Parent folder for subject** field. If it is empty, the template user's parent folder is used - that is exactly what we need.
- Click **OK**.
- Select the **Workflow** tab.
- Specify **Create a trainee for the department** in the **Description** field.
- Click **Save** to save the changes.

In the next step, we create the object description **TraineeFromUser**.

#### 2.16.2.2. Creating the Object Description TraineeFromUser

In DirX Identity Manager → **Provisioning** → **Domain Configuration** view, create the new object description **TraineeFromUser.xml** as a copy of **FunctionalUserFromUser.xml**:

- Browse to **My-Company** → **Customer Extensions** → **Object Descriptions**.
- Right-click **FunctionalUserFromUser.xml** and then select **Copy Object** from the context menu. The **Copy** dialog box opens.
- In the **Name** field, specify the new name **TraineeFromUser.xml** and then click **OK**. The copy is created in the **Object Descriptions** folder.

Next, we'll adapt the object description to the requirements described in "Preparing the Functional Users Environment":

- In the properties window of the object description, click **Edit**.
- In the **General** tab, specify **TraineeFromUser** in the **Description** field.

- Select the **Content** tab.
- Change the object name to **TraineeFromUser**:

```
<object name="TraineeFromUser"/>
```

- Change the naming rule for the **sn**:

```
<property name="sn"
    ...
        <namingRule>
                <reference baseObject="SvcUser"
address="dxrOULink" attribute="ou" />
     ...
</property>
```

Now the trainee gets the department name as its surname.

- For the givenName, use the fixed value **Trainee**:

In the naming rule for **<property name="givenName"**, replace the line

```
        <reference baseObject="SvcUser" attribute="givenName" />
```

with

```
         <fixedValue value="Trainee" />
```

- Each department can have just one trainee. The trainee's common name must be **Trainee for** *department* **Department**. So use the following naming rule for cn:

```
<property name="cn"
   type="java.lang.String" >
       <extension>
           <namingRule>
               <fixedValue value="Trainee for the " />
               <reference baseObject="SvcUser" address="dxrOULink"
attribute="ou" />
               <fixedValue value=" Department" />
           </namingRule>
       </extension>
```

```
</property>
```

- Now the object description is complete. Click **Save** to save the changes.

The new object description must be referenced to be included at program start. For this purpose, we add an import to the **main.xml** object in the **Configuration** → **Customer Extensions** → **Object Descriptions** folder:

- Select the **main.xml** object description and then click **Edit** in the properties window.
- Select the **Content** tab.
- Insert the following line after **<config>**:

```
<import
file="storage://DirXmetaRole/cn=TraineeFromUser.xml,cn=Object
Descriptions,cn=Customer
Extensions,cn=Configuration,$(rootDN)?content=dxrObjDesc"/>
```

- Click **Save** to save the changes.

Finally, restart the IdS-J service so that the changes become effective.

Now we are ready to create a new trainee for a department. The following section describes this task.

### 2.16.3. Creating a New Trainee for the Department

To create a trainee for a department:

- Log in to Web Center as **Taspatch Nik**.
- Search user **Berner Hans** from the **Human Resources** department. He is the manager of this department.
- Select **Berner Hans**. His overview page is displayed.
- Select the **Functional Users** tab to verify that he does not already have a functional user assigned.
- Select the **Create new functional user** operation from the **Users** menu. The workflow selection page appears.
- Select the workflow **Create a trainee for the department**.The **Enter Attributes** page appears.The values for Name, First Name and Last Name are initialized with default values as expected from the edits in the **TraineeFromUser.xml** object description.
- Click **Save.** The **Request Privileges** page appears.
- Click on **Groups** and assign the group **dxr Mailbox Users** of the **Windows Domain Europe** target system.
- Click **Save**.

In DirX Identity Manager, verify that the functional user **Trainee for the Human Resources Department** is created in the **Human Resources** folder:

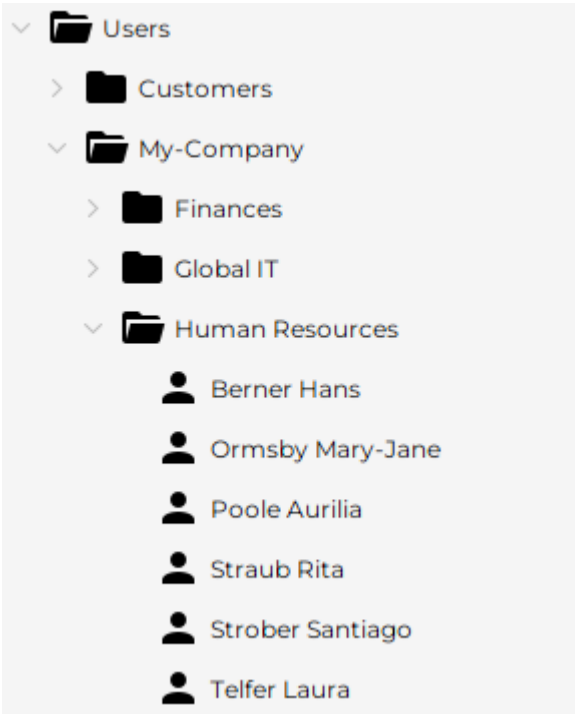



*Figure 17. Functional User Trainee for the Human Resources Department*

# 2.17. Using Risk Management

In this exercise, we show you how to activate and use DirX Identity's risk management system.DirX Identity's Risk Calculation workflow can calculate a risk value for every user in a domain depending on the privileges assigned to the user.The workflow then assigns this risk value to three different levels: low, medium, and high, which are displayed in the different graphical user interfaces at the user.DirX Identity's risk management system also provides a risk approval workflow that simulates a risk level resolution for a requested privilege assignment and enforces an additional approval step if the risk level is computed to be high.

## 2.17.1. Activating Risk Management

First, we'll activate risk management at the sample **My-Company** domain using DirX Identity Manager's **Provisioning** → **Domain** Configuration view:

- In DirX Identity Manager → **Provisioning** → **Domain Configuration** view, select the **My-Company** domain object.
- Click **Edit**.
- In the Compliance tab, check **Risk Check active** if it isn't already checked.
- Click **Save**.

Next, we'll add the **RiskGovernance** resource family to the resource families of the Java-based Server:

- Log in to DirX Identity Manager's **Connectivity** view group, and then select the **Expert View**.
- Navigate to **Connectivity Configuration Data** → **Configuration** → **DirX Identity Servers** → **Java Servers** → **My-Company** → **My-Company-S1-***servername* and click on the last entry.
- In the Resource Families tab, click **Edit**. Now move **RiskGovernance** from the **Available** table to the **Selected** table. Click **Save**.

Now we'll activate the risk policy from the **Provisioning** → **Policies** view:

- Navigate to **Policies** → **Risk Policies** → **Risk Policy**.
- In the **General** tab, check **Is active** (if it's not already activated).
- The **Risk Limits** fields in the **General** tab define values used for risk classification into low, medium, and high risk levels. We'll use these values in this exercise, so leave them as they are.
- The **Risk Factors** section in the **General** tab allows you to define up to nine different risk factors and their corresponding risk weights. We'll use the risk factors and weights supplied with this risk policy, so leave this section as it is.

Finally, we'll activate the Risk Calculation workflow **RiskGvnController** in the sample domain:

- In **Connectivity**, select the **Expert View**.
- Navigate to **Connectivity Configuration Data** → **Workflows** → **Default** → **Identity Store**, copy the entry **RiskGvnContoller** to Connectivity **Configuration Data** → **Workflows** → **My-Company** → **Main** → **Identity Store** by pressing and holding down the Strg-key and the left mouse button and then moving the cursor with this workflow to the new location.
- Click on the new entry **Connectivity Configuration Data** → **Workflows** → **My-Company** → **Main** → **Identity Store** → **RiskGvnContoller**.
- In the **General** tab, check **Is Active**.

Now we need to restart the Java-based Server and Apache Tomcat for Web Center for our changes to take effect.

## 2.17.2. Configuring Risk Values for Target Systems and Groups

The calculation of the risk values for the users depends on the risk factors and their risk weights defined in the risk policy and the defined risk weights of the target systems and of their individual groups. For the sample domain, the risk weights have already been specified:

- In **Provisioning** → **Target Systems**, click **Intranet Portal** and then click the **Advanced** tab. At the bottom of the page, you'll see the **Risk Parameters** with the **Target system risk weight** set to **2**.
- Navigate to **Intranet Portal** → **Accounts and Groups** → **General** → **Manager Portal** and click the **Operational** tab. At the bottom of the page, you'll see the **Risk Parameters** with the **Group risk weight** set to **3**.

- Look at the risk weights of the other target systems and their groups. We'll leave these values as they are.

### 2.17.3. Checking User Risk Parameters

As of now, risk values have not been calculated and the risk parameters of the user entries are empty. We can see this using DirX Identity Manager's Provisioning view in the sample domain:

- In **Provisioning**, select **Users** → **My-Company** → **Global IT** → and click **Pitton Lavina** to select it.
- Click the **Risk Parameters** tab to open it. You can see that all fields in this tab are empty.
- Look at the **Risk Parameters** tabs of other users. All the fields in each tab are empty.

Now we'll run the Risk Calculation workflow for the first time:

- Log in to DirX Identity's **Connectivity** view group and then select the **Expert View**.
- Navigate to **Connectivity Configuration Data** → **Workflows** → **My-Company** → **Main** → **Identity Store** → **RiskGvnContoller**. Right-click this entry and then select **Run Workflow**.

Now go back to the **Provisioning** → **Users** view in the sample domain to look at the user entries again:

- In **Provisioning**, select **Users** → **My-Company** → **Global IT** → and click **Pitton Lavina** to select it.
- Click the **Risk Parameters** tab to open it. You can see that the fields are now populated and the **Risk Level** is set according to the **Compound score**, which is calculated from the **Risk Factors** and the defined limits in the risk policy.
- Look at the risk parameters of other users, especially **Taspatch Nik** and **Wagner Retha**. All the fields in their **Risk Parameters** tabs are now filled.

Now switch to the Web Center and log in as **Taspatch Nik**:

- Open **Self Service** → **Display summary**. You see the risk displayed as a circle filled in red. If you move the mouse over the circle, you can see the tool tip High.
- Now choose **Users** → **Select user** and then click **Search**.
- Step through the different result pages. You will find that most of the users have a low risk value (indicated with a circle outlined in green) and **Wagner Retha** at the last page with a medium risk level (indicated with a yellow half-circle).
- Log out of Web Center and return to the DirX Identity Manager.

### 2.17.4. Changing a Target System Weight

Now we'll change the risk weight of the DirXmetaRole target system in the Provisioning view of the sample domain and then recalculate the risk levels in the Connectivity view:

- In **Provisioning** → **Target Systems**, click **DirXmetaRole**.

- In the **Advanced** tab, change **Target system risk weight** from **8** to **1**.

Now switch to the **Expert View** in the **Connectivity** view group and then run the risk calculation workflow again:

- Navigate to **Connectivity Configuration Data** → **Workflows** → **My-Company** → **Main** → **Identity Store** → **RiskGvnController**.
- Right-click the **RiskGvnController** entry and then select **Run Workflow**.

Now look again at the user entries in the **Provisioning** → **Users** view:

- In **Provisioning**, select **Users** → **My-Company** → **Global IT** → and then click **Taspatch Nik** to select it.
- Click the **Risk Parameters** tab to open it. You can see that his **Risk Level** has changed to **Low**.
- Look at the risk parameters of the other users, especially **Pitton Lavina** and **Wagner Retha**. All **Risk Levels** are now set to Low.

Switch to the Web Center and log in as **Taspatch Nik**:

- Open **Self Service** → **Display summary**. You see the risk displayed as **Low**.
- Now choose **Users** → **Select user** and then click **Search**.
- Step through the different result pages. You will find that mostly all of the users have a low risk value.
- Log out from the Web Center and return to the DirX Identity Manager.

Now we'll use the **Provisioning** → **Target Systems** view in the sample domain to return the **DirXmetaRole** target system's risk weight to its original value:

- In **Provisioning**, select **Target Systems** and then click on **DirXmetarole** to select it.
- In the **Advanced** tab, change **Target system risk weight** from **1** to **8**.

## 2.17.5. Changing the Risk Limits in the Risk Policy

In this step, we'll change the risk policy' risk limits and then recalculate risk using the new risk limits:

- In **Provisioning**, select **Policies** → **Risk Policies** → **Risk Policy**.
- Change the **Upper Risk limit** from **3.5** to **1.5** and change the **Lower Risk limit** from **1.5** to **0**.

Now let's run the Risk Calculation workflow again:

- In the **Connectivity** view group, select the **Expert View**.
- Navigate to **Connectivity Configuration Data** → **Workflows** → **My-Company** → **Main** → **Identity Store**.
- Right-click **RiskGvnController** and then select **Run Workflow**.

Now switch to Web Center and log in as **Taspatch Nik**:

- Choose **Users** → **Select user** and then click **Search**.
- Step through the different result pages. You will now find a lot of users with medium risk level (for example, **Pitton Lavina**) and some of the users with high risk level (for example,. **Wagner Retha**).
- Log out from Web Center and return to DirX Identity Manager.

Now we'll return the risk limits of the risk policy's risk limits to their original values:

- In the **Provisioning** view, navigate to **Policies** → **Risk Policies** → **Risk Policy**.
- Change the **Upper Risk** limit from **1.5** back to **3.5** and the **Lower Risk limit** from **0** back to **1.5**.

Now we'll run the Risk Calculation workflow again:

- In **Connectivity** → **Expert** view, navigate to **Connectivity Configuration Data** → **Workflows** → **My-Company** → **Main** → **Identity Store**.
- Right-click **RiskGvnController** and then select **Run Workflow**.

