

Oracle® Data Integrator

Studio Online Help

12c (12.2.1.2.0)

E77155-01

October 2016

Oracle Data Integrator Studio Online Help, 12c (12.2.1.2.0)

E77155-01

Copyright © 2010, 2016, Oracle and/or its affiliates. All rights reserved.

Primary Author: Laura Hoffman Miquel

Contributing Authors: Ale Paez, Joshua Stanley, Pratima Chennupati

Contributor: Alex Kotopoulos

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	xvii
Documentation Accessibility	xvii
Conventions	xvii
1 Oracle Data Integrator Studio Overview	
2 Using the Online Help	
2.1 About the Online Help	2-1
2.2 About the Help Center	2-1
2.3 Opening the Online Help.....	2-2
2.4 Searching the Documentation.....	2-3
3 Main Windows	
3.1 About Oracle Data Integrator Studio.....	3-1
3.2 About the Online Help.....	3-2
3.3 Check for Updates Wizard	3-3
3.3.1 Step 1: Select update source	3-3
3.3.2 Step 2: Select updates to install.....	3-3
3.3.3 Step 3: View download progress.....	3-4
3.3.4 Step 4: View summary	3-4
3.4 Find ODI Object Navigator	3-4
3.4.1 Standard	3-5
3.4.2 Advanced	3-5
3.5 Log Window	3-5
3.6 Manage Libraries Dialog.....	3-6
3.7 New Gallery.....	3-7
3.8 Preferences Dialog	3-8
3.8.1 Preferences Dialog - Environment Page.....	3-8
3.8.2 Preferences Dialog - Environment - Log Page.....	3-9
3.8.3 Preferences Dialog - Code Editor Page.....	3-10
3.8.4 Preferences Dialog - Code Editor - Caret Behavior Page.....	3-11
3.8.5 Preferences Dialog - Code Editor - Display Page.....	3-12
3.8.6 Preferences Dialog - Code Editor - Fonts Page.....	3-13
3.8.7 Preferences Dialog - Code Editor - Line Gutter Page.....	3-14
3.8.8 Preferences Dialog - Code Editor - Syntax Colors Page.....	3-14

3.8.9	Preferences Dialog - Credentials Page.....	3-15
3.8.10	Preferences Dialog - Credentials - HTTP Keystore Tab Page	3-16
3.8.11	Preferences Dialog - Credentials - Username Token Tab Page.....	3-16
3.8.12	Preferences Dialog - Certificates - X509 Certificates Tab Page.....	3-17
3.8.13	Preferences Dialog - Credentials - STS Configuration Tab Page	3-17
3.8.14	Preferences Dialog - Diagrams - Annotation Page	3-17
3.8.15	Preferences Dialog - Http Analyzer	3-19
3.8.16	Preferences Dialog - ODI - System Page	3-19
3.8.17	Preferences Dialog - ODI - System - Agent Page	3-20
3.8.18	Preferences Dialog - ODI - System - Concurrent Development Page.....	3-20
3.8.19	Preferences Dialog - ODI - System - Groovy Page.....	3-21
3.8.20	Preferences Dialog - ODI - System - Reports Page	3-21
3.8.21	Preferences Dialog - ODI - System - Scenarios Page	3-21
3.8.22	Preferences Dialog - ODI - User Interface Page	3-21
3.8.23	Preferences Dialog - ODI - User Interface - Designer Page	3-22
3.8.24	Preferences Dialog - ODI - User Interface - Operator Page	3-23
3.8.25	Preferences Dialog - ODI - User Interface - Topology Page	3-23
3.8.26	Preferences Dialog - ODI - User Interface - Versioning	3-24
3.8.27	Preferences Dialog - ODI - User Interface - Versioning - Subversion Page	3-24
3.8.28	Preferences Dialog - Web Browser and Proxy - Web Browsers.....	3-24
3.8.29	Preferences Dialog - Web Browser and Proxy - Proxy Page	3-25
3.8.30	Preferences Dialog - Web Browser and Proxy - Internet Files Page.....	3-26
3.9	Property Inspector	3-26
3.10	Save Files	3-26
3.11	Save to File	3-27
3.12	Thumbnail	3-27
3.13	Unable to Connect Dialog.....	3-27
3.14	Unable to Save Dialog	3-27
3.15	Update Center Dialog.....	3-28
3.16	Windows Dialog	3-28

4 Navigator-Independent Editors, Wizards, and Dialogs

4.1	Add ODI Objects to Subversion Dialog.....	4-3
4.2	Add Open Tool Dialog.....	4-4
4.3	Change Current ODI User Password Dialog.....	4-4
4.4	Change Wallet Password Dialog	4-5
4.5	Clean Stale Sessions Dialog	4-5
4.6	Clean Stale Sessions Results Dialog	4-6
4.7	Compare with Export File Dialog.....	4-6
4.8	Configure Subversion Repository with ODI Dialog.....	4-6
4.9	Connect to Debug Session... Dialog.....	4-7
4.10	Contents Of Deployment Archive Dialog	4-7
4.11	Create Branch From Tag Dialog	4-8
4.12	Create Default SVN Project Structure Dialog	4-8
4.13	Create Deployment Archive with All Repository Objects Dialog.....	4-8
4.14	Create Deployment Archive from Selected Objects Dialog.....	4-9
4.15	Create Deployment Archive from VCS Label Dialog.....	4-10

4.16	Create Full Tag Dialog	4-11
4.17	Create New Model and Topology Objects Wizard.....	4-11
4.17.1	Step 1 - Create a new model and associate it with a new or existing logical schema and a new or existing context	4-11
4.17.2	Step 2 - Associate the new model with a new or existing data server	4-12
4.17.3	Step 3 - Associate the new model with a new or existing physical schema.....	4-12
4.18	Big Data Configurations Wizard	4-13
4.18.1	Step 1 - Specify the general settings	4-13
4.18.2	Step 2 - Configure the data servers for selected Big Data technologies.....	4-14
4.18.3	Step 3 - Validate all the settings.....	4-14
4.19	Create Partial Tag Dialog.....	4-14
4.20	Select Tag Dialog.....	4-15
4.21	Create Version Dialog	4-15
4.22	Datasources Deployment Dialog	4-15
4.23	Debugger.....	4-16
4.23.1	Debug Variables.....	4-17
4.23.2	Debug Breakpoint.....	4-17
4.23.2.1	Breakpoint Properties	4-19
4.23.3	Debug Session	4-19
4.23.4	Debug Session Threads	4-20
4.23.5	Debug Data	4-20
4.24	Deployment Archive Details Dialog	4-20
4.25	Display Options Dialog.....	4-21
4.26	Drivers Dialog	4-21
4.27	Enter Export Key Dialog for Multiple Files Import	4-21
4.28	Enter Export Key Dialog for Single File Import	4-22
4.29	Enter Export Key Dialog to Compare with Export File.....	4-22
4.30	Enter Wallet Password Dialog	4-22
4.31	Export Key fields in Export Dialog	4-23
4.32	Export of <Object> Dialog.....	4-23
4.33	Export of <Version> Dialog.....	4-24
4.34	Export Multiple Objects Dialog	4-24
4.35	Export Selection Dialog.....	4-26
4.36	Groovy Editor	4-26
4.37	Import Report	4-27
4.38	Import Scenario Dialog	4-29
4.39	Import Selection Dialog	4-29
4.40	Initialize Deployment Repository Dialog.....	4-30
4.41	Inventory Dialog	4-30
4.42	Master Repository Creation Wizard	4-31
4.42.1	Step 1 - Repository Connection.....	4-31
4.42.2	Step 2 - Authentication.....	4-32
4.42.3	Step 3 - Password Storage.....	4-32
4.43	Master Repository Import Wizard	4-33
4.43.1	Step 1 - Repository Connection.....	4-33
4.43.2	Step 2 - Password Storage.....	4-34
4.44	Merge Dialog	4-35

4.45	Merge Results Window.....	4-35
4.45.1	Merge Object Selection.....	4-35
4.45.2	Merge Conflict Resolution.....	4-36
4.46	Merge Summary Dialog.....	4-37
4.47	Merge Summary Selection Dialog.....	4-38
4.48	New Password Dialog.....	4-38
4.49	New Wallet Password Dialog.....	4-38
4.50	OPSS Enterprise Roles Dialog.....	4-39
4.51	Oracle Data Integrator Login Dialog.....	4-39
4.52	Patch Deployment Repository Dialog.....	4-40
4.53	Preferences Dialog.....	4-40
4.54	Recover Password Storage Wizard.....	4-41
4.55	Release Tag Selection Dialog.....	4-41
4.56	Release Tag Wizard.....	4-42
4.56.1	Step 1.....	4-42
4.56.2	Step 2.....	4-42
4.57	Repository Connection Information Editor.....	4-42
4.58	Restart Debug Session Dialog.....	4-43
4.59	Restart Session Dialog.....	4-44
4.60	Restore Deleted ODI Object.....	4-44
4.61	Restore Object from Subversion Repository Dialog.....	4-45
4.62	Rollback the Patch Dialog.....	4-46
4.63	Search Messages Log Dialog.....	4-46
4.64	Select Version Control Management System Dialog.....	4-47
4.65	Simulation Report.....	4-47
4.66	Smart Export Dialog.....	4-48
4.67	Smart Export Report.....	4-50
4.68	Smart Import Report.....	4-51
4.69	Smart Import Wizard.....	4-54
4.69.1	Step 1 - File Selection.....	4-54
4.69.2	Step 2 - Import Actions.....	4-54
4.69.3	Step 3 - Summary.....	4-55
4.70	Start Debug Session... Dialog.....	4-56
4.71	Stop Session Dialog.....	4-57
4.72	Structure Pane.....	4-57
4.73	SVN Connection Dialog.....	4-57
4.74	Switch Authentication Mode Wizard.....	4-59
4.74.1	Step 1 - Login.....	4-59
4.74.2	Step 2 - Credentials.....	4-60
4.75	Password Storage Wizard.....	4-60
4.75.1	Step 1 - Login.....	4-60
4.75.2	Step 2 - Password Storage.....	4-61
4.76	Switch Repository Compatibility Mode Wizard.....	4-61
4.77	Target Object Selection Dialog.....	4-61
4.78	URL Examples Dialog.....	4-62
4.79	Version Browser Dialog.....	4-62
4.80	Version Compare Results Window.....	4-62

4.81	Version History Dialog	4-63
4.82	Version Selection Dialog	4-64
4.83	Version Tree Editor	4-65
4.84	Version Tree Filter Dialog.....	4-66