## 2.17.6. Scheduling the Risk Calculation Workflow

It makes sense to set up a schedule for the Risk Calculation workflow on a regular basis, for example, once a day:

- In DirX Identity Manager's **Connectivity** view group, select the **Expert View**.
- Navigate to **Connectivity Configuration Data** → **Schedules** → **My-Company**.
- Right click on the **My-Company** entry and then select **New** → **Schedule**.
- In **Name**, specify **Risk Schedule**.
- Check **Active**.
- Choose the workflow **Workflows** → **My-Company** → **Main** → **Identity Store** → **RiskGvnController** by clicking on the icon button with the three dots.
- Click **OK**.

## 2.17.7. Configuring the Risk Approval Workflow

DirX Identity provides a template risk approval assignment workflow that enforces an extra approval step from the company head if a requested privilege assignment will result in a high risk level for the user. In this exercise, we'll demonstrate how to activate this workflow in the My-Company sample domain.

First, we need to copy the example workflow **RiskApproval** to the My-Company domain:

- In **Provisioning** → **Workflows**, navigate to **Workflows** → **Definitions** → **Default** → **Assignments.**
- Copy the **RiskApproval** entry to **Workflows** → **Definitions** → **My-Company** → **Approval** by

pressing and holding down the Strg-key and the left mouse button and moving the cursor with this workflow to the new location.

- Click on the new entry **Workflows** → **Definitions** → **My-Company** → **Approval** → **RiskApproval**.
- In the **Workflow** tab, check Active to activate the workflow.
- Open the **When Applicable** tab. You can see that the **Priority** is set to **99**, which is now the highest priority of all activated approval workflows. As a result, this workflow will be started for the next privilege approval.
- Restart the Java-based Server and Apache Tomcat for the Web Center for the changes to take effect.

We want our risk approval workflow to flag for additional approval a risk level change from medium to high. Let's look at a user's current risk score:

- In the **Provisioning** → **Users** view, navigate to **Users** → **My-Company** → **Finances** → **Dalmar Christopher**.
- In the **Risk Parameters** tab, look for the **Compound score** value. This value should be at about **2** and the **Risk Level** should be **Medium**.

Next, we'll set the risk limit for high risk in our risk policy to a value that is a little bit higher than the Compound score of **Dalmar Christopher**:

- In the **Provisioning** → **Policy** view, navigate to **Policies** → **Risk Policies** → **Risk Policy**.
- Change the **Upper Risk limit** from **3.5** to **2.1**.

Now we'll run the risk calculation workflow again:

- In **Connectivity** → **Expert View**, navigate to **Connectivity Configuration Data** → **Workflows** → **My-Company** → **Main** → **Identity Store**.
- Right-click **RiskGvnController** and then select **Run Workflow**.

Next, we'll use Web Center to assign a new role to **Dalmar Christopher**.

- Log in to Web Center as **Taspatch Nik**.
- Open **Users** → **Select user**.
- Enter **Dalmar** in the search field and then click **Search**. You see that **Dalmar Christopher** is still at a medium risk level.
- Now click **Dalmar Christopher** and then select **Users** → **Assign privileges**.
- Enter **DXR** in the search field and then click **Search**.
- Select **DXR Domain Administrator** and then move it from **Available roles** to the **Assigned roles**. Click **Save**.
- Select **Work list** → **Show initiated workflows**. You'll see the assignment workflow for **Dalmar Christopher**. Click on this entry.
- In **Workflow Details**, you can see that the Calculate Risk activity is finished, and **Wagner Retha** must approve this privilege assignment.

- Log out from Web Center and return to DirX Identity Manager.

Let's verify the current workflow status:

- In **Provisioning** → **Workflows**, navigate to **Workflows** → **Monitor** → **My-Company** → **Approval** → **RiskApproval** → *current date*.
- Click on the running workflow **Dalmar Christopher → DXR Domain Aministrator**.
- Open the **General** tab. You can see that the **Calculate Risk** activity is finished, and the activity **approve by Priv Manager** is the current step.

Now we need to approve the new privilege as the privilege manager **Wagner Retha**. We'll use Web Center for this task:

- Log to Web Center as **Wagner Retha**.
- Click the open task **approval by Privilege Managers**. You can see that the risk after approval would be high. Accept this approval and then logout from Web Center.
- Return to DirX Identity Manager's Provisioning → Workflows view and refresh the General workflow view. Now the activity **approve by Priv Manager** is finished and the activity **Approval by Company Head** is the current step. This additional activity is started because the resulting risk level is high. If it was medium or low, no additional approval activity would be started.

Next, we'll approve the new privilege as the head of the company:

- Return to the Web Center and log in as **Hungs Olivier**.
- Click the open task **Approval by Company Head (high risk)** and then accept this request.
- Log out from the Web Center.

Let's have a look at the results of this approval workflow:

- Return to the DirX Identity Manager's **Provisioning** → **Workflows** view and refresh the **General** tab for the risk approval workflow.
- Now the activities **Approval by Company Head** and **Apply Changes** are finished, and the privilege is successfully assigned.
- In **Provisioning** → **Users**, navigate to **Users** → **My-Company** → **Finances** → **Dalmar Christopher**.
- Open the **Assigned Roles** tab. You can see the role **DXR Domain Administrator** in the enabled state.
- Open the **Risk Parameters** tab and look at the **Compound score** value. This value has not changed and the risk level is still medium. The reason for this is that the approval workflow does not recalculate the risk levels. The Risk Calculation workflow needs to run to recalculate the risk.

Let's run the Risk Calculation workflow again and then re-check Dalmar Christopher:

- In **Connectivity** → **Expert View**, navigate to **Connectivity Configuration Data** →

**Workflows** → **My-Company** → **Main** → **Identity Store**.

- Right-click **RiskGvnController** and then select **Run Workflow**.
- Return to the user entry **Dalmar Christopher** in **Provisioning** → **Users**. Refresh the **Risk Parameters** tab and look at the **Compound score** value. This value is now higher than **2.1** and the **Risk Level** has been changed to **High**.

# 3. About the My-Company Sample Domain

The My-Company sample domain is a realistic scenario of a small company that demonstrates most of the features that DirX Identity provides. It comprises many DirX Identity objects, such as users, departments, privileges, policies, request workflows and target systems. The My-Company sample domain is designed to be the basis for the quick start tutorial. You can select the My-Company sample domain during DirX Identity configuration. You can also load the Connectivity scenarios that correspond to the My-Company domain by hand.

My-Company represents a small, international industrial company called My-Company in a business-to-business (B2B) scenario, where additional companies act as customers and suppliers. My-Company designs and produces hardware and software products made available to end customers through re-seller companies who buy My-Company's products and sell them in their malls and stores. Supplier companies deliver the hardware and software that My-Company needs in order to build its products.

This chapter describes the objects in the My-Company sample domain and the relationships between them. We recommend that you read the descriptions given here while you look up the objects in the DirX Identity views.

## 3.1. Logging In

To log in to the My-Company domain, follow the procedure described in "Logging In" in the "Getting Started" chapter.

## 3.2. Users

Click the **Users** view. For the sample domain, this view presents a directory tree organized into user subtrees for three companies: Customers, Suppliers, and the employees of My-Company. The user tree also contains the standard query folders provided for the user view and a System folder that is not currently used.

The companies that make up the sample domain scenario are distributed throughout Europe and the U.S.A., as illustrated in the following figure.

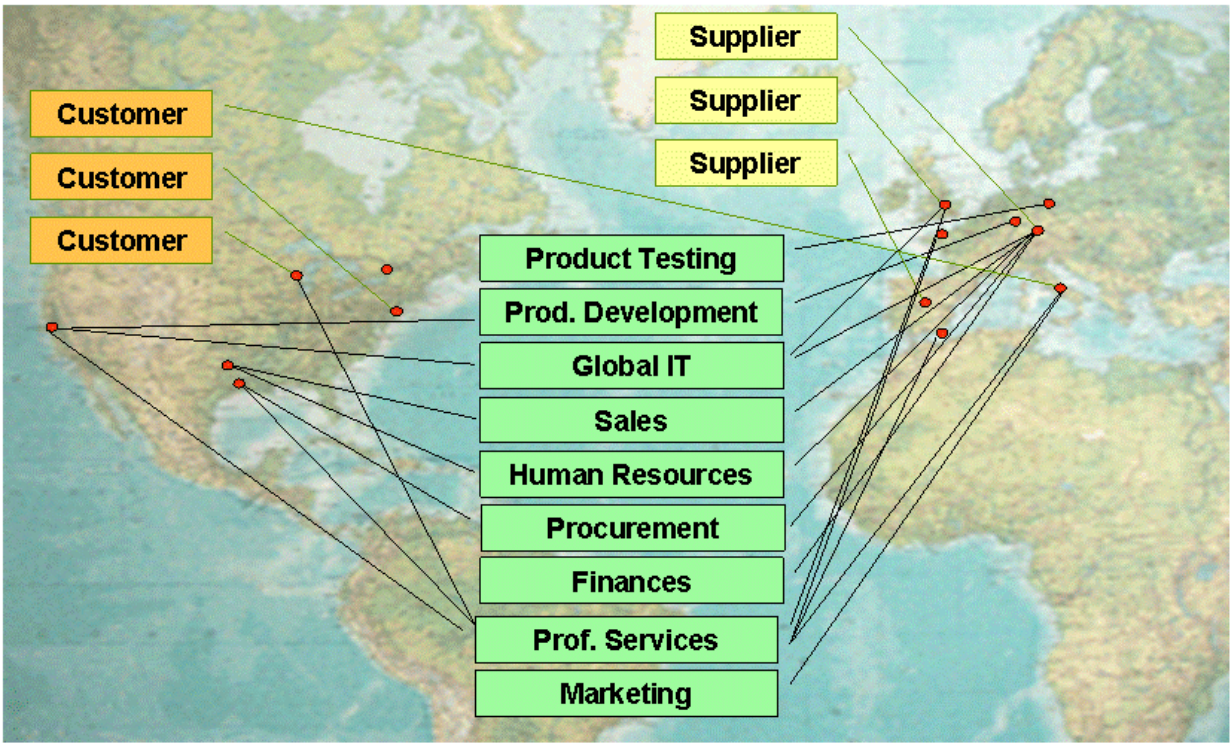



*Figure 18. DirX Identity Sample Domain Scenario*

### 3.2.1. My-Company

Open the My-Company subtree to see the organizational units that My-Company contains. In the sample domain scenario, organizational units are designed as tree objects. Also check the content and structure of the departments that are either lower-level organizational units or user entries. The organizational units are:

- **General Management (GM)** - this organizational unit is not modeled as a department. It consists of the general manager **Olivier Hungs** and his assistant **Gabriela Morton**, who manage all the other departments with about 60 employees. Both persons reside in Munich, Germany.
- **Finances (FI)** - this department, managed by **Christopher Dalmar**, is responsible for finances, cost control and auditing. It is also located in Munich, Germany.
- **Global IT (IT)** - this department handles most of the administrative tasks for the company, including the management of DirX Identity itself and most of the target systems. Its manager is **Nik Taspatch**, who resides in San Jose, California, U.S.A. The other people in this department are located in Europe.
- **Human Resources (HR)** - this department handles personnel issues. **Hans Berner** controls his department members in Europe and the U.S.A.
- **Marketing (MA)** - this department handles tasks like public relations, fairs and conferences and is located in Rome, Italy. Its manager is **Gianfranco Benetton**.
- **Procurement (PR)** - this department, managed by **Henry Filler**, buys the hardware and software components to build the final products and also buys computers and machines that are necessary for the company infrastructure. This department also controls the assembly of the components in the sub-contractor company **CompuTip** in

Munich.

- **Product Development (PD)** - this organizational unit consists of two sub-departments for hardware and software development that are managed by **Frederic Duplan**, **Veronique Cohu** and **Bill Sedran**. Hardware development is located in Frankfurt, Germany and software development resides in San Jose.
- **Product Testing (PT)** - this department performs integration and system tests on the final products. It is located in Berlin, Germany and is managed by **Klaus Reichel**.
- **Professional Services (PS)** - this organizational unit is distributed all over the world and performs pre- and post-sales. **Lionel Bellanger** manages it.
- **Sales (SA)** - this organizational unit consists of two sub-departments in Europe and the U.S.A. **Hatty Straub** and her assistant **Ruben Briner** manage Sales. **Sven Richter** and **Guo-Qiang Brailey** manage the subunits.

### 3.2.2. Suppliers

Three companies act as suppliers for My-Company:

- **MicroWare** (London, Great Britain) - produces the necessary hardware components
- **MediaComp** (Madrid, Spain) - delivers software components
- **CompuTip** (Munich, Germany) - assembles the hardware and software components to build the final products

### 3.2.3. Customers

Three re-seller companies are My-Company's customers:

- **Mercato Aurum** (Rome, Italy) - the distributor for My-Company products in Italy
- **MultiMarket** (Chicago, USA) - the distributor for My-Company products in the U.S.A.
- **TakeAway** (New York, USA) - the market for end customers in Great Britain

### 3.2.4. User Properties

Click a user in one of My-Company's departments. You'll see two rows of tabs that group different user properties and attributes according to their functions. These tabs and fields represent the default user definition for the My-Company sample domain. This user definition has been designed to provide a comprehensive set of user properties and attributes that you are, with some exceptions, allowed to change or even remove from the user definition if they do not meet your needs.

The default user definition is modeled as a set of object descriptions in XML. The definitions of user properties and attributes that you are not allowed to modify or delete can be found in the **Domain Configuration** view at **My-Company → Object Descriptions → User.xml**. If you go to **Domain Configuration → My-Company → Customer Extensions → Object Descriptions**, you will find two default object description files: **UserCommon.xml** and **User.xml**. As delivered with DirX Identity, **User.xml** is empty and is intended for your use. **UserCommon.xml** contains the sample set of user properties and attributes that you can modify or delete as necessary. You can also use the object description paradigm to create

your own custom user object descriptions. For example, you may want to define several different user types that display different properties and attributes depending on the type of user, such as "internal" user and "contractor". For more information about object descriptions and how to work with them, see the section on "Customizing Objects" in the *DirX Identity Customization Guide*.

The next sections briefly describe the user properties and attributes provided with the default user definition for the My-Company sample domain and gives some hints on what you can do with them. The context-sensitive help available on each tab (just click **Help**) provides complete details on each field.

#### 3.2.4.1. General User Properties

The General tab contains properties that identify the user. The My-Company sample domain uses the following properties in this tab:

- The "name" properties **Name, First Name, Middle Name, Last Name**.
- **Salutation** (the sample domain supplies salutations in a variety of languages), **Day of Birth**, **Title** (for example, **Dr.**) and **Gender** (the sample domain supplies the selections **Male**, **Female**, **Neutral**).
- The "identifier" properties **Employee Number** and **Master**. **Employee Number** (employeeNumber) is a customer-specific identifier that comes, for example, from a human resources system. **Master** (dxmOprMaster) is a directory-specific value that identifies the source of the user entry (it is the connected directory's **Master Name** operational attribute, written to the user entry by the connected directory's source workflow). This feature is intended for use in multiple master scenarios to protect user entries mastered by one directory source from being created or deleted by another master (modification by different masters is permitted).

  The My-Company domain has two sources of user entries: the HR-ODBC connected directory (which is already set up in the sample Connectivity scenario delivered with the sample domain), and the New-HR connected directory, which is set up as part of the tutorial exercise "Importing Identities" in the "Getting Started" section of the guide. The source workflows for these two directories are configured to use the **Master Name** operational attributes (the **Use Operational Attributes** field is checked in the workflow configurations). As a result, a user in the My-Company sample domain will either have the value **HR** to indicate that the HR-ODBC directory is the source of the entry, or the value **NEWHR**, which indicates that the New-HR directory is the source.

  All the default connected directories provided with DirX Identity have their **Master Name** operational attribute set to **HR**. If you intend to set up a multi-master scenario and use one or more of the default connected directories as templates, you should make sure you assign a different **Master Name** to your different master connected directories and set the **Use Operational Attributes** flag in the source workflows for these connected directories. For more information on creating connected directories, see "Managing Connected Directories" in the *DirX Identity Connectivity Administration Guide*. For more information on source workflows, see the section "Using the Source Workflows" in the *DirX Identity Connectivity Administration Guide*.
- **Description,** which briefly describes the person's tasks. A description is useful for

providing a human-readable way to identify a user when it is part of an extensive table-based display.

- **Employee Type (employeeType)**, which can be used to classify users. The My-Company sample domain defines the following employee types:
  - **Internal** - all employees of My-Company
  - **Contractor** - all contractors that work for My-Company
  - **Customers** - customers of My-Company
  - **Suppliers** - suppliers of My-Company
- **Business Category**, which identifies the type of business the user works for. The My-Company sample domain defines business categories for **Banking, Energy, Healthcare, Industry, Real Estate, and Transportation**. It assigns **Industry** to the My-Company employees, but does not assign a business category to employees in Customers and Suppliers.

The default General user properties tab also provides an **Identifier** property (dxmGUID) that customers can use to set up a global unique identifier (GUID) for each user. To use this property, the customer designs a unique identifier schema and then writes its own GUID generator to create the GUIDs according to the schema. The default connected directories delivered with DirX Identity implement a sample GUID schema and generator that generates a local GUID for a user from a fixed string prefix configured in each connected directory and the user's Employee Number, if the feature is activated in the corresponding source workflows. For more information, see the context-sensitive help on connected directories and workflows and the section "Understanding the Tcl-based Source Workflows" in the *DirX Identity Application Development Guide*. The My-Company sample domain does not use the **Identifier** property.

#### 3.2.4.2. Relationships to Other Users

The **Relationships** tab specifies the relationships that a user has to other users. The My-Company sample domain defines five types of user-to-user relationship:

- Owner - the administrator who can manage this user, or the person responsible for this user, if it is a functional user, for example, the representation of a hotline or a service. My-Company does not use this field.
- Manager - the user's manager; the My-Company sample domain uses this relationship for its Internal employee types
- Secretary - the user's secretary or administrative assistant
- Representative - the user's representative; that is, the person who can perform the user's functions if he or she is unavailable
- Sponsor - the user's sponsor; My-Company uses this relationship for its Contractor employee types

The values specified in these fields are links to other users. To see an example of this setup, go to **Users → My-Company → Human Resources** and then click on **Berner Hans**, the manager of the Human Resources department. You can see that he has a Manager link (Olivier Hungs, who is the manager of My-Company), a Secretary link (Laura Telfer) and a

Representative link (Mary-Jane Ormsby), but no Sponsor link because he is not a contractor.

Now click on **Strober Santiago**. His Manager is Olivier Hungs and his Sponsor is Hans Berner. Although Santiago Strober reports to Olivier Hungs the general manager and not to a department manager, he needs to have a department sponsor because My-Company contractors are required to have sponsors in their functional areas. Because he currently works in Human Resources, he is assigned the sponsor Hans Berner, the HR manager. This example demonstrates that you can set up any kind of relationship hierarchy, not just one that follows your managerial hierarchy.

These relationships can be useful in, for example, approval workflows to dynamically search for specific persons. For example, you could set up an approval workflow that searches for both the user and the user's representative so that you have both persons in your approval activity. If the user is not available, the representative has the right to approve. For more information on user-to-user linking, see the section "Working with Links at User Entries" in the section "Managing Users" in the *DirX Identity Provisioning Administration Guide*.

You can also use the relationships between users in conjunction with variable substitution to implement dynamic participant calculation in your request workflows. For more information about this concept, see the section "Participant Calculation" in the path "Understanding the Default Application Workflow Technology" → "Understanding Request Workflows" → "Customizing Request Workflows" in the *DirX Identity Application Development Guide*.

#### 3.2.4.3. Operational Information

A user's operational properties come directly from the DirX Identity system itself. Customers are allowed to change some of these properties, for example the end date of a user or the deactivation period. These properties influence the user's status field. For details about these properties, see the online help. To understand DirX Identity's state handling mechanism, see Managing States. The User and Account Life-Cycle section is the most interesting for beginners.

#### 3.2.4.4. Communication Information

The **Communications** tab contains communications-related user attributes like email, office, mobile and FAX telephone numbers, and blog or other Web addresses. The My-Company sample domain uses all of these fields.

This tab also specifies the user's preferred language, which determines the language used in mail messages delivered by request workflows, if mail text in that language has been set up. The My-Company domain has set up English and German for request workflow mail messages. All users in My-Company have a preferred Language set. If this is one of English or German, the mail is sent in that language. Otherwise the respective default values are taken. The users in Suppliers and Customers do not have any preferred language set, so mail messages they receive from request workflows will be sent in the default language. For more information about nationalizing mail messages in request workflows, see the path "Understanding Default Application Workflow Technology" → "Understanding Request Workflows" → "Request Workflow Architecture → "Nationalizing Request Workflows" in the *DirX Identity Application Development Guide*.

#### 3.2.4.5. Authentication Information

A user's authentication properties come directly from the DirX Identity system itself and are only displayed for information. Customers can change the password policy; that is, they can specify the type of password policy that a user must follow, but they cannot change any of these other properties. For details about these properties, see the online help.

#### 3.2.4.6. Links to Organizations

The **Organization** tab displays the user's links to business objects modeled as organizations and organizational units. The companies that make up the sample domain have been structured as organization business objects and the My-Company departments have been structured as organizational units underneath the My-Company organization. See the "Companies" section in "Business Objects" for more information about these objects and how they are structured and used. The users in the My-Company part of the sample domain are linked to the My-Company organization business object and the organizational business unit that corresponds to the department in which they work.

To understand this relationship, let's look at a user. Go to **Users → My-Company → Human Resources** and click on **Berner Hans**, the manager of the Human Resources department. Click the **Organization** tab. In the **Organization** field, you can see that he has a link to the My-Company business object (if you click the icon to the right of **Organization**, the My-Company business object is displayed). In **Organizational Unit**, he has a link to the organizational unit that represents **Human Resources**, which is the department in which he works. The other users in the My-Company part of the user tree have similar assignments. The organization and organizational business objects set up for the My-Company users are used to master department-specific information and to automatically assign department-specific roles to each member of a department. For more information about this sample domain structure, see the "Business Objects" section. For more information on user-to-business object linking, see Working with Links at User Entries.

#### 3.2.4.7. Links to Locations

The **Location** tab displays the user's links to business objects modeled as locations. Location business objects mirror the countries and branch locations in which My-Company and its customers and suppliers are located; for more information, see the section "Countries". Each user in the sample domain is linked to the location that corresponds to the branch location of the company in which he or she works.

To understand this relationship, let's look again at Hans Berner. Go to **Users → My-Company → Human Resources** and click on **Berner Hans**, the manager of the Human Resources department. Click the **Location** tab. In the **Location** field, you can see that he has a link to the business object that represents My-Company's main office of its Human Resources department: My-Company Munich (if you click the icon to the right of **Location** , the **My-Company Munich** business object is displayed).

Now let's look at the Location tab of Mary-Jane Ormsby. She is linked to the business object that represents the My-Company's Dallas, Texas branch office of its Human Resources department: My-**Company Dallas**.

The other users in the sample domain Users tree have similar location links. The location business objects set up for the sample domain users are used to master location-specific information for the employees who belong to the branch office. For more information about this sample domain structure, see the "Business Objects" section. For more information on user-to-business object linking, see Working with Links at User Entries.

#### 3.2.4.8. Links to Contexts

The **Context** tab is a placeholder for linking users to customer-created objects. The My-Company sample domain does not use the fields in this tab. Instead, you use the **Custom** folder in the **Business Objects** tree to store the new objects you create, and then use the Context link field in this user property tab to link your users to these customer-specific business objects.

#### 3.2.4.9. User Privilege Assignments

The **Assigned Roles, Assigned Permissions** and **Assigned Groups** tabs show the roles, permissions and groups currently assigned to the user.

Go to **Users → My-Company → Finances** and then click **Dalmar Christopher**. Now click his **Assigned Roles** tab. Here you can see the different ways in which privileges can be assigned to users:

- By hand (through self-service or administrator action) - an administrator or Dalmar himself has assigned the roles **Manager, Project Manager, Project Member**, and **Signature Level 3**. The **manual** label in the **Assigned by** field indicates this "direct" type of assignment. You can read more about the My-Company sample domain's privilege structure in the section "Privileges". **Project Manager** and **Project Member** are examples of privileges that use role parameters, as indicated by the **Role parameters** column. You can read more about the My-Company sample domain's use of role parameters in the section "Project Organization".
- By inheritance from a business object - Dalmar has inherited the **Finances Tasks** role from his link to the **Finances** business object, which references this role. To see this relationship, click his **Organization** tab and then click the ··· icon to the right of **Organizational Units**: **Finances**. This action displays the **Finances** business object. Now click the **References** tab, and you can see **Finances Tasks** displayed in **Privileges**. The **BO** label in the **Assigned by** field indicates this type of assignment. You can read more about the My-Company sample domain's business object structure in the section "Business Objects".

  Note that the **Signature Level 3** role assignment also uses inheritance, in this case, by role hierarchy: the Signature Level 2 role is a junior role for **Signature Level 3**. To see this relationship, click the ··· icon to the right of this role, and then click the icon to the right of **Privileges**: **Signature Level 3** to open this role. Now click **Assigned Junior Roles** and you can see **Signature Level 2**. Because this role is specified as a junior role, a user who gets the **Signature Level 3** role automatically gets this one, too. For information about creating role hierarchies, see Creating a Role Hierarchy.
- By a provisioning rule - My-Company's **Internal Employees** rule assigns the **Internal**

**Employee** role to all users under the My-Company user subtree whose **Employee Type** field (see the **General** tab) is set to **Internal**. The **rule** label in the **Assigned by** field indicates this type of assignment. You can read more about My-Company provisioning rules in the section "Rules".

#### 3.2.4.10. Account Ownerships

The **Account** tab shows the accounts in the target systems that have been assigned to the user by the DirX Identity Provisioning process. Assigning a privilege creates the necessary accounts automatically with no manual interaction. The assignments and their properties come directly from the DirX Identity system itself and cannot be changed here. For details about these properties, see the online help. For more information on the target system and account structure used in the My-Company sample domain, see Target Systems. For more information on user-account linking, see Working with Links at User Entries.

#### 3.2.4.11. Order Information

Many DirX Identity objects have an orders tab for monitoring changes to the object that are currently pending. The **Orders** tab for users shows attribute and privilege assignment changes for this user that are currently pending. For example, the My-Company sample domain has an attribute policy that prohibits unauthorized changes to a My-Company user's organization and location values; you can read more about this policy in "Attribute Policies". Consequently, when a user's location or organization is changed, a request workflow starts automatically to get approval for the change. The **Orders** tab tracks this approval process. For information about the properties shown in this tab, see the online help.

#### 3.2.4.12. SoD Exceptions

The **SoD Exceptions** tab lists the current SoD violations for the user and whether or not they have been approved. For information about these properties, see the online help. For more information on the SoD policies set up for the My-Company sample domain, see SoD Policies.

## 3.3. Business Objects

DirX Identity provides a set of sample business object types organized in the following default folder structure in the Business Objects view:

- Companies (the container for organization and organizational unit business objects)
- Cost Locations (the container for contain cost location business objects)
- Countries (the container for country and location business objects)
- Custom (the container for context business objects that customers have created)
- Projects (the container for project business objects)

The My-Company sample domain uses the companies, countries and projects business object structures. It does not implement any cost location or context business objects. You are free to extend this sample business object tree with, for example, cost location objects

in the **Cost-Units** tree or with your own types of business objects (use the context business object type dxrContent) in the **Custom** tree.