5 Operator Navigator Editors

5.1	Define Filter Dialog.....	5-1
5.2	Export the Log Dialog	5-2
5.3	Import of the Log Dialog	5-3
5.4	Load Plan and Scenario Folder	5-3
5.4.1	Definition	5-4
5.5	Load Plan Run Editor	5-4
5.5.1	Execution.....	5-4
5.5.2	Load Plan	5-5
5.5.3	Steps.....	5-6
5.5.4	Variables.....	5-8
5.6	Operator Navigator	5-8
5.7	Purge Log Dialog	5-9
5.8	Remove Temporary Objects Dialog	5-10
5.9	Regenerate Scenario Dialog.....	5-10
5.10	Restart Load Plan Dialog	5-11
5.11	Scenario Variable Editor	5-11
5.11.1	Definition	5-11
5.12	Session Editor	5-12
5.12.1	Definition	5-12
5.12.2	Steps.....	5-14
5.12.3	Blueprint.....	5-14
5.12.4	Details.....	5-15
5.13	Session Folder Editor.....	5-15
5.13.1	Definition	5-16
5.14	Session Step Editor.....	5-16
5.14.1	Definition	5-16
5.14.2	Details.....	5-18
5.15	Session Task Editor	5-18
5.15.1	Definition	5-19
5.15.2	Code.....	5-20
5.15.3	Connection.....	5-21
5.15.4	Details.....	5-21
5.16	Session Variable Editor	5-22
5.16.1	Definition	5-22
5.16.2	History.....	5-22
5.17	Start Load Plan Dialog	5-23
5.18	Stop Load Plan Dialog.....	5-23

6 Designer Navigator Editors

6.1	Add Step Wizard.....	6-3
-----	----------------------	-----

6.1.1	Step 1 - Select a type and location of the step	6-4
6.1.2	Step 2 - Select Step Name (for Serial, Parallel, Else, and Exception steps).....	6-4
6.1.3	Step 2 - Select Scenario (for Run Scenario steps).....	6-4
6.1.4	Step 2 - Select Variable to Evaluate (for Case steps).....	6-4
6.1.5	Step 2 - When Clause Evaluation (for When steps).....	6-5
6.1.6	Step 3 - Select Scenario Variables (for Run Scenario steps)	6-5
6.2	Attribute Editor	6-5
6.2.1	Definition	6-6
6.2.2	Description.....	6-8
6.2.3	Control.....	6-8
6.3	Attribute Matching Dialog	6-9
6.4	Condition Editor	6-10
6.4.1	Definition	6-10
6.4.2	Control.....	6-11
6.5	Connection Path Selection Dialog	6-11
6.6	Create a Lookup or a Join Dialog	6-12
6.7	Create <Version> Dialog	6-12
6.8	Cross Reference Editor	6-12
6.8.1	Definition	6-13
6.9	Datastore Editor	6-13
6.9.1	Definition	6-14
6.9.2	Attributes	6-14
6.9.3	Files	6-17
6.9.4	Journalizing	6-17
6.9.5	Partitions	6-17
6.9.6	Services.....	6-18
6.9.7	View Data Editor	6-18
6.9.8	Data... Editor.....	6-18
6.10	Designer Navigator.....	6-19
6.11	Diagram Editor.....	6-19
6.11.1	Definition	6-20
6.11.2	Diagram.....	6-20
6.12	Edit Choice Dialog	6-20
6.13	Export All Scenarios Dialog	6-21
6.14	Export Work Repository Dialog	6-21
6.15	Expression Editor	6-22
6.16	Fetch Data for Reverse Engineering Dialog.....	6-23
6.17	Filter Editor	6-23
6.17.1	Definition	6-24
6.18	Folder Editor.....	6-24
6.18.1	Definition	6-25
6.18.2	Version.....	6-25
6.19	Generate DDL Editor.....	6-26
6.20	Generate Mappings IN Editor.....	6-27
6.21	Generate Mappings OUT Editor.....	6-27
6.22	New <Object> Dialog.....	6-28
6.23	Import <Object> Dialog.....	6-28

6.24	Import Work Repository Dialog	6-29
6.25	Key Editor	6-29
6.25.1	Description.....	6-30
6.25.2	Attributes	6-30
6.25.3	Control.....	6-30
6.26	Knowledge Module Editor	6-31
6.26.1	Definition	6-31
6.26.2	Tasks	6-32
6.26.3	Options	6-36
6.26.4	Version.....	6-37
6.27	Load Plan Editor	6-38
6.27.1	Definition	6-38
6.27.2	Steps.....	6-40
6.27.2.1	Common Properties	6-41
6.27.2.2	Serial Step Properties	6-43
6.27.2.3	Parallel Step Properties.....	6-43
6.27.2.4	Run Scenario Step Properties.....	6-43
6.27.2.5	Case Step Properties.....	6-45
6.27.2.6	When Step Properties.....	6-45
6.27.2.7	Else Step Properties	6-45
6.27.3	Exceptions	6-46
6.27.4	Variables.....	6-46
6.28	Locked Objects Dialog.....	6-47
6.29	Lookup Scenario Dialog.....	6-47
6.30	Lookup Variable Dialog.....	6-48
6.31	Mapping Editor	6-48
6.31.1	Connector Points.....	6-48
6.32	Reusable Mapping Editor	6-49
6.32.1	Reusable Mapping Editor Logical Tab	6-49
6.33	11g-Compatible Mapping Editor	6-50
6.34	Mapping Editor Overview Tab.....	6-50
6.34.1	Definition Tab.....	6-51
6.34.2	Execution Tab	6-51
6.34.3	Scenarios Tab	6-51
6.34.4	Markers Tab	6-52
6.34.5	Memo Tab	6-52
6.34.6	Version Tab.....	6-52
6.34.7	Privileges Tab	6-53
6.34.8	Flexfields Tab.....	6-54
6.35	Mapping Editor Concepts.....	6-54
6.35.1	Component Styles.....	6-54
6.35.2	Component Attributes	6-56
6.36	Mapping Editor Validation Results Tab	6-57
6.37	Mapping Editor Logical Tab	6-57
6.37.1	Component Palette	6-58
6.37.1.1	11g Compatible Dataset Component	6-59
6.37.1.2	Aggregate Component.....	6-59

6.37.1.3	Dataset Component.....	6-60
6.37.1.4	Distinct Component	6-61
6.37.1.5	Expression Component.....	6-62
6.37.1.6	Filter Component.....	6-62
6.37.1.7	Flatten Component.....	6-63
6.37.1.8	Input and Output Signature Components.....	6-64
6.37.1.9	Jagged Component.....	6-64
6.37.1.10	Join Component	6-65
6.37.1.11	Lookup Component	6-67
6.37.1.12	Pivot Component.....	6-69
6.37.1.13	Reusable Mapping Component.....	6-70
6.37.1.14	Set Component.....	6-71
6.37.1.15	Sort Component.....	6-72
6.37.1.16	Split Component.....	6-72
6.37.1.17	Subquery Filter Component.....	6-73
6.37.1.18	Table Function Component.....	6-74
6.37.1.19	Unpivot Component	6-75
6.37.2	Datastore Component	6-76
6.37.3	Mapping Diagram	6-78
6.38	Mapping Editor Physical Tab.....	6-79
6.38.1	Properties	6-79
6.38.1.1	Physical Mapping Design Tab.....	6-79
6.38.2	Source Group and Target Group Properties.....	6-80
6.38.2.1	Execution Unit.....	6-80
6.38.3	Access Point Properties.....	6-81
6.38.4	Physical Component Properties	6-82
6.39	Marker Group Editor.....	6-84
6.39.1	Definition	6-85
6.40	Model Editor	6-85
6.40.1	Definition	6-86
6.40.2	Reverse Engineer	6-87
6.40.3	Selective Reverse Engineering	6-87
6.40.4	Control.....	6-88
6.40.5	Journalizing	6-88
6.40.6	Journalized Tables	6-90
6.40.7	Services	6-90
6.41	Model Folder Editor	6-90
6.41.1	Definition	6-91
6.42	Modify Case Step	6-91
6.43	Modify Run Scenario Step 1	6-91
6.44	Modify Run Scenario Step 2	6-92
6.45	New Mapping Dialog.....	6-92
6.46	New Scenario Dialog	6-92
6.47	Package Editor.....	6-93
6.47.1	Definition	6-94
6.47.2	Execution.....	6-94
6.47.3	Scenarios.....	6-95

6.47.4	Diagram.....	6-95
6.47.4.1	General	6-96
6.47.4.2	Additional Variables	6-98
6.47.4.3	Command	6-98
6.47.4.4	Advanced.....	6-99
6.47.4.5	Options.....	6-99
6.48	Procedure Editor	6-100
6.48.1	Definition	6-100
6.48.2	Tasks	6-101
6.48.3	Options	6-104
6.48.4	Execution.....	6-105
6.49	Project Editor	6-106
6.49.1	Definition	6-106
6.49.2	Markers	6-106
6.49.3	Memo.....	6-107
6.49.4	Version.....	6-107
6.49.5	Privileges.....	6-108
6.49.6	Flexfields	6-108
6.50	Run <Object> Dialog	6-109
6.51	Reference Editor	6-109
6.51.1	Definition	6-110
6.51.2	Columns	6-110
6.51.3	Expression.....	6-110
6.51.4	Behavior	6-111
6.51.5	Control.....	6-111
6.52	Replace Object Dialog	6-112
6.53	Sequence Editor.....	6-112
6.53.1	Definition	6-113
6.53.2	Values	6-113
6.54	Scheduling Editor	6-114
6.54.1	Definition	6-114
6.54.2	Execution Cycle.....	6-115
6.54.3	Variables.....	6-115
6.55	Run... Dialog	6-115
6.56	Scenario Editor	6-116
6.56.1	Definition	6-117
6.56.2	Variables.....	6-117
6.57	Scenario Generation Editor.....	6-117
6.58	Scenario Execution Report Editor.....	6-118
6.58.1	Definition	6-118
6.59	Scenario Variable Dialog.....	6-119
6.60	Shortcut Conflicts Dialog.....	6-119
6.61	Shortcut Editor	6-119
6.61.1	Definition	6-120
6.61.2	Execution.....	6-120
6.62	Sign In Dialog	6-121
6.63	Label Editor.....	6-121