The My-Company sample domain business object structure illustrates several ways to use business objects:

- To assign privileges referenced in the business objects automatically to all users linked to the business object; the "Companies" section describes the details.
- To master user attributes; the "Countries" section describes the details.
- As a source of role parameters; the "Projects" section describes the details.

The following figure shows an example of the relationship between a user and the business objects in the My-Company sample domain.

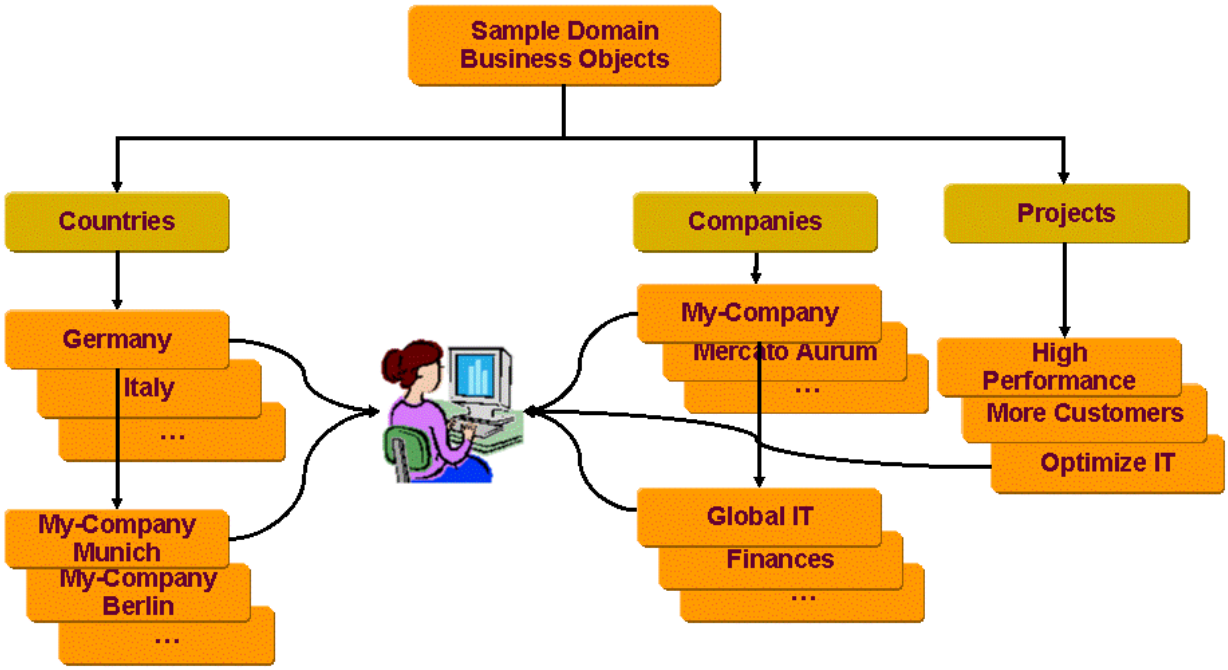



*Figure 19. Sample Domain Business Object Structure*

For more information on using business objects, see Managing Business Objects.

### 3.3.1. Companies

Click the **Companies** node in the **Business Objects** tree. You can see subfolders for the My-Company company and its customer companies Mercato Aurum, MultiMarket and TakeAway. Business objects for the sample domain supplier companies (CompuTip, MediaComp and MicroWare) have not been structured into this part of the business objects tree. These company subfolders represent organization business object types.

The My-Company organization is populated with subfolders for its departments (Finances, Sales, and so on). These subfolders represent organizational unit business object types and demonstrate automatic privilege assignment to users via business object inheritance. For example, click on **Finances**, and then click the **References** tab. In the **Privileges** field, you can see that this business object is linked to the **Finances Tasks** role in the Corporate Role

tree. To find this role, click **Privileges → Roles → Corporate Roles→ Department Specific**. The other department business objects are set up in a similar way: each department business object is linked to one or more roles that are relevant to this department, as follows:

- Human Resources is linked to the HR tasks role (see **Privileges → Roles → Corporate Roles→ Department Specific**)
- Information Technology is linked to the User Administrator role (see **Privileges → Roles → Corporate Roles→ Administration**)
- Marketing is linked to the Marketing Tasks role (see **Privileges → Roles → Corporate Roles→ Department Specific**)
- Procurement is linked to the Procurement Tasks role (see **Privileges → Roles → Corporate Roles→ Department Specific**)
- Product Development is linked to the HW Developer and SW Developer roles (see **Privileges → Roles → Corporate Roles→ Department Specific**)
- Product Testing is linked to the Testing Tasks role (see **Privileges → Roles → Corporate Roles→ Department Specific**)
- Professional Services and Sales are linked to the Sales Tasks role (see **Privileges → Roles → Corporate Roles→ Department Specific**)
- General Management has no links to privileges

Each My-Company user is linked to a "department" (organizational unit) business object and thus automatically inherits the role that is linked to this object. For example, go to **Users → My-Company → Sales** and then open **Klarmann Bruno**. Click the **Organization** tab.

You can see that this user belongs to the organizational unit **Sales**. If you click the icon to the right of the **Organizational Unit** field, the **Sales** business object is displayed. Click the **References** tab, and you can see that the role **Sales Tasks** is linked to this business object. Click **Close** and then click the **Assigned Roles** tab. You can see **Sales Tasks** listed as an assigned role, and the **Assigned by** field indicates that the role has been assigned by a business object (**BO**).

If you change a user's organizational unit, DirX Identity automatically unassigns the obsolete privileges and assigns the new ones that are relevant to his new organizational unit. For example, return to the **Organization** tab for **Klarmann Bruno** and then click the  icon to the right of the **Organization** field. The **Companies** business object tree is displayed. Select **Marketing** to move the user to the Marketing organization, and then click **Save**. This action automatically removes the Sales Tasks role and assigns the Marketing Tasks role to the user. Conversely, if you change the privileges that are linked to a business object or make changes to the linked privileges themselves, all users linked to the business object will automatically be updated with the new privilege assignments and/or the updated privilege.

Note that changing a user's organizational unit means changing his link to an OU business object in the Business Objects tree. Moving a user from one department to another in the Users tree has no effect; for example, if you move **Klarmann Bruno** from the **Users → My-Company → Sales → Sales Europe** node to the **Users → My-Company → Marketing** node, he remains linked to the Sales department until you change this business object link. The

sample domain has set up the My-Company User tree to mirror the structure of the Business Object tree, but any kind of Users tree structure is permissible. For example, you may want to create an "administrative area" structure, where users are grouped according to the administrator who is responsible for managing them.

While the **Companies → My-Company** "department" organizational unit business objects use only the privileges link, it is also possible to link business objects to other items, such as locations, cost units, categories, and contexts. If you want to create category or context objects, you must create them under the **Custom** node in the **Business Object** tree. For more information about business objects, see Managing Business Objects in *DirX Identity Provisioning Administration Guide*.

### 3.3.2. Countries

Click the **Countries** node in the **Business Objects** tree. You can see subfolders for each country in which My-Company, its customers and its suppliers are located: **CA** (Canada), **DE** (Germany), **ES** (Spain), **FR** (France), **GB** (England), **IT** (Italy), **US** (USA). These subfolders represent country business object types. Each country subfolder in turn contains subfolders for company branch locations - for example, My-Company Paris (FR), My-Company Barcelona (ES), MediaComp Madrid (ES) and so on. These subfolders represent location business object types.

The objects in the **Countries** folder demonstrate the inheritance of business object data into user attributes and serve as samples for data inheritance. The business object is linked to the user object and masters the business data. If you change the business object linked to the user (because, for example, the user moves to a different branch office), the user data is automatically updated with the information from the currently linked business object. This update is performed immediately during the save operation. Likewise, if you change data at the business object (for example, the branch office street address), the change is propagated to all users linked to this object. This update runs in the background and can take some time to take effect, especially when the number of assigned users is high.

In the sample domain, each user is linked to a location business object and thus automatically inherits the location information in this object. For example, open **Countries → DE → My-Company Munich** and look at the postal code, street and address information there. Now go to **Users → My-Company → Finances** and open **Abele Marc**. Click the **Location** tab. You can see that this user belongs to **My-Company Munich** and has the same information in his postal code, street and postal address attributes. If you click the icon to the right of the **Location** field, the **My-Company Munich** business object is displayed showing this same information. Now click the  icon to the right of the **Location** field. The **Countries** business object tree is displayed. Select **My-Company Berlin** and then click **OK**. Now you can see that Marc Abele's postal code, street and postal address information has changed to reflect the **My-Company Berlin** location.

Note that if you save this change (by clicking **Save**), an approval workflow is automatically started. This action occurs because there is an attribute access policy for My-Company users that protects against changing locations and organizations without manager approval (you can read more about this policy in "Attribute Policies") and the domain-level flag to enable this type of workflow is set (see Request Workflow Parameters).

The My-Company sample domain also contains a sample JavaScript routine in its user object description that calculates the user's country information from the value in his Location field (dxrLocationLink). So when a user's location changes, his country information is updated automatically, too. To see the elements of this feature, go to **Domain Configuration → My-Company → Customer Extensions → *Object Descriptions**. Click **UserCommon.xml**, and then scroll or search to the Locality tab definitions. Here you see the property definition for the country field (**c**). It uses the **CountryFromLocation** JavaScript, which you can find in **Domain Configuration → My-Company → JavaScripts**. The country information is stored in the **Country** proposal list, which you can find at **Domain Configuration → My-Company → Customer Extensions → Proposal Lists**. For more information on using JavaScripts and proposal lists in object descriptions, see the following sections in the *DirX Identity Customization Guide:*

- For JavaScripts, see the section "Using JavaScript Files" in the chapter "Customizing Programming Logic".
- For proposal lists, see the section "Specifying Proposal Lists in Property Descriptions" in the section "Properties and Property Elements" in the chapter "Customizing Objects"

### 3.3.3. Projects

Click the **Projects** node in the **Business Objects** tree. You can see three subfolders that correspond to the three optimization projects that My-Company runs:

- **HighPerformance** - a project for general internal process enhancements.
- **OptimizeIT** - a project for better IT structures.
- **MoreCustomers** - a project to enhance sales and marketing activities.

These subfolders represent project business object types. My-Company's project organization uses these business objects as the source for role parameters in user-project assignment: there is a **Project** role parameter that references these business objects, and there are **Project*Member** and **Project Manager** roles that use this role parameter:

- To view the role parameter, click **Domain Configuration → My-Company → Customer Extensions → RoleParams → My-Company** and open the **Project** role parameter. This parameter has been created to list all business objects in the **Project** tree for selection during a role assignment. If you look at the **DN** area in the **General** tab, you can see that it links to the **Business Objects → Projects** tree, and will thus display all business objects contained in this tree in a drop-down list (in Web Center and DirX Identity Manager) when a role that uses this role parameter is assigned to a user.
- To view the roles, go to **Privileges → Roles → Corporate Roles → Project Specific**. Open **Project Member** and then click the **Role Parameters** tab. If you place your cursor over the **Project** column, you can see that it points to the **Project** role parameter (the Tool Tip shows the role parameter's distinguished name (DN).). Consequently, when someone assigns this role to a user, he is asked (via drop-down list) which project the role is for - HighPerformance, OptimizeIT or MoreCustomers - the values that come from the Projects business objects tree.

The sample domain illustrates how to use business objects to populate role parameter selections; you can also use them to build proposal lists. You can also associate a proposal

list with a role parameter to create one list (using business objects or other objects as a source) that you can use for both role parameter selection and drop-down proposal lists. For more information on using business objects, see the section "Managing Business Objects" in the *DirX Identity Provisioning Administration Guide*. For more information about creating role parameters and proposal lists, see the section "Customizing Parameters" in the *DirX Identity Customization Guide*.

# 3.4. Privileges

The My-Company sample domain contains a set of sample privileges that allow for provisioning of the users in the related target systems structured into a folder hierarchy based first on privilege type, and then on categories of privilege that correspond to My-Company's business structure. Click the **Privileges** view. You will see **Roles**, **Permissions** and **Groups** folders. The next sections provide more information on how these folders are organized.

## 3.4.1. Roles

The **Roles** subtree is subdivided into a **B2B Roles** subtree and a **Corporate Roles** subtree. The B2B Roles subtree contains all roles for customers and suppliers, while the **Corporate *Roles** subtree contains all roles for the My-Company employees and contractors.

There are six **Customer** roles: all customers are classified as **Silver Customers**. Some are **Gold Customers** and only a few are **Platinum Customers**. These classes determine the bonus program. Another set of roles defines the services a customer can subscribe to: **Customer Newsletter, Hardware Beta Programs** and **Software Beta Programs**.

Supplier roles include a **Standard Class** for each supplier and, for suppliers that perform very well, a **First Class** role.

The **Corporate Roles** subtree contains another six folders. A collection of all administrative roles makes up the **Administration** folder. The **Department Specific** roles folder contains all roles that define tasks for the various organizations in My-Company. The **Project Specific** roles folder collects all roles necessary for project management. The **Physical Access** folder contains all roles that pertain to granting access to secured computer rooms in several My-Company branches. The **Self-Service** folder contains all roles that users can assign themselves via user self-service subscription. All other roles are collected under the **General** folder. Within this folder is a very complex role - **Cost Location Manager** - that is part of a specific use case; to read more about this example, see the section "Policies for Hierarchical Role Parameters" in the *DirX Identity Provisioning Administration Guide*.