6.63.1	Definition	6-122
6.64	Path Selection Dialog.....	6-122
6.65	Step Report Editor.....	6-123
6.65.1	Definition	6-123
6.66	Sub-Model Editor.....	6-123
6.66.1	Definition	6-124
6.66.2	Distribution.....	6-124
6.67	Synchronizing Label Dialog	6-125
6.68	Target Load Order Dialog	6-125
6.69	User Function Editor	6-125
6.69.1	Definition	6-126
6.69.2	Implementations	6-126
6.70	Variable Editor	6-127
6.70.1	Definition	6-127
6.70.2	Refreshing	6-128
6.70.3	History.....	6-128
6.71	HTTP Analyzer.....	6-129
6.72	HTTP Analyzer Test Window.....	6-129
6.72.1	HTTP Analyzer Test Window	6-130
6.72.2	HTTP Analyzer Request/Response Area	6-130
6.73	HTTP Analyzer Instances Window.....	6-132
6.74	HTTP Analyzer Dialog.....	6-132
6.75	Proxy Settings Dialog	6-133
6.76	Rule Settings Dialog	6-134
6.77	HTTP Keystore Dialog	6-134
6.78	Rename Credentials Dialog.....	6-135
6.79	Select Web Service Description Dialog.....	6-136
6.80	Certificate Error Dialog.....	6-136
6.81	Dimension and Cube Accordion	6-136
6.82	Dimension Editor	6-136
6.82.1	Definition Tab.....	6-137
6.82.2	Levels Tab	6-137
6.82.2.1	Levels of Dimension.....	6-137
6.82.2.2	Level Attributes Table.....	6-137
6.82.2.3	Natural Key Members Table.....	6-138
6.82.2.4	Parent Level References Table	6-139
6.82.2.5	Parent Level Reference Key Members Table	6-139
6.82.3	Hierarchies Tab	6-140
6.82.3.1	Hierarchies Table.....	6-140
6.82.3.2	Hierarchy Members Table.....	6-140
6.82.3.3	Skip Levels Table	6-141
6.83	Cube Editor	6-141
6.83.1	Definition Tab.....	6-141
6.83.1.1	Binding Table	6-142
6.83.2	Details Tab	6-142
6.83.2.1	Dimensions Table	6-142
6.83.2.2	Key Binding Table	6-142

6.83.2.3	Measures Table	6-143
6.84	Dimension Component Properties	6-143
6.84.1	General Properties	6-144
6.84.2	History Properties.....	6-144
6.85	Cube Component Properties	6-145
6.85.1	General Properties	6-145
6.85.2	Target Properties.....	6-145
6.85.3	Connector Points.....	6-146
6.85.4	Cube Component Attribute Properties	6-146
6.85.4.1	Target Properties	6-146

7 Topology Navigator Editors

7.1	Action Editor	7-2
7.1.1	Definition	7-2
7.1.2	Details	7-2
7.1.3	Lines.....	7-3
7.2	Action Group Editor	7-3
7.2.1	Definition	7-4
7.3	Change Work Repository Password Dialog	7-4
7.4	Context Editor	7-4
7.4.1	Definition	7-5
7.4.2	Agents.....	7-5
7.4.3	Schemas.....	7-6
7.5	Create ODI Work Repository Wizard.....	7-6
7.5.1	Step 1 - Specify ODI Work Repository Connection Properties.....	7-6
7.5.2	Step 2 - Specify ODI Work Repository Properties	7-7
7.5.3	Step 3 - Specify the Password of the ODI Work Repository to attach	7-7
7.5.4	Step 4 - Specify the name of the ODI Work Repository to attach.....	7-8
7.6	Data Server Editor.....	7-8
7.6.1	Definition	7-9
7.6.2	JDBC.....	7-13
7.6.3	JNDI	7-14
7.6.4	On Connect/Disconnect	7-15
7.6.5	Properties	7-15
7.6.6	DataSources	7-16
7.6.7	JAgent	7-16
7.7	Datatype Editor	7-17
7.7.1	Definition	7-18
7.7.2	Converted to	7-18
7.7.3	Converted from.....	7-19
7.8	Enterprise Scheduler Dialog.....	7-20
7.9	Export Execution Environment Dialog.....	7-20
7.10	Export Logical Topology Dialog.....	7-21
7.11	Export Master Repository Dialog	7-22
7.12	Export Topology Dialog.....	7-23
7.13	Import Execution Environment Dialog	7-23
7.14	Import Logical Topology Dialog	7-24

7.15	Import Master Repository Dialog.....	7-24
7.16	Import Topology Dialog	7-25
7.17	Index Type Editor	7-25
7.17.1	Definition	7-26
7.18	Language Editor.....	7-26
7.18.1	Definition	7-26
7.19	Language Element Editor	7-27
7.19.1	Definition	7-28
7.19.2	Implementation.....	7-28
7.20	Logical Agent Editor.....	7-28
7.20.1	Definition	7-29
7.21	Logical Oozie Engine.....	7-29
7.21.1	Definition	7-29
7.22	Logical Schema Editor.....	7-30
7.22.1	Definition	7-31
7.23	Master Repository Editor.....	7-31
7.23.1	Definition	7-32
7.24	New Action Line Dialog	7-32
7.25	Oozie Runtime Engine	7-32
7.25.1	Definition	7-33
7.25.2	Properties	7-34
7.26	Physical Agent Editor.....	7-34
7.26.1	Definition	7-35
7.26.2	Datasources.....	7-35
7.26.3	Properties	7-36
7.26.4	Load balancing.....	7-36
7.27	Physical Schema Editor	7-37
7.27.1	Definition	7-38
7.27.2	Context	7-41
7.27.3	Process Definition.....	7-41
7.28	Schedule Dialog.....	7-43
7.29	Select Repository Dialog	7-43
7.30	Select Repositories Dialog.....	7-44
7.31	Server Template Generation Wizard	7-44
7.31.1	Step 1 - Agent Information	7-44
7.31.2	Step 2 - Libraries and Drivers	7-44
7.31.3	Step 3 - Datasources.....	7-45
7.31.4	Step 4 - Template Target and Summary	7-45
7.32	Sub-language Editor	7-45
7.32.1	Definition	7-45
7.33	Technical Environment Dialog	7-46
7.34	Technology Editor.....	7-47
7.34.1	Definition	7-48
7.34.2	SQL.....	7-51
7.34.3	Advanced	7-54
7.34.4	Language.....	7-55
7.35	Test Connection Dialog.....	7-56

7.36	Test Connection Dialog for Enterprise Scheduler	7-57
7.37	Topology Navigator	7-57
7.38	Work Repository Editor	7-58
7.38.1	Definition	7-59

8 Security Navigator Editors

8.1	Change ODI User Password Dialog.....	8-1
8.2	Clean Up Security Settings Dialog	8-2
8.3	Export Security Settings Dialog	8-2
8.4	Import Security Settings Dialog.....	8-3
8.5	Method Editor	8-4
8.5.1	Definition	8-4
8.6	Object Editor	8-4
8.6.1	Definition	8-4
8.6.2	Flexfields	8-5
8.7	Object Instance Editor	8-5
8.7.1	Security.....	8-5
8.7.2	Version.....	8-6
8.8	Password Policies Security Settings Dialog	8-6
8.9	Profile Editor.....	8-6
8.9.1	Definition	8-7
8.10	Profile Method Editor.....	8-7
8.10.1	Definition	8-7
8.11	Roles Editor.....	8-7
8.11.1	Definition	8-8
8.12	Security Navigator.....	8-8
8.13	User Editor.....	8-9
8.13.1	Definition	8-9
8.13.2	Authorization	8-10
8.14	User Connections Dialog	8-10
8.15	User Method Editor	8-11
8.15.1	Definition	8-11

Preface

This preface describes the document accessibility features and conventions used in *Oracle Data Integrator Studio Help*.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Oracle Data Integrator Studio Overview

The resources assembled here list the Oracle Data Integrator online help topics for the Oracle Data Integrator Studio development tool. The Oracle Data Integrator Studio online help is accessed through the Help Center.

To use the online help features effectively, see [Using the Online Help](#).

Oracle Data Integrator navigator-independent editors and dialogs are covered in the following chapters:

- [Main Windows](#)
- [Navigator-Independent Editors, Wizards, and Dialogs](#)

Navigator-specific object editors are described in the following chapters:

- [Designer Navigator Editors](#)
- [Topology Navigator Editors](#)
- [Operator Navigator Editors](#)
- [Security Navigator Editors](#)

Using the Editors

Use the editors to create and modify objects. Oracle Data Integrator provides the following editing features:

- **Undo/Redo:** When typing text in text fields or text areas you can undo or redo changes in editors, dialogs, wizards, and in the Property Inspector with CTRL+Z and CTRL+Y.
- **Autocomplete:** Certain text components and drop-down lists support the autocomplete feature.

Using the Online Help

Oracle Data Integrator online help is accessed through the JDeveloper Help Center. To use the online help features effectively, see the following topics:

- [About the Online Help](#)
- [About the Help Center](#)
- [Opening the Online Help](#)
- [Searching the Documentation](#)

2.1 About the Online Help

The JDeveloper Help Center comprises two windows: the help navigator and the help topic viewer. For more information, see [About the Help Center](#).

The following types of content are available in the Oracle Data Integrator help:

- *Context sensitive online help topics*, which open when you press **F1** or click **Help** in the ODI Studio user interface.
- *Procedural and conceptual topics*, which can be opened from the Content navigator or links in related help topics.

From the Help Center, you can also access additional documentation on Oracle Technology Network (OTN).

The Help Center search feature lets you search for a word or a phrase in the installed documentation, OTN, and the Fusion Middleware Documentation Library. See [Searching the Documentation](#) for more information.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

2.2 About the Help Center

The Help Center enables you to browse the table of contents, locate relevant topics in the dynamic help links lists, and do a full text search of installed and online content. It also provides a **Favorites** navigator for saving links to frequently referenced topics. The Help Center comprises two windows, the help navigator on the left and the help topic viewer on the right.

The main Help Center toolbar includes the items listed in the following table.

Name	Description
Keep Help Center on Top	Use to keep the Help Center on top of all other open windows. The keyboard shortcut is Alt+K.
Navigators	Use to open the Help Center windows you have previously closed.
Search	Use to perform a search in the installed documentation, Oracle Technology Network (OTN), and the Fusion Middleware and Database Libraries.

The left pane of the Help Center displays tabs that help navigate through content:

- **Contents** - Displays the table of contents for all installed content in the help system, including traditional online help and the developer guide.
- **Favorites** - Displays the help topics and external links you have saved for quick retrieval.
- **Dynamic** - Displays links to a set of relevant help topics based on the current selection and IDE component in use.