The sample domain's privilege structure demonstrates how to group privileges so that access policies can easily be applied to them. For example, the **Roles → Corporate Roles → Self Service** folder contains all the roles that users can assign to themselves with Web Center self-service. Now go to **Policies → Access Policies → My-Company → Grant Policies** and then click on **Self Service**. This access policy controls which users (in this case, users with the employeeType "Internal") can assign to themselves these roles with self-service. If you click the **Resources** tab and look at the **Resources filter** values, you can see that the policy directs DirX Identity to apply this policy to the roles in the **Roles → Corporate Roles → Self Service** folder. If you subsequently add a new role to this folder, the Self Service access

policy is automatically applied to it. If you look at the **Resources** field, you can see additional physical access roles listed here. As a result, the policy will be applied to these roles explicitly, but any new role added to the physical access folder will not have the policy applied to it. So the sample domain's privilege structure shows, too, that you can keep privileges segregated from access policies and add them as necessary. You can read more about access policies in "Access Policies".

The sample domain also structures privileges that use role parameters into a separate tree; this is the **Project Specific** tree underneath **Corporate Roles**. You can read more about how role parameters are used in the My-Company sample domain in the section "Project Organization".

However, role managers are free to structure their privileges as they like. The My-Company domain is only an example of a privilege structure.

Now click a role, and then click the **Assigned Permissions** tab to view the attached permissions of the role. In the sample domain, the permissions are almost always symmetrical to the roles, which mean that each permission has exactly one corresponding role, with the following exceptions:

- The **Contractor** role consists of the **Contractor** permission and the **Restricted File Share** permission.
- The **Internal Employee** role uses the **Internal Employee**, the **Group File Share**, the **Standard Tools** and the **Accounting** permissions. This is an example of a complex role.
- The **Manager** role is a combination of the **Manager** permission and the **Signature Level 2** permission.
- The **Training Manager** role consists of the **Manager** and the **Trainer** permissions. The **Trainer** permission is inherited from the assigned **Trainer** junior role (check the **Assigned Junior Roles** tab). The **Trainer** value in the **Source** column in the **Assigned Permissions** tab indicates this role hierarchy.
- The **SW Developer Tasks** and **HW Developer Tasks** roles (these are department-specific roles) have a combination of a common **Development Tasks** permission and a hardware or software-specific permission (SW Developer or HW Developer). The **SW Developer Tasks** role also includes a **RACF Standard** permission that allows membership in standard RACF groups.
- The **Test Tasks** role (another department-specific role) has the **Test Tasks** and **RACF Standard** permissions.
- The **MVS Administrator** role (in Administration) has the **MVS Administrator** and **RACF Standard** permissions.

Click a role, and then click the **Users** tab to see who is assigned to the role.

The **Signature Level 3** role provides another example of a role hierarchy. Click this role in the **General** folder. It contains the **Signature Level 2** role as a junior role (click the **Assigned Junior Roles** tab). This means that an assignment of the **Signature Level 3** role includes the automatic assignment of the **Signature Level 2** role. You can also see that the **Signature Level 2** role is used by a senior role if you click the **Senior Roles** tab.

The roles in the **Physical Access** folder are examples of roles that are designed to be assigned manually. The follow-on tutorial "Using Manual Provisioning" demonstrates the use case for these roles. They are also examples of roles that are assigned via provisioning rules; which are described in more detail in the section on "Rules".

In this example, the following roles are set to **Requires approval**:

- All the administrator roles
- All the physical access roles
- The general roles **Manager**, **Signature Level 3** and **Trainer**
- The project-specific role **Project Manager**
- The self-service roles **Manager Analyst Relations, Parking Place - Munich** and **Corporate Credit Card**

Click the **Manager** role, and then click the **Approvals** tab. Here you can see that the **Manager Nomination** workflow has been directly assigned to process the approval of assignments of this role to users. This is an example of direct workflow selection, where the workflow is directly linked to the privilege. The **Project Manager** role also has a direct assignment to the **Manager Nomination** approval workflow.

Click the icon to the right of the **Assignment** field to display the **Manager Nomination** workflow. Now double-click the **Approval by Company Head** activity icon and then click the **Participants** tab. Here you can see that the general manager Olivier Hungs and his secretary Gabriela Morton have been explicitly specified as approvers (only one must approve). This is an example of a static participant definition. For more information about participant calculation, including how to use variable substitution and user attributes to create dynamic approval definitions, see the subsection "Participant Calculation" in the path "Understanding Request Workflows" →"Customizing Request Workflows" → "Using Variable Substitution" in the *DirX Identity Application Development Guide*.

The other roles that require approval do not use direct workflow assignment. Instead, they use rule-based workflow selection and configure the "When Applicable" rules in the My-Company approval workflow definitions so that DirX Identity's dynamic workflow selection algorithm always selects the **4-Eye Approval** workflow to process approvals for these roles. For example, go to the My-Company workflow definitions (**Workflows → Workflows → Definitions → My-Company → Approval**). Open the **4-Eye Approval** workflow and then look at it is **When Applicable** tab. The **Priority** parameter is set to a high number so that the selection algorithm will choose it first. If you look at the **When Applicable** tab of the other workflows - for example, the Manager Nomination workflow - you can see that the priorities are set very low to prevent them from being selected. The **4-Eye Approval** workflow works with approval policies to retrieve the manager of the user (the subject) and the owner of the privilege as approvers. Both of these users must approve the assignment.

You can read more about DirX Identity's assignment workflow selection mechanisms, the **4-Eye Approval** workflow, and approval policies in the following places:

- In the *DirX Identity Application Development Guide*, see "Understanding the Default Application Workflow Technology" → "Understanding Request Workflows" → "Request Workflow Architecture" → "Selecting Request Workflows" → "Assignment Workflow

Selection"

- In the *DirX Identity Application Development Guide*, see "Using Request Workflows" → "Understanding Assignment Workflows" → "How Approval Works" → "4-Eye Approval"
- In the *DirX Identity Provisioning Administration Guide*, see "Managing Policies" → "Delegated Administration" → "Managing Access Policies" → "Policies for Approvals"

### 3.4.2. Permissions

The **Permissions** subtree is subdivided into the same folders as the role subtree. It contains a **B2B Permissions** subtree and a **Corporate Permissions** subtree. The **B2B Permissions** subtree contains all permissions for customers and suppliers, while the **Corporate Permissions** subtree contains all permissions for the My-Company employees and contractors.

Click a permission, and then click the **Assigned Groups** tab to view the attached groups. Most of the permissions use groups in a direct relationship, but some permissions use match rules to select some of the attached groups:

- The **Windows Administrator** permission uses two **Administrator** groups from the Windows target systems. The match rule selects the correct group based on the country attribute (c) of the user. View the groups to see the corresponding settings for the c attribute.
- The **HR Tasks** permission contains four groups.The **HR Access** and **HR Portal** groups are always assigned (the country attribute is set to "*"), the **SAP R3 Client** group is selected via the match rule. The **Marketing Tasks** permission structure is similar.
- The **Sales Task** permission is more complex. The **CRM Access** and **Sales Portal** groups are always assigned (the country attribute is set to "*"), the **SAP R3 Client** and **FS Sales** group are selected via the match rule.
- The **Group File Share** permission is the most complex. It represents a combined access right that allows access to the department-specific file share (attribute ou). It also selects the correct file share based on the country in which the employee resides (attribute c). See the **Match Rule** tab for the rule.In the **Assigned*Groups** tab, you will see nine file share groups. Only **FS Human Resources** and **FS Sales** are available in Europe and the USA. View the groups for the corresponding settings of the c and ou attributes.

Approval for the corporate permissions follows the same configuration as for the corporate roles; see the "Roles" section for more information.

### 3.4.3. Groups

The **Groups** folder in the **Privileges** view is a virtual view of the group folders in the target systems in the **Target Systems** view (see the "Target Systems" section for details). The folder is structured according to target system, and is displayed in the **Privileges** view to provide a single view of the privileges contained in a domain.

# 3.5. Policies

Click the **Policies** view. Here you can see three different subtrees:

- The **_Queries** subtree, which provides a set of default query folders for checking on active and inactive policies, operations, rules and delegations. For more information about queries, see the section "Creating a Query Folder" in the *DirX Identity Customization Guide*.
- The **Policies** subtree, which consists of a **_Queries** folder and a set of folders for Access Policies, Attribute Policies, Delete Policies, Event Policies, Operations, Password Policies, Rules and SoD Policies.Each of these subfolders contains a set of default policies and rules and a set of sample domain-specific policies in the folder My-Company. The next sections describe these folders in more detail. The **_Queries** folder provides a set of default query folders for checking on active and inactive policies, operations, rules and delegations. For more information about queries, see the section "Creating a Query Folder" in the *DirX Identity Customization Guide*.
- The **Delegations** subtree, which is initially empty and is the container for objects that relate to delegated administration based on access policies: the process of assigning one's access rights to DirX Identity data (or a subset of these rights) to someone else, optionally for a specified period of time. An administrator can delegate the access rights he has to manage users and privileges, assign privileges to users or approve requests for privilege assignments to another user or administrator via the DirX Identity Web Center. For more information, see the section "Delegated Administration" in the *DirX Identity Provisioning Administration Guide*.

Note that the sample domain-specific access policies are all delegatable to other users (their **Is delegatable** flag is checked), while the default access policies are initially not delegatable. The reason for this setup is that delegation impacts DirX Identity performance when it must retrieve a large number of delegatable resources.The sample domain has a small user community and a small set of business objects, so delegating all the access rights controlled by the sample domain's access policies does not affect performance. When setting up your access policies, we recommend that you restrict the number of access policies you make delegatable to protect your system's performance. Decide which access rights are really critical and should be delegatable, and keep that number small. For more information about setting up access policies, see the subsection "Guidelines for Access Policy Setup" in the "Managing Policies" section of the *DirX Identity Provisioning Administration Guide*.

## 3.5.1. Access Policies

Access policies control self-service and delegated administrative access to DirX Identity's resources (user and privilege data). An access policy defines a set of access rights to almost any object.

There are two types of access policies: the type that restricts operations on specific objects - for example, "modify user" - and the type that restricts access to specific Web Center menus and menu items. To be able to work on an object, you must have the right to view and use the menu item that relates to managing the object and you must be allowed to manage the object; both of these access policy types must be set up correctly. For example,

you can have the right to use the "modify user" menu, but if you are not allowed to manage any users, your rights to the menu are meaningless. Conversely, you can have the right to modify ten users, but if you are not allowed to use the "modify user" menu, you cannot search for these users.

In the **Policies** view, click the **Access Policies** node in the tree. You can see three subtrees:

- The **_Queries** folder. This folder contains a set of default query folders for access policies. The default queries here can help you to answer specific questions about access policies; for example, "Which access policies are active?" or "Which access policies are available for Users?". You can also define your own query folders either here or in your own domain-specific area to set up special queries for access policies.
- The **Default** folder. This folder is present in any domain. The access policies here provide basic access rights to DirX Identity user and privilege data for users and administrators. Target system groups (see the DirXmetaRole target system) are used to categorize administrator types; as delivered, the only member of each the group is the domain administrator (DomainAdmin). The policies in this folder are organized according to each object type: Accounts, Groups, Menus, Password Policies, Request Workflows and so on. Open some of these subtrees and then click on some of the access policies to read about what they do.
- The **My-Company** folder. This folder contains additional access policies set up for the My-Company sample domain. These policies are organized into sub-folders that correspond to the operation they control: Approval, Create, Grant, Read and Modify, and so on. The remainder of this section describes these policies in more detail. For more details on access policies and how to use them, see the "Managing Policies" section of the *DirX Identity Provisioning Administration Guide*.

My-Company Approval Policies

The policies in **Approval Policies** are copies of the DirX Identity default policies for approvals that have been activated for use (the **Is active** flag is set). These policies are explained in more detail in the "Approval Policies" section in "Request Workflows".

My-Company Create Policies

The **Create Policies** folder contains one access policy - **Managers create roles** - which allows all department managers - that is, all users that are members of the **allDepartmentManagers** group - to create new corporate roles. Open the policy. In the **Operations and Object Type** part of the **General** tab, you can see that the **create** operation is checked and that the object to be created is a role (**dxrRole**). Click the **Subjects** tab. The **Group of Persons** field specifies the group name **allDepartmentManagers**, which is defined in **Target Systems → DirXmetaRole → Groups → Templates**. If you go to this location and open the group, and then click the **Members** tab, you can see the users in the sample domain that are assigned to this group. In **Policies → Create Policies**, open the policy again, and then click the **Resources** tab. You can see that it is set up to look for roles in the path **Privileges → Roles → Corporate Roles** (the notation in the field gives the LDAP directory path to the node in the role tree).

My-Company Grant Policies

The **Grant Policies** folder supplies the following access policies:

- **Cost location admins**, **cost location managers**, and **cost location manager role** - illustrate how to use access policies to restrict the assignment of hierarchical role parameters. For details about this feature and this sample implementation, see the section "Policies for Hierarchical Role Parameters" in the "Managing Policies" section of the *DirX Identity Provisioning Administration Guide*.
- **Customer self service** - enables the following types of access to the roles in **Privileges → B2B → Customers → Customer Services:**
- Unregistered users can subscribe to these roles during self-service registration (by clicking **Register** in Web Center).
- Registered logged-in users can subscribe to these roles at any time (by selecting **Subscribe privileges** in Web Center's **Self Service** menu).
- Users in the Sales department can assign these roles to other users, for example, to customers.
- **Customers can request self service roles** - enables any employee of My-Company's customers to assign himself the roles in the **Customer Services** role folder. (Note that customers already have this right as the result of the **Customer self-service** policy, so this policy is not really necessary).
- **Project manager grants privileges** - enables all users that are members of the **allProjectManagers** group (defined in **Target Systems → DirXmetaRole → Groups → Templates**) to assign all project-specific roles to all users.
- **Resource manager grants privileges** - enables all users who are referenced in the **Owner** field of a privilege to assign the privilege to users.
- **Self service** - enables all internal employees of My-Company to assign the roles in **Privileges → Roles → Corporate Roles → Self Service** to themselves. The self service feature of the Web Center uses this policy; selecting **Subscribe privileges** in Web Center's **Self Service** menu will list these roles as available for self-assignment.

My-Company Menu Policies

The My-Company menu access policies control the selections on the Web Center main menu bar (Self-Service, Delegation, Work List and so on) that are available to different categories of My-Company users. These policies are:

- All My-Company employees can use the **Delegation**, **Self service** and **Work list** menus in Web Center. These policies define the basic access rights of users in the sample domain. Open the policy **Users have the Self service menu**. In the **General** tab, you can see that the operation is **execute** and the protected object is a set of menu items (**dxrMenuItems**). If you go to **Domain Configuration → My-Company → Proposal Lists → Menus** and then click on **Self Service**, you can see this menu definition. In the **Subjects** tab, you can see that the policy operates on a search for users in the My-Company part of the **Users** tree (the **Persons Filter** fields). In the **Resources** tab, you can see the set of menu items to which the policy applies: the **Resources** field provides the DN of the Self Services menu in **Proposal Lists**. The other two policies have a similar setup.
- All My-Company employees in the Human Resources and Sales departments and all

managers can use the user management menu. Open the **Managers have the User Management** menu policy. The operation (execute) and the protected object (menu items) are the same. The **Subjects** tab shows that the policy applies to a defined group - the **allDepartmentManagers** group defined in the DirXmetaRole target system (see **Target Systems → DirXmetaRole → Groups → Templates**). The **Resources** tab gives the DN to the set of menu items that correspond to the **Users** selection in Web Center.

- All My-Company managers can use the activity management and role management menus. These two policies also apply to the members of the **allDepartmentManagers** group and gives them the right to use the **Role** menu and context-sensitive selections in the **Task list** menu, like delegating an incomplete task to another user.

The access rights granted to a particular user are thus an aggregation of the different access policies: for example, a "normal" user has basic rights, a user in the HR department has additional rights, and a manager in the HR department has further rights.

These policies have been activated for the My Company domain (the **Enable menu policies** flag is checked in **Domain Configuration → My-Company → Policies**).

My-Company Password Policies

The My-Company password access policies control who can set and read user and target system account passwords. These policies are:

- **Users handle passwords of their accounts** - enables all users in the My-Company subtree of the Users tree to work with the passwords of their accounts.
- **Users handle their passwords** - enables all users in the My-Company subtree of the Users tree to read and set their passwords.

The **setPassword** operation means that the user can create and change the password, and the **readPassword** operation means that the user can see the value of the password if it is displayed in Web Center. The **readPassword** operation is intended for use with passwords of privileged accounts, which are functional accounts like "root" in a UNIX target system or "Administrator" in a Windows target system and are typically assigned to multiple users: When one user is removed from a privileged account, DirX Identity automatically changes the password and then propagates the new password to the target system. The other users who are still assigned to the account must then be able to look up and read the new password. The **readPassword** operation is intended for this use case when the sample domain configuration parameter **Enable privileged accounts** is set (see the **Privilege Resolution** tab at the domain). You can read more about privileged accounts in "Managing Target Systems" → "Managing Target System Accounts" → "Managing Privileged Accounts" in the *DirX Identity Provisioning Administration Guide*.

My-Company Read and Modify Policies

The access policies in the **Read and Modify Policies** folder are:

- **Anyone can read locations** - enables unregistered users to read the properties of Location business objects in the **Business Objects** tree. This policy allows unregistered users to be able to set their Location attributes during self-registration.
- **GM can read all users** - enables the employees in the General Management

department (Olivier Hungs and Gabriella Morton; see the **Subjects** tab) to read the properties of all users in the sample domain: My-Company, Customers, and Suppliers.

- **Group owners can handle their groups** - enables the owner of a target system group to read and modify it (look at the matching rule in the **Rules** tab; it locates the user whose DN matches the DN set in a group's owner attribute). For example, go to **Target Systems → Intranet Portal** and click on **HR Portal**. The owner is **Berner Hans**. The access policy allows Hans Berner to read and modify this group. This policy is an example of a rule to achieve dynamic resolution of an access policy. For more information, open the policy, click the **Rules** tab, and then click **Help**.
- **HR can read all employees** - enables all users in the Human Resources department to read all of the properties of all My-Company users.
- **Managers modify their employees** - enables My-Company managers to read and modify the properties of My-Company users who report to them. This policy locates the user whose DN matches the DN set in a user's manager attribute. It is an example of how to use "relationship" user links to achieve dynamic resolution of an access policy; see the section "Relationships to other Users" for more information.
- **Permission owners can handle their permissions** - enables the owners of permissions to read and modify them.
- **PR can handle suppliers** - enables all users in the My-Company Procurement department (PR) to read and modify the properties of users in the Suppliers part of the Users tree.
- **Project managers modify all employees** - enables the members of the **allProjectManagers** group (see **Target Systems → DirXmetaRole → Groups → Templates**) to read and modify the properties of users in the My-Company part of the Users tree. This policy allows project managers to assign privileges to project members (the read operation here permits this task).
- **Role owners can handle their roles** - enables the owners of roles to read and modify them.
- **SA and PS can handle customers** - enables all users in the My-Company Sales (SA) and Professional Services (PS) departments (users who have the Sales or PS department number assigned to them) to read and modify the properties of users in Customers part of the Users tree.
- **Users can handle themselves** - enables all users in the sample domain to read and modify their own properties.
- **Users can read locations/OUs** - enables all users in the sample domain to read the properties of the business objects in the Countries and Companies parts of the Business Objects tree.

My-Company Report Policies

The My-Company report access policies control who can run reports; they affect the reports that are available to different categories of users in Web Center. These policies are:

- **GM can execute privilege reports** - enables all users in the General Management department to run reports on roles, permissions and groups.

- **GM can execute user reports** - enables all users in the General Management department to run reports on users and delegations.
- **HR can execute all BO reports** - enables all users in the Human Resources department to run reports on all business objects in the **Business Objects** tree (in the sample domain, the **Companies**, **Countries** and **Projects** subtrees are populated with business objects; see the section "Business Objects" for more information). For example, log in to Web Center as **Berne** Hans*, the manager of the Human Resources department. Click **Companies** and then search for all companies to return a list of entries. Next, select **List → Run report** from the context menu of a list entry. You can see the reports on the **Companies** business objects that Hans Berner is allowed to run.

My-Company Request Workflow Policies

The My-Company request workflow access policies control the management and use of request workflows in the **Workflows → Definitions → My-Company** request workflow tree. These policies are:

- **All users can delegate to specific persons** - enables all sample domain users to delegate tasks assigned to them by request workflows to their representative, their sponsor, or their manager. (see the section "Relationships to other Users" for more information about these user properties). For more information about delegation, see the section "Delegated Administration" in the *DirX Identity Provisioning Administration Guide*.
- **All users can see all workflow instances** - enables all sample domain users to view all request workflow instances in the **Monitor** subtree of the **Workflows** view. For more information on monitoring workflow instances, see the section "Managing Request Workflows" in the *DirX Identity Provisioning Administration Guide*.
- **All users can see tasks of all other users** - enables all sample domain users to view the tasks assigned by request workflows of all other sample domain users. For example, any user logged in to Web Center can use **Show tasks list** to view the tasks assigned to a different user (provided there is a policy that allows the user to use the **Show tasks list** menu). You might want to change this policy to make it more restrictive (make sure you change the policy's name if you change its meaning). For example, you could modify the policy to allow users to see tasks of other users only if they have the same department number by adding the following rule: **$(subject.departmentnumber)="$(resource.departmentnumber)"** (see the **Rules** tab of this policy).
- **Anyone can execute self-registration workflows** - enables unregistered users to start the Customer Self Registration workflow (by clicking **Register** in Web Center).
- **Anyone can handle self-registration workflows** - enables unregistered users to step through the activities in the Customer Self Registration workflow (the previous policy allows them to start it; this policy gives read and modify access so that they can interact with it).
- **Initiator stops** and/or **suspends and resumes workflow instance** - enables a person who started a request workflow to stop or suspend and resume it.
- **Managers can change participants** - enables all My-Company managers to change a participant in an approval activity from the **Task list** context menu (according to the My-

Company menu policy for activity management described in the "My-Company Menu Policies" section).

- **Managers can delegate to specific persons** - enables all My-Company managers to delegate tasks assigned to them by request workflows to the users they manage, to their representative, or to their secretary (see the section "Relationships to other Users" for more information about these user properties).
- **Managers execute workflows** - enables all My-Company managers to start all workflows defined in the **Workflows → Definitions → My-Company** tree.
- **Managers handle all workflow definitions** - enables all My-Company managers to read and modify the workflow definitions in the **Workflows → Definitions → My-Company** tree.
- **Managers handle all workflow instances** - enables all My-Company managers to monitor the progress of request workflows in the **Monitor** subtree of the **Workflows** view.

My-Company View Assignment Policies

View assignment policies allow you to control the visibility of privilege assignments and accounts and so hide sensitive information. To activate this feature, you must set the **Enable view policies** flag in **Domain Configuration → My-Company → Policies**. If this flag is enabled, users can see by default all assignments of privileges.

My-Company view assignment access policies control the privilege assignments and accounts that are visible in Web Center to different categories of users in the sample domain. The policies provided here are activated, but the feature must be enabled at the domain level (by checking the **Enable view policies flag** in **Domain Configuration → My-Company → Policies**) for the policies to take effect.

The following sample policies exist for the sample domain:

- **Internal users can see some roles** - enables My-Company internal employees to view the internal employee and contractor roles, since these roles represent low security risks. Click the **Subjects** tab. In the **Persons filter** fields, you can see that the policy applies to users and who reside in the My-Company part of the Users tree and whose **employeeType** properties are set to "internal". Click the **Resources** tab. In the **Resources** field, you can see that the **Contractor** and **Internal Employee** corporate roles are selected. When this policy is enabled, a logged-in internal employee of My-Company can view the internal employee and contractor roles, but he cannot view any other privileges. Users in the Customers and Suppliers organizations cannot view these roles.
- **Internal users can view all accounts** - enables internal My-Company users to view all the accounts in all the My-Company target systems. Click the **Subjects** tab. In the **Persons filter** fields, you can see the same My-Company internal users settings as the other view assignment policy. Click the **Resources** tab. In the **Resource filter** fields, you can see that the policy applies to target system accounts (**dxrTargetSystemAccount**) of all the My-Company target systems (**cn=TargetSystems,cn=My-Company**). When this policy is enabled, My-Company internal employees can view all accounts in all target systems, but employees in Customers and Suppliers cannot see them. Note that you could create a more restrictive policy from this one by copying it and then changing the

**Search base** field from allowing all target systems to be viewed to allowing only low-security target systems to be viewed (for example, cn=Extranet Portal,cn=TargetSystems,cn=My-Company).

Be aware that using view assignment access policies can result in a user being unable to view any privileges or accounts. When implementing this feature, it is a good idea to create a view assignment access policy that gives an administrator the right to view all assignments and accounts. For more information on policies for viewing assignments, see the *DirX Identity Provisioning Administration Guide*.

### 3.5.2. Attribute Policies

An attribute policy is a type of object policy that is used to track and control changes to specific attributes of specific object types. Click the **Attribute Policies** node in the Policies tree view. You can see two subtrees: **Default** and **My-Company**.

The **Default** subtree contains the set of sample attribute policies provided with DirX Identity for specific object types like business objects, accounts, groups, and so on. These policies can be applied to any domain.

The **My-Company** subtree contains one attribute policy: **User - Location and Organization** - that applies to the sample domain. Click it to display its properties. In the **General** tab, you can see that the policy is activated and an approval workflow **Modify Location and Organization** is selected. The purpose of this attribute policy is to prohibit unauthorized changes to a My-Company user's organization and location values; these values are links to organizational unit and location business objects described in the "Business Objects" section. If you click the **Configuration** tab, you can see that the policy applies to the user object type (**dxrUser** is selected) and controls two user attributes (**dxroulink** and **dxrlocationlink**). As a result, any time a user's organizational unit link (**dxroulink** in the policy's **Configuration** tab; the value assigned to a user's **Organizational Unit** property in the **Users** view) or its location link (**dxrlocationlink** in the policy's **Configuration** tab; the value assigned to a user's **Location** property in the **Users** view) is changed, the **Modify Location and Organization** approval workflow automatically starts to require approval of the change if the attribute modification approval flag is set at the domain level (see Request Workflow Parameters).

You can add your own attribute policies; for example, you could add an attribute policy that protects a user's manager attribute against unauthorized change. To set up and enable an attribute policy:

- Create the request workflow that will manage the approval process for the attribute policy (the easiest way is to copy the sample workflow and tailor it to your needs) and then activate it with the **Is active** flag
- Create the attribute policy: specify the attribute(s) you want the policy to protect, the name of the workflow you created to handle the event, and activate it by setting the **Is active** flag
- Enable the attribute policy at the domain level: set the **attribute modification approval** flag in **Domain Configuration → Request Workflows** tab

Because the overhead of attribute policy execution can affect your identity system's

performance and introduce a lot of complexity, we recommend that you protect only the most sensitive attributes in your environment from unauthorized changes. For more information on setting up attribute policies, see Managing Attribute Policies.

### 3.5.3. Delete Policies

A delete policy is a type of object policy that is used to track and control the deletion of a selected set of object types. The **Delete Policies** folder contains a default delete policy and a My-Company delete policy. Click the **My-Company** delete policy. It is identical to the default delete policy, but it has been activated (the **Is active** flag is checked). This delete policy protects roles and users in the sample domain from being deleted without approval. Click the policy, and then click the **Configuration** tab. You can see the **dxrRole** and **dxrUser** object types are listed in the **OD Names** column. When a user or role is deleted, this policy ensures that an approval workflow is automatically started for approval of the deletion. For more information on delete policies, see Managing Delete Policies.