The right pane of the Help Center displays selected content and search results:

- **Help content viewers** - Displays the selected online help and developer's guide contents. Multiple tabbed pages open for selected content.
- **Search results** - Displays the results of a full-text search.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

2.3 Opening the Online Help

To open the online help, use any of these methods:

- Press **F1** at any time to display context-sensitive help.
- Click **Help** in any dialog or the Help icon in any window to display context-sensitive help.
- From the main menu, select **Help > Search**.
- From the main menu, select **Help > Table of Contents**.
- From the main menu, select **Help > Help Favorites**.
- From the main menu, select **Help > Dynamic Help**.
- From the **Start Page**, click any link with a book or help topic icon.

The Help Center opens and the selected help page is displayed.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

2.4 Searching the Documentation

You can perform a full-text search on the documentation installed as online help, Oracle Technology Network (OTN), and the Fusion Middleware and Database Online Documentation Libraries.

To do a full-text search from the Help Center:

1. Select **Search** in the Help menu. The Help Center opens.
2. Enter the word or phrase you are searching for in the Search field.
3. Optionally, click the Search Options icon to define search criteria and narrow the search results.

By default, **Local Documentation** and the **Fusion Middleware library** are selected.

Note: The search criteria set in the Search Options menu only apply to the online help search.

4. Set the other search options as needed; these apply only to the online help search.
5. Click **Start search** or press **Enter**.

The Search Results page opens in the help viewer area, with the titles and sources of each matching document, as well as the beginning text.

6. Double-click the title of the help page to select it.

Each help page displays in a separate page and the Search Results page remains available. OTN and Documentation Library pages open in your default browser.

Using the Boolean Expressions option

You can use Boolean expressions in your search. The Boolean expressions search option is a recursive tree structure for expressing search criteria involving boolean expressions. The BooleanExpression is based on the following grammar:

```
BooleanExpression ::
    BooleanExpression AND BooleanExpression
    BooleanExpression OR BooleanExpression
    BooleanExpression NOT BooleanExpression
    BooleanExpression + BooleanExpression
    BooleanExpression - BooleanExpression
    + BooleanExpression
    - BooleanExpression
    NOT BooleanExpression
    StringExpression (base case)
```

The tree is built in such a way that StringExpressions are the leaf items and the expressions occupy the branch items. For example Oracle Discoverer AND Browser OR Query are parsed into:

```

      AND
     /  \
Oracle Discoverer  OR
                  /  \
                  Browser  Query
```

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

Main Windows

This chapter lists the help topics for Oracle Data Integrator Studio main windows. Main windows are Oracle Data Integrator Navigator-independent and used for managing and working with different object types and resources.

This chapter includes the following topics:

- [About Oracle Data Integrator Studio](#)
- [About the Online Help](#)
- [Check for Updates Wizard](#)
- [Find ODI Object Navigator](#)
- [Log Window](#)
- [Manage Libraries Dialog](#)
- [New Gallery](#)
- [Preferences Dialog](#)
- [Property Inspector](#)
- [Save Files](#)
- [Save to File](#)
- [Thumbnail](#)
- [Unable to Connect Dialog](#)
- [Unable to Save Dialog](#)
- [Update Center Dialog](#)
- [Windows Dialog](#)

3.1 About Oracle Data Integrator Studio

Use to obtain the Oracle Data Integrator version number, as well as detailed information about the components, properties, and installed extensions.

To open the About Oracle Data Integrator Studio window, select **About** in the Help menu. The About Oracle Data Integrator Studio window includes the following tabs and options:

About

Displays the detailed version and build number of your Oracle Data Integrator Studio.

Version

Displays the version information for the installed components, including the Java platform and Oracle IDE.

Properties

Displays the current value of each property set in the Oracle Data Integrator Studio configuration file.

External Components

Displays a list of Java libraries loaded by the Studio, the version of the library, and path from where the Java file is taken.

Extensions

Displays details about each installed extension, its identifier, version number, and if it is currently loaded.

Export

Displays the options in the Export drop-down list that enable you to export the details displayed in the About... window.

Use **Save to File...** to open the Save dialog box where you can enter a file name and select a location in which to save all the details displayed on the About pages.

Use **Copy to Clipboard** to copy all the details displayed on the About pages.

Use **Copy Build to Clipboard** to copy only the build name and number displayed on the About page.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

3.2 About the Online Help

The Help Center comprises two windows: the help navigator and the help topic viewer. More

The following types of content are available:

- *Context sensitive online help topics*, which open when you press **F1** or click **Help** in a dialog or wizard, or click the help icon in a wizard.
- *Procedural and conceptual topics*, which can be opened from the Content navigator or links in related help topics.
- *Developer guides*, which provide end-to-end information for developing applications with specific technologies.

From the Help Center, you can also access additional documentation on Oracle Technology Network (OTN).

The *Help Center search feature* lets you search the installed documentation, the documentation available from OTN, and the Fusion Middleware Documentation Library.

You can also customize the way you view content.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

3.3 Check for Updates Wizard

Use to browse, download, and install global ODI objects. ODI objects are made available for distribution by Oracle and by ODI users in separate Update Centers. Each ODI object is packaged into an update bundle and placed in an Update Center.

Note: Only global user-functions, global KMs, and custom mapping components can be obtained using the Check for Updates wizard.

The Check for Updates wizard has the following steps:

- [Step 1: Select update source](#)
- [Step 2: Select updates to install](#)
- [Step 3: View download progress](#)
- [Step 4: View summary](#)

3.3.1 Step 1: Select update source

Use to select the source that has the required updates.

Properties	Description
Search Update Centers	Select to search for Update Centers. By default, this is selected. Note: An internet connection is required for web-based Update Centers.
Proxy Settings	Click to define the proxy settings for your internet connection. Note: If you are unable to connect to the internet, an " Unable to Connect Dialog " is displayed. You must verify the proxy settings in this dialog and retry.
Add	Click to add additional file-based or web-based Update Centers.
Remove	Click to remove the selected Update Center. Note: Only newly added Update Centers can be removed.
Edit	Click to edit the selected Update Center.
File Name	Enter or browse to the location of the downloaded ODI object that you want to install.

3.3.2 Step 2: Select updates to install

Use to select the ODI objects that you want to install.

Properties	Description
Available Updates	Select the ODI object to be installed.
Search	Enter the name or a part of the name of the ODI object to reduce the number of ODI objects displayed in this list.
Show Upgrades Only	Select to view newer versions of the ODI object that you have already installed.
Available Updates Tree	List of available ODI objects.

Properties	Description
Information Pane	Information regarding the selected ODI object.
Expand All	Click to expand the list of ODI objects.
Collapse All	Click to collapse the list of ODI objects.

3.3.3 Step 3: View download progress

Use to transfer the selected ODI objects.

Click **Stop** to cancel the process.

3.3.4 Step 4: View summary

Use to view a summary of the downloaded ODI objects

Click **Finish** to start the installation.

Note: After importing a Global KM or user-function, you must click **Refresh** to see it in the Designer navigator. After importing a Custom Mapping Component, you must exit and restart ODI to see it in the mapping components palette, if a mapping has already been opened in the current ODI session.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

3.4 Find ODI Object Navigator

Use the Find ODI Object Navigator to search for ODI objects.

These features are available:

- Open any currently closed navigator, or bring a currently open navigator to the foreground, using the window menu on the tool bar.
- Move, size, float, minimize, maximize, restore, or close the Find ODI Object navigator using the context menu. Right-click the **Find ODI Object** tab or press Alt+Minus to display the context menu.

The Find ODI Object Navigator has the following tabs:

- [Standard](#)
- [Advanced](#)

Select **Match Case** if you want to consider case when matching search text against the entered value.

Search Results

If an object is found, it is listed in the pane at the bottom of the Find ODI Object navigator. Double-click the search result to open it in the appropriate viewer for editing. If no object is found, the pane remains empty.

Click **Load Search Criteria** to browse for an XML search criteria file and reuse these search options.

Click **Save Search Criteria** to save the current search criteria in an XML file.

Click **Search** to start the search.

3.4.1 Standard

Use to perform a search with standard search criteria.

The Standard tab has the following sections:

- [Simple Search](#)
- [Favorite Criteria](#)
- [Date and User](#)

Simple Search

Properties	Description
Search Scope	Select the search scope from the list.
Object Type	Select the object type from the list.
Name	Enter the name of the object to search. You can use the wildcard % to return a number of matching objects.

Favorite Criteria

Enter the favorite search criteria.

Date and User

Search for an object based on its creation date or last modified date, and narrow search results by user.

3.4.2 Advanced

Use to perform a search with advanced search criteria.

Select the search criterion and enter the value to search.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

3.5 Log Window

Use to view specific feedback from the Integrated Development Environment (IDE).

To display the context menu for the contents of the Log window, right-click within the window. To bring up the context menu for the Log window as a window, right-click on the Messages-Log tab. From this context menu, you can:

- Copy the contents of the window.
- Select all data within the window.
- Wrap the text in the window.
- Clear the contents of the window.
- Search.
- Close the window.

Other actions may be available within the tabbed sections generated by specific processes.

From the context menu for the window, you can:

- Close the window.
- Close all other tabs but for the currently selected tab.
- Close all tabs within the window.

Related Topics

"How to Use the Log Window" in the "Using Web Services" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

3.6 Manage Libraries Dialog

Use to view existing libraries and create new ones.

Libraries

Properties	Description
Libraries List	Lists the currently defined system and user libraries. Select a library to display its definitions in the fields to the right.
New	Creates a new library.
Load Dir	Adds a directory containing library tables.
Remove	Removes the currently selected directory or file. You cannot remove system directories or files.
Library Name	Displays the name of the library to be created or edited.
Deployed by Default	Select if this library is deployed by default. A library that is deployed by default is assumed to be part of the application. Libraries used by single applications are generally deployed with the application. Libraries meant to be shared across applications are generally not.
Class Path	Displays the class path associated with the library.
Source Path	Displays the source path associated with the library.
Doc Path	Displays the doc path associated with the library.
Add Entry	Adds an entry to the currently selected path type in the paths list.
Add URL	Specifies a URL for the doc path.
Remove	Removed entries from the paths list.

Java SE Definitions

Use to view existing Java SE definitions and create new ones.