### 3.5.4. Event Policies

An event policy is a type of object policy that is used to track the creation of and changes to a selected set of object types. The **Event Policies** folder contains a default event policy and a My-Company event policy. Click the **My-Company** event policy. It is identical to the default event policy, but it has been activated (the **Is active** flag is checked). In this event policy, organizations, organizational units, users, and accounts in the sample domain have been flagged for event monitoring (context objects are also included in the policy for completeness, but are not activated for event monitoring). If you click the **Configuration** tab and look at the **OD Name** column, you can see the object description names. As a result, a change to any object with one of these selected types or a creation of a new object of one of these types triggers an event-based workflow that stores information about the creation or change event for compliance purposes (in this list, you can see that the context object type **dxrContext** is set to **false**; which deactivates it from being monitored).

Note that because a single event policy allows you to specify all the object types you want to track for events, you only need to have one event policy for a domain. For more information on event policies, see Managing Event Policies.

### 3.5.5. Password Policies

The **Password Policies** folder keeps all the password policies that pertain to My-Company. The following password policies exist:

- **Default** - the default password policy for users or accounts that have no password policy assigned. The **Active** and the **Default policy** flags are set for this policy. Note that you can set only one password policy to be the default policy.
- **Critical areas** - the password policy assigned to users or accounts that work with security-sensitive resources, for example, human resources or financial databases. In My-Company, all persons working in the General Management, Finances, Human Resources and Global IT departments are deemed to work in critical areas and thus have this special password policy assigned. For example, go to **Users → My-Company → Finances** and click on **Tinker Boris**. Now click the **Authentication** tab. In **Password**

**Policy**, you can see that **Critical areas** has been assigned.

- **Services** - the password policy assigned to "functional users" or "service accounts". The My-Company sample domain does not currently provide any examples of functional users. Note that the Services password policy does not specify an expiration time period, so the passwords of accounts that are assigned this policy will never expire. Companies often use this password strategy, but it's not very secure. An alternative is to set a password timeout of 6 months, then notify an administrator about 30 days in advance that the password will expire.

Note that all of these password policies have the maximum number of characters set to **8**. This setting is necessary for a centralized password scenario to accommodate target systems that cannot handle longer passwords (for example, a target system running UNIX).

### 3.5.6. Operations

Rules use operations. The **Operations** folder in the **Policies** view contains the set of standard operations supplied with DirX Identity. The My-Company rules use these standard operations (there are no special operations defined for the sample domain). You can read more about operations in the section "Managing Consistency Rules" in the *DirX Identity Provisioning Administration Guide*.

### 3.5.7. Rules

DirX Identity supports the following rule types:

- Consistency rules, which are used to check and clean up the Identity Store.
- Validation rules, which are used to remedy deviations between information in local target systems and the same information in the Identity Store.
- Provisioning rules, which are used to automatically provision a specific set of users with a specific set of privileges.

In the **Policies** view, open the **Rules** folder. The **Default** folder contains the default set of consistency and validation rules supplied with DirX Identity that are applicable to any domain to satisfy a variety of different use cases. The **My-Company** folder contains additional rules that apply only to the sample domain. In this folder, you can see two subtrees: **Consistency** and **Role based scenario**. The next sections describe the rules in these subtrees in more detail. To read more about rules and how to use them, see Managing Rules.

My-Company Consistency Rules

The My-Company sample domain provides the following additional consistency rules:

- **Cleanup inofficial memberships** - this rule is intended to be used after a validation workflow run to enforce consistency between the external account-group memberships in a target system - in this case, the Windows Domain Europe target system - and the validated information about these memberships imported into the Identity Store. The rule checks the account-group memberships in the Windows Domain Europe target system and automatically removes any account-group

memberships that are not allowed.

- **Transport mail attribute from account to users** - this rule is an example of a consistency rule for managing attribute flow between external target systems and the Identity Store. The Windows Domain Europe target system runs a Microsoft Exchange server that manages Windows email addresses for accounts on that system. When you run a validation or an initial load workflow, this email information is brought into the Identity Store as an attribute of the imported account. This rule copies the email addresses defined in the external target system from the imported accounts, provisions them to the user entries in the Identity Store, and from there possibly to other relevant accounts in other target systems through the "master attribute" mechanism. For more information about this general process, see Managing Attribute Flow.

My-Company Role-Based Scenario Rules

If we look at **Role based scenario**, we can see that it consists of a set of B2B and a set of corporate provisioning rules. Some examples are:

- The **Standard Customer** rule, which assigns the **Silver Customer** role to all users under the **dc=Customers,cn=Users,cn=My-Company** subtree with **employeeType** set to **Customer**. The **Standard Supplier** rule is very similar to this rule.
- The **Mercato Aurum = Platin** rule, which assigns the **Platinum Customer** role to all Mercato Aurum employees (**o=Mercato Aurum,dc=Customers,cn=Users,cn=My-Company**). The **TakeAway = Gold** rule is similar.
- The **Internal Employees** rule, which assigns the **Internal Employee** role to all users under the **o=My-Company,cn=Users,cn=My-Company** subtree with **employeeType** set to **Internal**.
- The **Signature Level 1** rule, which assigns the **Signature Level 1** permission to all internal users (**o=My-Company,cn=Users,cn=My-Company** subtree with **employeeType** set to **Internal**.).
- The **Access to Berlin - Data Center** rule, which assigns the **Berlin - Data Center** group in the Physical Access target system to all internal employees who are located in My-Company's Berlin office (**l=My-Company Berlin**).The other "Access" rules are similar to this one.

### 3.5.8. SoD Policies

Open the **SoD Policies** folder. It shows segregation of duty policies.(Some people use the term "separation of duty" instead.) The **Default** folder shows policies that can be relevant for any customer domain, while the **My-Company** folder shows the policies that are only relevant for the sample domain.

Open the **My-Company** folder. Here you can see the SoD policies that can be used for the sample domain. For example, the SoD policy **Contractor** <> **Manager** specifies that My-Company contractors cannot be managers; that is, a user with the **Contractor** privilege cannot also have the **Manager** privilege. Should someone assign a conflicting privilege to a user, DirX Identity runs an SoD mitigation workflow that automatically notifies specific users (called "participants") about the SoD violation; these users may override the violation if necessary; DirX Identity will not assign the privilege unless the violation is specifically

overridden. In the case of the **Contractor <> Manager** conflict, two company heads - Olivier Hungs and Gabriela Morton - are notified of the violation. One of these users must approve or reject the privilege assignment. Violations to a policy are stored below the policy in this subtree and in the user entry (see the **SoD Exceptions** tab). For a demonstration of the SoD process, see the follow-on tutorial Applying SoD Policies.

To use the sample SoD policies, you will need to activate them individually (check each policy's **Is active** flag) and then check the **Segregation of Duty (SoD) checks** flag at the domain.You can read more about SoD policies and how to create them in the section "Managing SoD Policies" in the *DirX Identity Provisioning Administration Guide*.

## 3.6. Request Workflows

The My-Company sample domain provides a set of request workflows that are necessary for secure provisioning in My-Company.In the **Workflows** view, open the **Definitions** subtree. You'll see a **Default** folder, a **My-Company** folder and a **System** folder.

The **System** folder is the repository for standard, system-wide request workflows provided with DirX Identity. If you open it, you can see one request workflow named **SendMail** which is the standard workflow for sending email messages.

Open the **Default** folder. You will see a set of folders that contain the request workflows supplied with DirX Identity that can be applied to any domain. Customers who are setting up their own domains should evaluate whether these default request workflows apply to their environment, and then copy the workflows they want to use into their separate domain-specific area, change them to their requirements and then activate them for use by checking their **Active** flag.The section "Using Request Workflows" in the *DirX Identity Application Development Guide* provides more details about these default request workflows and how to use them.

Open the **My-Company** folder. You can see that the folder organization here is similar to the **Default** folder. The My-Company sample domain uses many, but not all the default request workflows. It also provides the following request workflows that are specific to the My-Company domain:

- The **Approve Customer Self Services** workflow in the **Approval** folder. This request workflow requests an approval of a Sales Manager when a customer uses Web Center self-service to subscribe to privileges. It consists of an **Approval by Sales Manager** activity that contains the manager of the Sales Department, Ruben Briner (this is an example of a static approval definition). Users can only subscribe to the privileges available in customer self-service if this person approves it. If he rejects the approval, the **Notification if Rejected** activity informs the user who is subject to the approval. The getting started tutorial "User Self-Registration" in this guide provides an example of this process. The subsection "User Self-Registration Workflows" within "Using Request Workflows" in the *DirX Identity Application Development Guide* describes this type of workflow in more detail.
- The **Physical Access** workflow in the **Service Management** folder. This request workflow is an example of a manual provisioning workflow. It requests manual provisioning of the **Physical Access** target system, which manages access by My-Company employees to

secured rooms. The follow-on tutorial "Using Manual Provisioning" in this guide provides an example of how to set up and use this manual provisioning workflow.The subsection "Manual Provisioning Workflow" within "Using Request Workflows" in the *DirX Identity Application Development Guide* describes this type of workflow in more detail. The DirX Identity use case document "Service Management" provides more information on the service management concept in DirX Identity and how to use it.

- The **Modify Location and Organization** workflow in the **Users** folder. This request workflow is an example of an attribute modification workflow for controlled modification of a user's attributes. This workflow is automatically started when someone changes a user's location or organization attributes, which have been declared in the **User - Location and Organization** attribute policy to be attributes that require change control (see Attribute Policies. The follow-on tutorial Applying Attribute Modification Approval provides an example of how to set up and use attribute modification approval. The subsection User Modification Workflows describes this type of workflow in more detail.

## 3.7. Target Systems

My-Company uses a corporate network with two subnets in Europe and the U.S.A. The following figure shows an overview of the IT infrastructure.

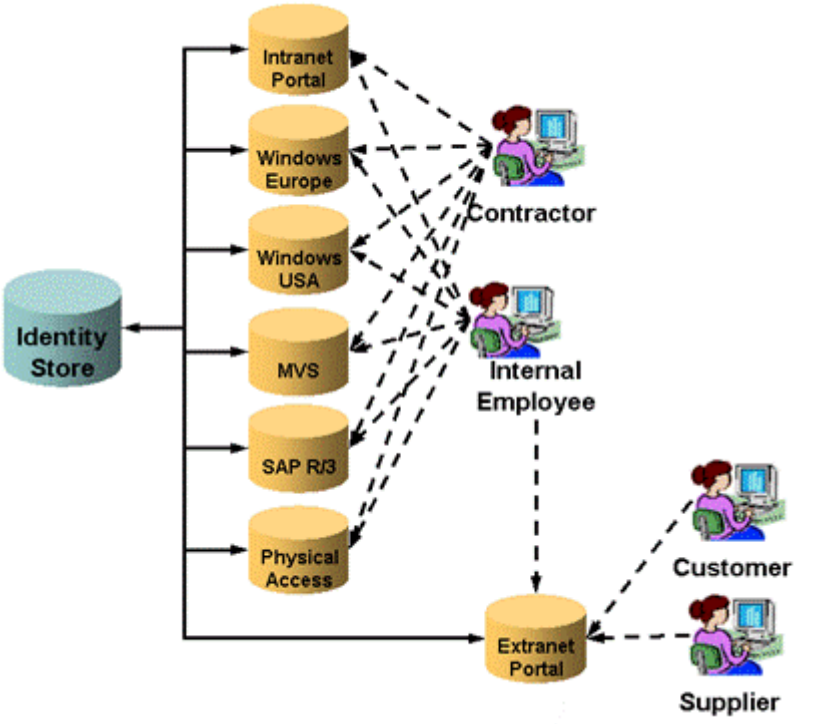



*Figure 20. Sample Domain IT Structure*

Customers and suppliers both have access to the Extranet Portal application. All internal employees and the contractors work with a set of applications mainly based on the Windows operating system. Information is provided by an Intranet Portal application. Some employees and contractors need to work with MVS or SAP R/3 applications. The central Identity Store provisions all of these target systems.

Click the **Target Systems** view to view the target systems.They are:

- **DirXmetaRole** (type LDAP with no assignment states) - the standard target system with all the groups that are necessary for managing DirX Identity itself. There are no accounts in this target system; the users serve this purpose. Typical groups here are UserAdmins, who are responsible for user administration, and TSAdmins, who are responsible for target system administration. A set of template groups serves for special purposes. Here DirX Identity can automatically collect, for example, all department managers or all project managers. These groups can be used for specific provisioning or access policy tasks.
- **Extranet Portal** (type LDAP with separate account and group trees) - a target system used for electronic data exchange of information between the Procurement department and the supplier companies and between the Sales departments and the customer companies. This target system contains groups to classify customers and suppliers and an Administrator group.
- **Intranet Portal** (type LDAP with a common account and group tree) - a target system that provides information to specific groups of internal employees and contractors. This target system contains group portals, such as a Sales portal for the employees of the Sales departments and a Training portal for the employees of the Training department. Note that the accounts and groups in this target system are kept in one common subtree.
- **MVS** (type RACF with separate account and group trees) - this target system is used for software development and testing in the U.S.A. To keep it simple, all software development and all software test tools are collected in one group. In reality, having several groups makes more sense, especially when expensive licenses for the tools must be purchased. (Auditing and management of licenses is easy with DirX Identity. You always know exactly how many people use a specific license.)
- **SAP R3** (type SAP R3 UM with separate account and group trees) - this target system is comprised of several applications like HR (human resources), CRM (customer relationship management) and FI (finance application).For simplicity, only access to these applications is modeled.In reality, there might be specific roles for different people in each of the applications.
- **Signatures** (type Virtual) - this target system handles lists of three different signature levels. Level 1 is for all internal employees, level 2 for all managers and level 3 only for a few selected persons in general management. Virtual lists mean that there is no real target system representation that must be synchronized; as a result, this type of target system has no accounts and no assignment states. Perhaps the lists are printed and used by the Finances department to verify the signature level of an order. A better method is for Procurement to look up the signature level directly in the DirX Identity Store. The most sophisticated variant is that there is a special application that verifies the signature level automatically against these lists during an order workflow. If this application is, for example, the FI application, it can be modeled either as an associated target system or the lists can be modeled as groups in the SAP R/3 target system (it is not done this way in this example).
- **Windows Domain Europe** (type Windows 2000 with a common account and group tree) - this target system is the Windows 2008 domain in Europe. All European employees work in this environment.

- **Windows Domain USA** (type Windows 2000 with a common account and group tree) - this target system is the Windows 2008 domain in the U.S.A. All U.S. employees work in this part of the My-Company network.
- **Physical Access** (type Request Workflow) - this target system is an example of an "offline" target system that is provisioned manually by an administrator through request workflows that are started automatically by real-time workflows, instead of being directly provisioned by DirX Identity Provisioning. In the sample domain, this target system manages physical access by My-Company employees to computer rooms in My-Company's Berlin, Frankfurt and Munich locations that are secured with keyless entry systems. The follow-on tutorial "Using Manual Provisioning" in this guide demonstrates how to work with this type of target system. The DirX Identity use case document "Service Management" provides more information on the service management concept in DirX Identity and how to use it.

Open each target system subtree to check the details, especially the groups in each target system. Click the **Member** tab in the groups to see which persons have these access rights.

For details about the types of target systems and how to configure them, see Customizing Target Systems.

# 3.8. Auditing

Click the **Auditing** view. You can see two separate folders: **Status Reports** and **Audit Trail**.

## 3.8.1. Status Reports

The **Status Reports** tree is designed to provide a single (virtual) point of view for distributed status reports. Open the folder. You can see two folders: **Default** and **Customer Specific**.

Open the **Default** folder. This folder contains the central report definitions that you can also find in **Domain Configuration → Reports**. Now open **Target System Specific**. This folder is organized into target system-specific subfolders; each one contains the relevant **Report** folder for the target system, which contains the reports that are valid for this type of target system. For example, the folder **Default → Target System Specific → LDAP → Reports** can be found under **Domain Configuration → Target Systems → LDAP → Reports**. It contains the valid reports for the LDAP target system type.

The reports in the **Default** folder and its subfolders are defaults that you can use, but do not modify them here because your changes will be overwritten with DirX Identity upgrades.

Now open the **Customer Specific** folder. This folder refers to **Domain Configuration → Customer Extensions → Reports** and is the location at which you can set up your own reports. To create reports for your domain, examine the status reports in the **Default** folder, copy the ones that apply to your environment into a separate domain-specific folder that you create in the **Customer Specific** subtree, and then change them to your requirements. For information on how to customize the default status reports, see the subsection "Customizing Status Reports" in "Customizing Auditing" in the *DirX Identity Customization Guide*. There are no sample domain-specific reports, but there are a lot of default reports that you can use.

Under the **Customer Specific** subtree, you can see a **Target System Instances** folder. If you open it, you can see a list of target system-specific folders that correspond to the sample target system types provided with the sample domain (see Target Systems for more information). You can use these subfolders to create reports for a specific target system in your domain: copy a default report for a target system type to the corresponding target system instance folder in your domain-specific part of the tree, and then modify it to your requirements. Your report is then also visible under the corresponding folders in **Target Systems** → *target system instance* → **Configuration → Reports**. Try creating some reports in the **Auditing → Status Reports** view and then check to see if they're visible in the other views.

Note the **Target Systems** subfolder that also appears in the **Customer Specific** subtree. This subfolder is not operational and will be removed in a future DirX Identity release.

### 3.8.2. Audit Trail

Now open **Audit Trail** and then open **Audit Policies**. Here you can see three different subtrees:

- The **_Queries** folder, which provides a set of default query folders for checking on active and inactive audit policies.For more information about queries, see Creating a Query Folder.
- The **Default** folder, which provides a set of default audit policies for objects that can be applied to any domain.These policies are samples and have not been activated (their **Active** checkbox is clear).
- The **My-Company** folder, which contains the corporate audit policies for objects in the sample domain. The policies here are copies of the default audit policies that have been activated for use (their **Active** checkbox is checked) but are otherwise unchanged from the defaults. To enable them, check the flag at the domain configuration level (**Enable Auditing for → Service Layer** in the **Compliance** tab).

The procedure for using the default audit policies in your own domain is the same as for using default status reports and the other default DirX Identity objects: determine which default audit policies apply, copy them from the **Default** folder into your own domain-specific folder with the name of your domain, and then customize the copies to your requirements. Segregating your customizations from the **Default** folder allows you to preserve them across DirX Identity upgrades and provides a simple structure for synchronizing your customizations into your production system with the DirX Identity transport mechanism after you've tested them sufficiently. For information on how to set up audit trail and customize audit policies, see Managing the Audit Trail. For information on the DirX Identity transport mechanism, see Transporting Data.

## 3.9. Domain Configuration

The My-Company sample domain sets a number of global parameters that control DirX Identity Provisioning operation for the domain. To view them, select DirX Identity Manager's **Provisioning** view, click **Domain Configuration** and then click the top-level node **My-Company**.

### 3.9.1. General Domain Parameters

The **General** tab supplies information about the domain, such as the name by which it is known throughout the Provisioning system, a description of its function, its type (for example, "test domain" or "production domain"), and information about the server for the Connectivity configuration domain that corresponds to the domain. The **Include domain into topic** flag is a DirX Identity version compatibility flag that is always set .for any domain.

### 3.9.2. Policy Parameters

The **Policies** tab provides the following parameters:

- **Disable access policies** - when checked, this flag disables access policy handling (by turning off the DirX Identity Security Manager). You should only set this flag during testing because it allows all users to have access to all DirX Identity features. This flag is not set in the My-Company sample domain.
- **Web Center/Enable view policies** - when checked, this flag enables previously defined Web Center view assignment policies (if the individual policies themselves have been activated). View assignment policies control the privilege assignments and accounts that are visible in Web Center and DirX Identity Manager to different categories of users in the My-Company sample domain. If you check this flag, the view assignment policies provided with the My-Company sample domain will take effect. To examine these policies, go to the **Policies** view, click **Policies → Access Policies → My-Company → View Policies**. You can read more about the My-Company view policies and what they control in the "Access Policies" section of this chapter. Web Center* /**Enable menu policies** - when checked, this flag enables previously defined Web Center menu policies (if the individual policies themselves have been activated) that control the Web Center menus seen by different logged-in users. This flag is not set in the My-Company sample domain, allowing all users to see all Web Center menus and submenus after they log in. If you check this flag, the menu policies provided with the My-Company sample domain take effect, and users see only the Web Center menus defined by these policies. To examine these policies, in the **Policies** view, click **Policies → Access Policies → My-Company → Menu Policies**. We recommend that you set this flag, and then log in as different sample domain users to explore the different menu selections available to each user. You can read more about the My-Company menu policies and what they control in the "Access Policies" section of this chapter.

### 3.9.3. Timing Parameters

The **Timing** tab provides domain-wide controls that relate to time limits. The My-Company domain uses the standard settings of a new domain for these parameters. For more information about the fields displayed in this tab, see the online help and read User and Account Life-Cycle in *DirX Identity Provisioning Administration Guide*.

### 3.9.4. Permission Parameters

The **Permission Parameters** tab shows the user attributes to be used in security policies or permission match rules in the My-Company sample domain. The "Getting Started" exercise "Using Permission and Role Parameters" " in the section "Adding a New User"

demonstrates how the sample domain uses permission parameters for automatic privilege assignment. For detailed information about permission parameters, see the path "Managing the Privilege Structure" → "Managing Privilege Resolution" → "Handing Permission Parameters" in the *DirX Identity Provisioning Administration Guide*.

### 3.9.5. Privilege Resolution Parameters

The **Privilege Resolution** tab provides parameters that relate to privilege assignment and resolution. The My-Company domain does not use any of these parameters. For information about the fields displayed in this tab, see the online help and the following documentation links:

- **Offline resolution** - "Tuning the Provisioning System" in the *DirX Identity Provisioning Administration Guide*.
- **Smooth account creation** - "Smooth Account Creation" in "Managing Target Systems" → "Managing Target System Accounts" → "Managing Personal Accounts" in the *DirX Identity Provisioning Administration Guide*.

### 3.9.6. Request Workflow Parameters

The **Request Workflow** tab provides parameters for controlling DirX Identity request workflows. For information about the fields displayed in this tab, see the online help. The My-Company sample domain uses these parameters as follows:

- **Attribute modification approval** - the My-Company sample domain does not initially set this parameter. We recommend that you set it to explore how DirX Identity handles changes to security-sensitive attributes. See Applying Attribute Modification Approval for a demonstration of how to use this feature.
- **Approval on deassign** - the My-Company sample domain does not set this parameter; when a user loses a privilege, the relevant request workflow automatically starts.
- **Approval content read-only** - the My-Company sample domain does not set this parameter; approvers are allowed to change the data they are approving.
- **User creation** - the My-Company sample domain does not set these flags because they are not applicable to this scenario (the sample does not provide a Web Services setup in which a client triggers user creation from Web Services). Note that these flags are not relevant to Web Center; if it runs in the Business Suite, it creates users directly, and if it runs in the Professional Suite, it creates them using request workflows. Note, however, that you can customize Web Center to use the direct user creation mechanism as well as use request workflows when it runs in the Professional Suite.
- **Default language**- the My-Company sample domain supports the selection of English and German as default languages for request workflow mail messages; the other languages shown in the selection list are not set up. You can set up additional languages to be used in request workflows. For information on this task, see the path "Understanding Default Application Workflow Technology" → "Understanding Request Workflows" → "Request Workflow Architecture → "Nationalizing Request Workflows" in the *DirX Identity Application Development Guide*.

### 3.9.7. Domain Compliance Parameters

The **Compliance** tab provides domain-wide controls that relate to compliance measures. For information about the fields displayed in this tab, see the online help. The My-Company sample domain uses these parameters as follows:

- **Segregation of Duty (SoD) checks** -The My-Company sample domain does not set this flag because it decreases performance of privilege resolution. For a demonstration of how to set up and use SoD checking, see Applying SoD Policies.
- **Enable Auditing for** - The My-Company sample domain does not set these flags.We recommend that you enable them to explore the DirX Identity's auditing feature either with audit file creation or with an online connection to the DirX Audit database. For more information on this feature, see Managing the Audit Trail and the DirX Audit documentation. For more information on the sample audit policy setup, see Auditing.
- **Enable Client Signature for** - The My-Company sample domain does not set these flags.See Managing Auditing for more information about setting up your DirX Identity environment for client signature and verifying signatures.
- **Check Certificate Owner** - The My-Company sample domain does not set these flags.For more information on this feature, see Customizing the Certificate Owner Check in *DirX Identity Customization Guide*.

## 3.10. Project Organization

My-Company's project organization demonstrates several different DirX Identity concepts and is distributed across the following objects:

- The role parameter **Project** is defined in the view **Domain Configuration → My-Company → Customer Extensions → RoleParams → My-Company**. It references the My-Company business objects in the **Business Objects → Projects** tree.You can read more about these objects and their use as role parameters in the "Business Objects" → "Projects" section of this chapter.
- The **Project Specific** roles subtree in the view **Privileges → Roles → Corporate Roles** contains the roles **Project Member** and **Project Manager**. Both roles use the role parameter **Project**.The match rule in each role's **Role Parameters** tab specifies that the **Project** parameter of the user-to-role assignment must be compared with the corresponding parameter of all group objects (dxrproject) that are part of all attached permissions.
- The **Project Specific** permission subtree in the view **Privileges → Permissions → Corporate Permissions** contains the permissions **Project Member** and **Project Manager** that are used by the respective roles.
- The **Project Manager** permission has a match rule (click the **Match Rules** tab), which means it uses a combination of role and permission parameters. Click the **Assigned Groups** tab.You can see that there are two **Project Management** groups assigned to this permission. Click the ... icon to the right of each group and then click its **Permission Parameters** tab. You can see that the **Country** field is different. It allows for the selection of one of these groups for U.S.A. and European users and illustrates group selection via permission parameter match rules.Note that the **Projects** permission

parameter of these groups is set to "*", which means that they are independent of the **Projects** permission parameter that is used for selection in the other groups. For example, look at the **OptimizeIT Portal** group. You will see that the **Country** permission parameter is set to "*" and the **Projects** permission parameter is used for group selection from the role parameter.

The **Project Manager** permission is also assigned to the groups **HighPerformance Portal**, **MoreCustomers Portal** and **OptimizeIT Portal** because a project manager is also a project member. Three manager groups are also assigned (**Manage HighPerformance Portal**, and so on). It is also assigned to the **allProjectManagers** group, which is part of a "grant" access policy implemented for My-Company. A project manager is automatically assigned to this group so that he is allowed to grant project-specific privileges to users. You can read more about My-Company access policies in the "Access Policies" section of this chapter.

- The **Project Member** permission does not use a match rule. It uses the **Projects** parameter to select from the groups **HighPerformance Portal, MoreCustomers Portal** and **OptimizeIT Portal**, which are part of the Intranet Portal target system (see the view **Target Systems → Intranet Portal → Accounts and Groups → Project Portals**. In this example, we assume that the complete management of projects is carried out through this Intranet Portal application. Each of these groups has the **Country** permission parameter is set to "*" and the **Projects** permission parameter set to a project-specific role parameter value.

# 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.