Properties	Description
Libraries	Lists the currently defined system and user definitions. Select a definition to display its contents in the fields to the right. Predefined definitions are not editable.
New	Creates a new definition.
Load Dir	Adds a directory containing definition files.

Properties	Description
Remove	Removes the currently selected directory or file. You cannot remove system directories or files.
Java SE Executable	Enter the executable associated with the installed Java SE or click Browse to navigate to its location. When this value is entered, the following three fields automatically display the default values for the paths associated with the executable.
Java SE Name	Displays the current name for the Java SE. The name is used only as an identifier. Enter a new identifying name if you wish. You can reorder the entries using the Move Up and Move Down buttons.
Location	Shows the types of paths.
Class Path	Displays the class path associated with the definition.
Source Path	Displays the source path associated with the definition.
Doc Path	Displays the doc path associated with the definition.
Add Entry	Adds an entry to the currently selected path type in the paths list.
Add URL	Specifies a URL for the doc path.
Add Macro	Opens the Path Macro Selection dialog, where you can select a macro for the default run target.
Remove	Removes entries from the paths list.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

3.7 New Gallery

Use to create Oracle Data Integrator Studio Master and Work login credentials and to create or import a master repository.

To use the Gallery, select a category in the **Categories** list and double-click the item you wish to create in **Items** list.

Categories

Select a category to display the object types associated with that category.

At the end of the list of categories is **All Items**, which lists all the available items in alphabetical order.

Items

Displays the items associated with the selected category. You can filter the items shown.

Show All Descriptions

Select to show a description for each item. By default, this is deselected.

Related Topics

"Creating a new Master Repository using a previous Master export" in the "Exporting and Importing" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

3.8 Preferences Dialog

Use to set default display options and other general aspects of Oracle Data Integrator Studio.

The Preferences dialog includes the following pages:

- [Preferences Dialog - Environment Page](#)
- [Preferences Dialog - Environment - Log Page](#)
- [Preferences Dialog - Code Editor Page](#)
- [Preferences Dialog - Code Editor - Caret Behavior Page](#)
- [Preferences Dialog - Code Editor - Display Page](#)
- [Preferences Dialog - Code Editor - Fonts Page](#)
- [Preferences Dialog - Code Editor - Line Gutter Page](#)
- [Preferences Dialog - Code Editor - Syntax Colors Page](#)
- [Preferences Dialog - Credentials Page](#)
- [Preferences Dialog - Diagrams - Annotation Page](#)
- [Preferences Dialog - ODI - System Page](#)
- [Preferences Dialog - ODI - System - Agent Page](#)
- [Preferences Dialog - ODI - System - Concurrent Development Page](#)
- [Preferences Dialog - ODI - System - Groovy Page](#)
- [Preferences Dialog - ODI - System - Reports Page](#)
- [Preferences Dialog - ODI - System - Scenarios Page](#)
- [Preferences Dialog - ODI - User Interface Page](#)
- [Preferences Dialog - ODI - User Interface - Designer Page](#)
- [Preferences Dialog - ODI - User Interface - Operator Page](#)
- [Preferences Dialog - ODI - User Interface - Topology Page](#)
- [Preferences Dialog - ODI - User Interface - Versioning](#)
- [Preferences Dialog - ODI - User Interface - Versioning - Subversion Page](#)
- [Preferences Dialog - Web Browser and Proxy - Web Browsers](#)

3.8.1 Preferences Dialog - Environment Page

Use to customize the graphical user interface (GUI) and other preferences of your Oracle Data Integrator Studio.

The Preferences Dialog displays the preferences associated with your login name.

Undo Level

Enter a number that determines the number of times Oracle Data Integrator must perform undo operations. The default value is 50.

Navigation Level

Enter a number that determines the number of times you can go back and forth on browsing history, using the **Back** and **Forward** navigation buttons shown in the following table.

Back	
Forward	

The default value is 20.

Look and Feel

Select a JGoodies look and feel from the drop-down list. The new look and feel is reflected in the IDE after you restart Oracle Data Integrator. The default is `Oracle`.

Reset Skipped Messages

Click to see all message types, including those that you have previously opted not to see.

3.8.2 Preferences Dialog - Environment - Log Page

Use to set default display options and other general aspects of Oracle Data Integrator's behavior.

Save Logs to File

Select to save the logs generated by Oracle Data Integrator to log files. For example, you could select **Save Logs to File** to save a large amount of log data for later searches or analysis.

When you select this option, all output to the Log window from running, debugging, and profiling operations is saved to log files. The filename of each log file indicates the operation, the project name, and the date and time the information was generated. The filename is generated in the `operation_projectname_yyyymmddhhss.log` format, where the date and time is given in year, month, minute, hour, and second.

When you clear or close the Log window, the remaining information is automatically saved to the current log file on disk, including the information in the buffer. Select the `<name of the option>` option in the Tools Settings page of the Edit Run Configuration dialog box to clear the Log window before running it. This results in a new log file being created with every run and this log file mirrors what appears in the Log window. If that option is not selected, then every new run of a project in the same session concatenates in one log file, just as it does in the Log window.

Note that after you have selected this option, the output from all run operations are saved to log files. The number and length of these files can quickly become quite large.

Log Directory

Enter a directory for the saved log files. The directory is automatically created.

Maximum Log Lines

Enter a number that determines the maximum number of lines to be displayed in the Log window. After the maximum is exceeded, the previous information is cleared from the window. The default value is 3000.

Hyperlink Color

Select a color to be used in the Log window for all links.

Program Output Color

Select a color to be used in the Log window for all output from programs run.

Program Error Color

Select a color to be used in the Log window for displayed errors.

Program Input Color

Select a color to be used in the Log window for all input to programs run.

3.8.3 Preferences Dialog - Code Editor Page

Use to customize editor look and feel, general behavior, and Code Insight and Java Insight options.

Auto-Indent New Lines

Select to direct the editor to auto-indent a new line when you press Enter at the end of a line. The new line will automatically be indented at the same initial indentation as the line preceding it. Selected by default.

Perform Block Indent or Outdent for Selections

Select to direct the editor to perform a block indent or block outdent on a selection when you press Tab or Shift+Tab respectively. With this option selected, when you press Tab on a selected block of text, the entire block will be indented to the current tab size. Shift+Tab on the same block would outdent it, as a block, to the current tab size. Selected by default.

Use Smart Home

Select to contextualize the cursor's understanding of home (the beginning of the line).

With this setting selected, pressing Home positions the cursor at the start of the line after any leading spaces or tabs. Pressing Home again repositions the cursor at the start of the line before any leading spaces or tabs. Continuing to press Home toggles the cursor between these two locations.

With this setting deselected, pressing Home simply places the cursor at the start of the line. Selected by default.

Use Smart End

Select to contextualize the cursor's understanding of end of line.

The behavior is analogous to that for smart home, except that the cursor responds to the End key, and its behavior regarding the end of the line and any trailing spaces is altered. Deselected by default.

Use Jump Scrolling for Keyboard Navigation

Select to implement jump scrolling, which involves behavior of the keyboard arrow keys.

With this setting selected, when you navigate offscreen using the keyboard arrow keys, the editor view will "jump" to recenter the cursor location in the middle of the editor view.

With this setting deselected, the editor view will scroll the editor view the minimum amount to bring the cursor back into view. Deselected by default.

Use Change of Case As Word Boundary

Select to have change of case regarded as the boundary of a word, for example, when double-clicking to select a word. Deselected by default.

Enable Cut or Copy of Current Line with No Selection

Select to apply all cut and copy operations to the current line whenever there is no text selection in the editor. Selected by default.

Automatically Copy Paste Imports

Select to automatically add imports when references are introduced to classes that have not yet been imported. Selected by default.

Adjust Indentation When Pasting

Select to correct the indentation of a pasted in item that includes indentation. Selected by default.

Reformat Code Block When Pasting

Select to have the code correctly reformatted when you paste it into a new location. Deselected by default.

Escape When Pasting In String Literals

Select to include correct escape characters in pasted in string literals. Selected by default.

Use Middle Mouse Button Scrolling

Select to enable scrolling using the middle button of the mouse. Deselected by default.

Warn When Opening Files Containing Long Lines

Select to enable warning while opening files that contain long lines. Selected by default.

Line Length (in Characters)

Specify the length of line for which you would like to get a warning message. The default line length is 10,000 characters.

3.8.4 Preferences Dialog - Code Editor - Caret Behavior Page

Use to configure the shape, color, and blinking characteristics of the caret (or cursor) in the editor.

Enable Caret Blinking

Select to describe a blinking character in the editor. Selecting this option enables the slide bar below. Selected by default.

Caret Blink Rate

Slide the bar to set the rate of blinking (in seconds) for the caret. The valid range is from 0.1 seconds to 2.0 seconds. The default setting is 0.5 seconds.

Insert Mode Caret Shape

Select a shape from the drop-down list for the caret when the editor is in insert mode. Selections provide vertical lines and underbars of various thicknesses, as well as both a hollow and a solid box. The default setting is `Double Width Vertical Bar`.

Overwrite Mode Caret Shape

Select a shape from the drop-down list for the caret when the editor is in overwrite mode. Selections provide vertical lines and underbars of various thicknesses, as well as both a hollow and a solid box. The default setting is `Solid Box`.

Caret Color

Select a color for the caret. The default setting is black.

3.8.5 Preferences Dialog - Code Editor - Display Page

Use to set the general display features for the source editor.

Enable Text Anti-Aliasing

Select to allow smooth-edged characters where possible. Selected by default.

Show Whitespace Characters

Select to render spaces, new lines, carriage returns, non-breaking spaces, and tab characters as alternate visible characters. Deselected by default.

Select Breadcrumbs

Select to show the breadcrumb bar, which shows the hierarchy of nodes from the current caret position up to the top of the file. Hover the mouse cursor over a node to display information about the node. Selected by default.

Show Scroll Tip

Select to enable the Tip window that displays the methods in view while scrolling. Selected by default.

Show Code Folding Margin

Select to show the code folding in the right margin. Code that can be folded is marked by a - in the margin. Code that has been folded is shown by a + in the margin. Selected by default.

Blocks can now be unfolded by double clicking directly on the folded section or by dragging an object onto a folded block, which is useful when adding components or text to a highly nested document.

You can set the blocks to be initially folded in the Java Source Editor in the Code Editor - Java - Folding page of the Preferences dialog.

Margin Indicates Current Block

Select to highlight the active folding block. Expanded and collapsed blocks highlight after a transition by briefly fading their background color. This is useful in documents where nested blocks can be hard to distinguish. Selected by default.

Show Visible Right Margin

Select to render the right margin (gray line) visible in the source editor. The visible right margin generally is used to keep lines short, so that they can print more easily without having lines wrap or be truncated. You can specify the column at which the right margin is displayed in the Code Editor - Code Style page of the Preferences dialog. Selected by default.

Right Margin Color

Select a color for the margin, when it is visible. The default setting is gray.

Enable Automatic Brace Matching

Select to enable automatic brace matching in the source editor based upon the cursor position. Set the options for brace matching on the Java page of the Preferences dialog. The editor will check braces, brackets, and parentheses. Selecting this option enables the brace matching delay slider, below. Selected by default.

Highlight Enclosing Block

Select to highlight the enclosing block. Selected by default.

Highlight Enclosing Parentheses

Select to highlight the enclosing parentheses. Selected by default.

Brace Matching Speed

Slide the bar to set the delay (in seconds) before matching braces, brackets, or parentheses are highlighted. The valid range is from 0.1 seconds to 2.0 seconds. The default setting is 0.3 seconds.

3.8.6 Preferences Dialog - Code Editor - Fonts Page

Use to set the fonts for the source editor.

Font Name

Select the name of the font from the drop-down list. By default, Oracle Data Integrator will show all system fonts here. To limit the selection to fixed-width fonts, select **Display Only Fixed-Width Fonts**. The default value is `DialogInput`.

Display Only Fixed-Width Fonts

Select to limit the fonts available in this dialog page to fixed-width fonts. If you select this, the first time you do, Oracle Data Integrator will run a check on your system fonts to detect which ones are fixed-width fonts. Once it has finished, the selections available under **Font Name** will change accordingly.

This system check is required only once. Subsequently, if you deselect or select this check box, the fonts displayed under Font Name toggle between all fonts and fixed-width fonts. Deselected by default.

Font Size

Select the font size. The default is 12.

Sample Text

Type in sample text for display in the field below. The default is The quick brown fox jumps over the lazy dog.

Sample Text in Selected Font and Size

Displays the text you typed in **Sample Text** in the font style and size you selected.

3.8.7 Preferences Dialog - Code Editor - Line Gutter Page

Use to set the appearance of line gutters for the source editor. You can set several preferences for the look and feel of the source editor.

Show Line Numbers

Select to display line numbers in the line gutter to the left of the code in the source editor. Deselect to hide line numbers. Deselected by default.

Enable Line Selection by Click Dragging in Gutter

Select to be able to highlight any number of consecutive lines in the editor by clicking in the gutter and dragging the cursor without releasing the mouse button. Selected by default.

Use Look and Feel Colors

Select to use the colors defined for the current look and feel.

Use Editor Colors

Select to use the colors defined for the editor. This is the default setting for line gutter colors.

Use Custom Colors

Select to define your own colors for the line gutters and line numbers.

3.8.8 Preferences Dialog - Code Editor - Syntax Colors Page

Use to control the colors and font style used by the source editor.

Scheme

A scheme is a collection of all the settings for all available styles saved under an identifying name. To activate a scheme for the source editor, select it from this drop-down list. The list contains all the initial schemes bundled with Oracle Data Integrator and any custom schemes that you create. All initial schemes are editable. To make changes to any previously existing scheme, select it first. To create a new scheme based upon an existing scheme, select it first. You will always start with some scheme selected, even if it is just Default. As soon as you begin making changes to settings on this page, the current selection in the Scheme list goes blank, indicating that you have modified an existing scheme and that those settings are not currently saved under a particular scheme. Likewise, if you delete a scheme, the current selection in this list goes blank, indicating that the settings that once constituted the recently deleted scheme are not currently saved as a scheme.

Create a new scheme by clicking **Save As** and assigning a new name. Once you have dismissed the Save Scheme dialog, a newly defined scheme appears in the Scheme list and all the changes you have made, from the first change until the moment you click **Save As**, are reflected in the new scheme.

Edit an existing scheme by clicking **Save As** and assigning the same name. All the changes you have made, from the first change until the moment you click **Save As**, are reflected in the edited scheme.

Remove an unwanted scheme by selecting it and clicking **Delete**.

Language

Select a language category of Oracle Data Integrator from the drop-down list. The list of elements in **Available Styles** changes in response to this selection. The default setting is Default. Under Default, the available styles include a number of highlight styles. Any item in the Default list not preceded by the word "Default" is a highlight style.

Style Sample

Type in sample text (or accept the default) to view the currently defined font style and foreground and background colors for any style other than a highlight style.

Available Styles

Select the element you wish to modify from the list of styles available for the selected language category. Then use the other controls on the page to configure how the selected style is displayed. Note that the style Default plain text is special. It is the root of all styles, and controls the background color used by the editor. If you change the background color of this style, the background color used by the editor will be changed as well.

Font Style

Select a font style to be applied to the element style currently selected in the Available Styles list. The current font definition for that element appears here when the element is selected. Select Use default to instruct the editor to select a default appropriate for the style. Selecting this option for the Java keyword style, for instance, means that the editor will apply the font style currently defined for Default keyword, which you can also customize.

Foreground

Select a foreground color for the element selected in the Available Style list. Make your selection by clicking on the color button, which opens a palette for you to choose from. The color currently defined for an element shows here when that element is selected.

Background

Select a background color for the element selected in the Available Styles list. Make your selection by clicking on the color button, which opens a palette for you to choose from. The color currently defined for an element shows here when that element is selected.

3.8.9 Preferences Dialog - Credentials Page

Use to provide credentials to be used when the HTTP Analyzer secured services or applications, for example, web services secured by policies. More

Oracle Data Integrator has a set of preconfigured credentials, HTTPS Credential, and this is always present. You cannot delete or edit HTTPS Credential, but you can copy it to create a new credential of the same type.

Once you have configured credentials in this page, you can choose which to use in the HTTP Analyzer Test window.

3.8.10 Preferences Dialog - Credentials - HTTP Keystore Tab Page

HTTPS encrypts an HTTP message prior to transmission and decrypts it upon arrival. It uses a public key certificate signed by a trusted certificate authority. When Integrated WebLogic Server is started for the first time, it generates a `DemoIdentity` that is unique to your machine, and the key in it is used to set up the HTTPS channel. Use this tab to enter details of the certificate keystore for the client and for the server.

The client keystore identity is used for configuring HTTPS. The server keystore identity is used when the HTTP Analyzer is acting as a server; it is not used when connecting to a remote server.

For more information about keystores and keystore providers, see *Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server*.

When the default credential `HTTPS Credential` is selected, you need to specify the keystores that Oracle Data Integrator and the HTTP Analyzer should use when handling HTTPS traffic. Two keystores are required to run the HTTP Analyzer: one containing the certificates of all the hosts to be trusted by Oracle Data Integrator and the Analyzer (client trust) when it makes onward connections, the other containing a key that the Analyzer can use to authenticate itself to calling clients (server keystore). The client keystore is only required when mutual authentication is required.

Element	Description
Client Keystore	Enter or browse to the location of the client keystore.
Client Keystore Password	Enter the password for the client keystore.
Client Private Key Password	Enter the client private key password.
Client Trusted Certificate Keystore	Enter, or browse to, the location of the client certificate keystore. The default is <code>jdeveloper-install/Oracle/Middleware/wlserver_10.3/server/lib/DemoTrust.jks</code> .
Client Trusted Keystore Password	Enter the password for the client keystore. The default is <code>DemoTrustKeyStorePassPhrase</code> .
Append default CA certificates	Select to add the CA certificates. Selected by default.
Server Keystore	The keystore on the server. The default is <code>jdeveloper-install/Oracle/Middleware/wlserver_10.3/server/lib/DemoIdentity.jks</code> .
Server Keystore Password	Enter the password for the server keystore. The default is <code>DemoIdentityKeyStorePassPhrase</code> .
Server Private Key Password	Enter the private key password. The default is <code>DemoIdentityPassPhrase</code> .
Default	Click to return to the default settings.

3.8.11 Preferences Dialog - Credentials - Username Token Tab Page

Username token is a way of carrying basic authentication information. Use this tab to supply a token based on username/password to provide authentication. You can change the default username and password values.

Element	Description
Username	Enter the username. The default is <code>weblogic</code> .

Element	Description
Password	Enter the password. The default is <code>weblogic1</code> .

3.8.12 Preferences Dialog - Certificates - X509 Certificates Tab Page

X509 is a PKI standard for single sign-on, where certificates are used to provide identity, and to sign and encrypt messages. Use this tab to supply details of an X509 certificate. When you supply a valid keystore and the password for the keystore, the client key aliases are populated.

In order to use the client key and certificate a valid key password must be entered for the selected key alias.

If Oracle Data Integrator has any problems finding and opening the keystore, error messages will be displayed.

Element	Description
Client Keystore	Enter, or browse to, the path of the keystore that has the necessary key and certificate information configured.
Client Keystore Password	Client Keystore Password.
Client Key Aliases	Select the alias for the client key to be used to sign the authentication tokens.
Client Key Password	Enter the password for the client key.
Server Keystore	Enter, or browse to, the path of the keystore that has the necessary key and certificate information configured.
Server Keystore Password	Enter the password for the keystore.
Server Certificate Aliases	Select the alias for the server certificate.

3.8.13 Preferences Dialog - Credentials - STS Configuration Tab Page

A Secure Token Service (STS) is a web service that issues and manages security tokens over HTTPS. Use this tab to enter the Security Token Server provider URL and optionally a policy URL.

Note: The client truststore must contain the server public key, otherwise when the HTTP Analyzer requests the SAML token it will fail.

Element	Description
Security Token Service URL	Enter the endpoint URL for the STS provider.
STS Policy URL	(Optional) The default is the SOAP body encryption policy <code>policy:Wssp1.2-2007-EncryptBody.xml</code> . Accept the default, or enter or browse to a new policy.

3.8.14 Preferences Dialog - Diagrams - Annotation Page

Use to specify diagram display properties for this type of diagram only.

Edit Preferences for

Select the required object type. Choose to edit the preferences that apply to the whole diagram, or to edit the preferences that apply to only a specific object type. The available object types are listed in the drop-down list:

- **All Nodes**
Select to apply preference changes to all nodes for this type of diagram only.
- **All Edges**
Select to apply preference changes to all edges for this type of diagram only.
- **Attachment**
Select to apply preference changes to attachments for this type of diagram only.
- **Dependency**
Select to apply display preferences to all dependencies for this type of diagram only.
- **Group**
Select to apply display and color/font preferences to all groups for this type of diagram only.
- **Link**
Select to apply color/font preferences to all links for this type of diagram only.
- **Note**
Select to apply color/font preferences to all notes for this type of diagram only.

Display Options

- **Presentation**
Select to display the diagram in an expanded, compacted, or symbolic view layout.
- **Show Title**
Select to display the title of the diagram.
- **Title Alignment**
Select to align the title left, center, or right.

Graphical Options

- **Fill Color**
Select the fill color for the diagram.
- **Font**
Select the font for the diagram.
- **Font Color**
Select the font color for the diagram.
- **Line Color**
Select the line color for the entire diagram.

3.8.15 Preferences Dialog - Http Analyzer

Use to set preferences for the HTTP Analyzer.

You can use the default settings, or you can set rules and filters for each listener. You can also define outgoing HTTP proxy details for each configured proxy item.

The HTTP Analyzer can be used to examine, or replay the messages of anything that makes use of the HTTP protocol.

Default Proxy For JDeveloper And New Processes

Select to set the selected proxy as the default. Selected by default.

Host

Displays the host name of the selected proxy. The default is localhost.

Listen On Port

Displays the port of the selected proxy. The default is 8099.

Override Outgoing Proxy (Replace IDE Settings)

Use to override the proxy settings in the Web Browser/Proxy Page of the Preferences dialog. Select this to enable the Configure Proxy button.

Rules

Lists the available rules used to run HTTP messages through the HTTP Analyzer.

3.8.16 Preferences Dialog - ODI - System Page

Use to set system options and related aspects of Oracle Data Integrator Studio.

Default context for Designer

Specify the context to be used when editing designer objects.

Default context for Execution

Specify the context to be selected by default in the Execution dialog when executing any object. If an invalid context name is specified, the default context in Designer Navigator is used.

Default path for generation of Data Services

Enter a default directory for the generated Data Service. The generated source code and the compiled Web Service are placed in this location. This directory is a temporary location that can be deleted after generation.

Delete Linked Sessions

Specify whether to automatically delete linked sessions when deleting a scenario in the Scenarios accordion of Operator Navigator, a Mapping, Procedure, Package, or Variables. The default value is Ask, which means users are prompted to delete the linked session.

Export default Java encoding

Enter the default Java encoding. The default value is ISO8859_1.

Export default XML charset

Enter the default charset encoding. The default value is ISO-8859-1.

Export default version

Enter a number that determines if versioned objects are exported during the master repository export. The default value is 1.0, which means versioned objects are exported. If 0 is specified, versioned objects are not exported.

Never transform non-ASCII characters to underscores

Select to never transform non-ASCII characters to underscores during an export or alias generation. Enable this option only on platforms that fully support your encoding.

JDBC connection timeout

Specify the timeout in seconds for database connections. The default value is 7,200.

Process Model Datastores only

Specify whether to generate DDL code for datastores that do not exist in this model. The default value is Ask, which means a confirmation message is displayed.

Query Buffer size

Specify a number that determines the size of the cache for prepared statements (Queries) issued on the repositories. The default value is 0. This setting applies only to repositories on Oracle instances. Oracle Data Integrator Studio must be restarted for changes to take effect.

Rollback all open transactions on step failure

If selected, all open transactions are rolled back when a session step fails, otherwise open transactions are rolled back in the end when session completes in Error status.

3.8.17 Preferences Dialog - ODI - System - Agent Page

Use to specify the default agent.

Default Agent

Specify the agent to be selected by default in the Execution dialog when executing any object. If an invalid agent name is specified, the local agent is used.

3.8.18 Preferences Dialog - ODI - System - Concurrent Development Page

Use to set concurrent editing options.

Check for concurrent editing

Select to check whether changes have been made to the same object by other users when saving changes to the object. If concurrent changes have been made by other users, the object cannot be saved.

Lock object when opening

Specify whether to lock an object when opening for editing. The default value is Ask, which means users are prompted to lock the object when opening.

Unlock object when closing

Specify whether to unlock an object when closing after editing. The default value is *Ask*, which means users are prompted to unlock the object when closing.

3.8.19 Preferences Dialog - ODI - System - Groovy Page

Use to specify an additional Groovy execution classpath.

You can define a Groovy execution classpath in addition to all classpath entries available to Oracle Data Integrator Studio before executing the Groovy script.

In the Classpath field, enter the classpath and click **OK**.

Note: You do not need to restart Oracle Data Integrator Studio after adding or changing the classpath.

3.8.20 Preferences Dialog - ODI - System - Reports Page

Use to set report options.

Default PDF generation directory

Enter the directory where the generated PDF file will be saved by default when you generate a report.

Directory for saving your diagrams (PNG)

Enter the directory where the PNG file will be saved by default when you print a model diagram with Common Format Designer.

PDF Viewer

Enter the complete path of the program used to view generated PDF files, including the filename. This is necessary to use the "Open file after generation" option.

3.8.21 Preferences Dialog - ODI - System - Scenarios Page

Use to set scenario options.

Scenario naming convention

Enter the scenario naming convention. This applies to internal agents only. Use this setting to define the default naming pattern for new scenarios created for objects using Oracle Data Integrator Studio or tools. The following tags can be used in the pattern:

- `%PROJECT_NAME%`: Name of the project containing the object.
- `%FOLDER_PATH%`: Folder path to the object in the project tree, separated with underscores.
- `%FOLDER_NAME (n) %`: Name of one folder in the path, starting from the bottom ($n=1$ corresponds to the object's parent) to the top folder in the project tree. If the folder does not exist for the given index, an empty string is returned.
- `%FOLDER_NAME%`: Shortcut to `%FOLDER_NAME (1) %`.
- `%OBJECT_NAME%`: Name of the source object of the scenario.

3.8.22 Preferences Dialog - ODI - User Interface Page

Use to set exit and credentials options.

Bypass 'Exit ODI' prompt on exit

Specify whether to bypass the Exit ODI prompt on exit. The default value is Ask, which means users are prompted on exit.

Save Login Credentials

Select to save login credentials to the `snps_login_work_12.xml` file. If this option is disabled, you must provide credentials every time you log in.

Save Login Credentials into Wallet

Select to save login credentials using Oracle Wallet Manager instead of the `snps_login_work_12.xml` file.

Skip Export Key warning

If selected, the **Export Key warning** will not be shown.

3.8.23 Preferences Dialog - ODI - User Interface - Designer Page

Use to set Designer Navigator options.

Display 'Attribute Matching' dialog for mappings

Select to automatically map source columns to target columns when new datastores are added to a mapping by detecting column name matches.

Bypass 'Create Lookup or Join' dialog in Dataset

Select to bypass the dialog that displays when a Lookup or Join is created in a dataset.

Create a Default Dataset on creation of a new Mapping

Select to create a default dataset when a new mapping is created.

Display lock icons in the tree view

Select to display lock icons in the Designer Navigator tree view for locked objects. Disabling this option can improve speed when displaying the tree view.

Display markers and memo icons in the tree view

Select to display marker and memo icons in the Designer Navigator tree view. Disabling this option can improve speed when displaying the tree view.

Display 'Change Notification' dialog for mappings

Select to display the Change Notification dialog that describes changes made to a mapping and the affected datastores and attributes, and prompts for resolution of issues.

Display Missing References in the tree

Select to display icons in the Designer Navigator tree view for objects that are missing references to dependent objects. Disabling this option can improve speed when displaying the tree view.

Display resource names in the tree view

Select to display the resource name of a datastore in the Models accordion. This might be useful when the resource name differs from the datastore name.

Keeps in the cache the list of models whose DBMS is not accessible

Select to keep in the cache the list of models whose DBMS is not accessible. This speeds up expanding and displaying the nodes under these models.

Maintain 11g SDK compatibility for migrated interfaces

Select to maintain compatibility for migrated interfaces. When this option is enabled, interfaces upgraded during an ODI import remain compatible with the ODI 11g SDK.

Show the CDC Modifications in the tree

Select to show the CDC modifications in the Designer Navigator tree view.

Skip object move warning

Select to skip the Object Move Warning dialog displayed when an object or folder is moved within the Designer Navigator.

Skip Shortcut name changed warning

Select to skip the Move Shortname Name Warning dialog displayed when a shortcut is renamed within the Designer Navigator.

3.8.24 Preferences Dialog - ODI - User Interface - Operator Page

Use to set Operator Navigator options.

Delay between two refresh (seconds)

Specify the number of seconds between two refreshes in Operator Navigator. This applies only when auto-refresh mode is enabled. The default value is 5.

Enable Operator display limit dialog

Select to display a warning dialog each time the Operator Navigator display limit is exceeded.

Hide On Connect and Disconnect Steps

Select to hide the On Connect and Disconnect steps.

Operator display limit (0=no limit)

Specify the operator display limit. When the number of sessions to display in Operator Navigator exceeds this number, a confirmation message is displayed. The default value is 100.

3.8.25 Preferences Dialog - ODI - User Interface - Topology Page

Use to skip the warning dialog related to work repository creation.

Skip warning when using localhost in JDBC Url for Work Repository Creation wizard

Select to skip the warning dialog that displays when a work repository is created with a localhost value.

3.8.26 Preferences Dialog - ODI - User Interface - Versioning

Use to assign colors to ODI objects and fields. When comparing the different versions of an object based on the action performed, the objects and fields appear in the assigned colors.

To change the color assigned to a particular object or field, click the arrow on the bottom left corner of the color box.

3.8.27 Preferences Dialog - ODI - User Interface - Versioning - Subversion Page

Use to specify Subversion settings.

Working Folder Path

Select the Working Folder Path by opening a file browser dialog and navigating through the file system of the ODI User to the appropriate folder. The working folder contains artifacts exported from the Subversion repository to execute various version management operations like export, commit, restore, and so on. ODI replicates the same folder structure present in the Subversion repository while storing these artifacts in the working folder.

Merge Working Folder

Open a file browser dialog and navigate through the file system to the appropriate folder. This folder contains the ODI artifacts exported from the Subversion repository during the Merge operation.

3.8.28 Preferences Dialog - Web Browser and Proxy - Web Browsers

Use to specify the default browser used by Oracle Data Integrator when you run, for example, JSP and servlet pages.

Web Browsers

By default, lists the browsers that JDeveloper has detected on your machine. You can:

- Set one of the browsers as the default to be used by selecting it under **Default**.
- Add a new web browser by clicking **Add** and entering information in the appropriate fields.
- Delete a browser from the list by selecting it and clicking **Remove**.

Name

Enter or edit the name of your browser.

Application

Enter the full path of the browser program file to run your web application.

Application Command Line Parameters

Enter command line parameters to run when the browser opens. `${URL}` is the default value, which is the URL of the page to be loaded by the web browser. Parameters you enter here are appended to your application URL defined in Application. For example, if you want to use Firefox as your browser and open to a local host page in a new tab, you might specify `-remote "openURL(http://localhost8080/mypage.jsf new-tab)"`, in which case the following line is executed: `C:\Program Files (x86)\Mozilla Firefox\firefox.exe -new-window http://localhost:8080/somepage.jsf`.

Icon

Choose the icon that will indicate your default browser. This icon appears in the Browser Preview toolbar button on your application.

3.8.29 Preferences Dialog - Web Browser and Proxy - Proxy Page

Use to specify the default browser used by Oracle Data Integrator when you run, for example, JSP and servlet pages.

No Proxy

Select this option when your system does not use a proxy to access the internet.

Use System Default Proxy Settings

Select this option to have the server use the default proxy settings on your machine. These are taken from the operation system (in the case of Windows and MacOS) or from the window manager (in the case of Linux).

- On Windows, go to Control Panel, Internet Options.
- On MacOS you go to System Preferences, Network, Advanced, Proxies.
- On Linux, the proxy settings are configured in the window manager (for example, on Gnome, this is in System, Preferences, Network Proxy).

Use Automatic Configuration Script

Specify the location of an automatic configuration script. For example, the URL for a corporate wpad.dat file.

Manual Proxy Settings

Use to manually define the proxy settings specific for your organization.

Host

Enter the URL or DNS name of your organization proxy server.

Port

Enter the port number your machine uses to access the proxy server.

No proxy for

By default, Oracle Data Integrator picks up the proxy exceptions the first time it is started. In this field you can amend the list to add exceptions so no proxy is used when you are accessing internal sites. This field is only populated when Manual Proxy Settings is selected. Use "|" to separate entries and "*" as a wild card.

Proxy Server Requires Authentication

Select to enable the username and password fields for accessing proxies that require authentication.

User Name

Enter the username for the proxy server.

Password

Enter the password for the proxy server.

Test proxy

Click to test these proxy settings and confirm that they work as expected.

3.8.30 Preferences Dialog - Web Browser and Proxy - Internet Files Page

Use to specify how to manage cookies.

Enable Internet Cookies

Select this option to enable internet cookies. The default is selected.

Clear All cookies

Click to clear all cookies.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

3.9 Property Inspector

Use to display the properties of the object selected in a mapping or flow. For example, if the target datastore is selected in a mapping, the target datastore's properties are displayed in the Property Inspector.

Name	Description
Auto Extend	Click to toggle on and off the automatic expansion of the Property Inspector to display the full contents of the active tabbed page when the cursor is over the Property Inspector. When the cursor moves to another part of the user interface, the Inspector returns to the default position.
Freeze View	Click to continue displaying the current contents of the Property Inspector even after you select a different object.
New View	Click to open a new Property Inspector window.
Edit	If available, click to open an editor for the currently selected component.
Find Area	Click to search for properties by name. Enter one or more characters to match the first part of the property or event name and click the Next and Previous arrows to search through the list.
Help	Click to view the Javadoc or tag reference topic for the currently selected component.

The Property Inspector displays groups of properties relevant to the selected component within tabbed pages.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

3.10 Save Files

Use to select the files you want to save before closing Oracle Data Integrator Studio.

From the list of ODI objects that have been modified, select the ODI objects you want to save.

You can click **Select All** to select all items from the list or **Deselect All** to deselect the selected items in the list.

Click **OK** to save the selected items and to disconnect from the repository.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

3.11 Save to File

Use to select a file (or files) appropriate to the current context. (This dialog is sometimes used during save operations).

Location

Location of the file that you want to save.

File Name

Name of the file.

File Type

The file format/type that you want to save the file in. Click **Save** to create the file.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

3.12 Thumbnail

The thumbnail window offers a global view of the selected mapping or flow. By moving the selection area in the thumbnail window, you can move the visible objects in the mapping or flow diagram.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

3.13 Unable to Connect Dialog

Use to update your proxy settings if you are unable to connect to the internet to check for updates.

For the description of the fields in this dialog, see "[Preferences Dialog - Web Browser and Proxy - Proxy Page](#)".

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

3.14 Unable to Save Dialog

Use to view the errors due to which the current ODI component could not be saved.

The cause of the error is displayed in the dialog box.

Click **OK** to close this dialog box and to fix the error.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

3.15 Update Center Dialog

Use to create or edit an Update Center.

Properties	Description
Name	Enter the name that describes the Update Center.
Location	Enter a valid web address where the web-based Update Center is located, or enter or browse to a file-based location for a local Update Center.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

3.16 Windows Dialog

Use to activate or to close windows in your working area within Oracle Data Integrator Studio. This is particularly useful when your work area is crowded with many windows.

Windows

Displays an alphabetical list of the currently open windows. To move quickly through the list, begin typing the window name to be selected. At the first unique identifying letters, the cursor jumps to that name. To activate a window, select it and click **Activate**. To activate a window and dismiss the dialog box, press **Enter**.

Activate

Click to activate the selected window. Activation is immediate.

Close

Click to close the selected window or windows. Closure is immediate.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

Navigator-Independent Editors, Wizards, and Dialogs

Oracle Data Integrator Studio provides several navigator-independent editors, wizards, and dialogs to facilitate the tasks of creating different types of objects and connecting to Oracle Data Integrator, the Master Repository and the Work Repositories.

This chapter includes the following topics:

- Add ODI Objects to Subversion Dialog
- Add Open Tool Dialog
- Change Current ODI User Password Dialog
- Change Wallet Password Dialog
- Clean Stale Sessions Dialog
- Clean Stale Sessions Results Dialog
- Compare with Export File Dialog
- Configure Subversion Repository with ODI Dialog
- Connect to Debug Session... Dialog
- Contents Of Deployment Archive Dialog
- Create Branch From Tag Dialog
- Create Default SVN Project Structure Dialog
- Create Deployment Archive with All Repository Objects Dialog
- Create Deployment Archive from Selected Objects Dialog
- Create Deployment Archive from VCS Label Dialog
- Create Full Tag Dialog
- Create New Model and Topology Objects Wizard
- Big Data Configurations Wizard
- Create Partial Tag Dialog
- Select Tag Dialog
- Create Version Dialog
- Datasources Deployment Dialog
- Debugger

-
- Deployment Archive Details Dialog
 - Display Options Dialog
 - Drivers Dialog
 - Enter Export Key Dialog for Multiple Files Import
 - Enter Export Key Dialog for Single File Import
 - Enter Export Key Dialog to Compare with Export File
 - Enter Wallet Password Dialog
 - Export Key fields in Export Dialog
 - Export of <Object> Dialog
 - Export of <Version> Dialog
 - Export Multiple Objects Dialog
 - Export Selection Dialog
 - Groovy Editor
 - Import Report
 - Import Scenario Dialog
 - Import Selection Dialog
 - Initialize Deployment Repository Dialog
 - Inventory Dialog
 - Master Repository Creation Wizard
 - Master Repository Import Wizard
 - Merge Dialog
 - Merge Results Window
 - Merge Summary Dialog
 - Merge Summary Selection Dialog
 - New Password Dialog
 - New Wallet Password Dialog
 - OPSS Enterprise Roles Dialog
 - Oracle Data Integrator Login Dialog
 - Patch Deployment Repository Dialog
 - Preferences Dialog
 - Recover Password Storage Wizard
 - Release Tag Selection Dialog
 - Release Tag Wizard
 - Repository Connection Information Editor
 - Restart Debug Session Dialog
 - Restart Session Dialog
 - Restore Deleted ODI Object

- Restore Object from Subversion Repository Dialog
- Rollback the Patch Dialog
- Search Messages Log Dialog
- Select Version Control Management System Dialog
- Simulation Report
- Smart Export Dialog
- Smart Export Report
- Smart Import Report
- Smart Import Wizard
- Start Debug Session... Dialog
- Stop Session Dialog
- Structure Pane
- SVN Connection Dialog
- Switch Authentication Mode Wizard
- Password Storage Wizard
- Switch Repository Compatibility Mode Wizard
- Target Object Selection Dialog
- URL Examples Dialog
- Version Browser Dialog
- Version Compare Results Window
- Version History Dialog
- Version Selection Dialog
- Version Tree Editor
- Version Tree Filter Dialog

4.1 Add ODI Objects to Subversion Dialog

Use to add all the unversioned ODI objects to a Subversion repository.

Properties	Description
Select	Option to select or unselect unversioned ODI objects. Note: If you do not want to add all the unversioned ODI objects to the Subversion repository, click the Unselect All icon on the toolbar and then select specific unversioned objects.
Name	Name of the unversioned ODI object to be added to the Subversion repository. This is a read-only column.
Type	Type of ODI object to be added to the Subversion repository. This is a read-only column.
Path	Path of the unversioned ODI object in the ODI repository. This is a read-only column.
Comments	Log message, which describes the changes you are committing.

Related Topics



"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.2 Add Open Tool Dialog

Use to add or remove an Open tool.

Note: Before adding an Open tool, add your Open tool JAR or ZIP files to the `<ide.user.dir>/oracledi/userlib` folder and restart Oracle Data Integrator. The `<ide.user.dir>` property displays when you select the Properties tab of the About Oracle Data Integrator Studio dialog box (**Help** main menu > **About Oracle Data Integrator**).

This operation must be performed each time an Open tool is added to ODI Studio and for each agent that runs sessions using this tool.

Properties	Description
Open Tool Class Name	<p>Enter the name of the class.</p> <p> Click to add the Open Tool you have entered in the Open Tool Class Name field.</p>
Class	<p> Click to search for the Open Tool's Java class. The Select a Class dialog box displays. You can browse for the name of the Open tool's Java class or search by entering a part of the name in the field in top of the dialog box.</p> <p>In the Class column, select the line containing your Open Tool.</p> <ul style="list-style-type: none"> ▪ If the Open tool is found on the classpath, then the supplied icons, the Open tool's syntax, description, provider, and version number are displayed. ▪ If the Open tool is not found, and error message is displayed. You should change the classpath, or move the Open tool to the right directory. <p>Note: This operation must be performed only once for a given master repository.</p> <p>An Open Tool name cannot start with <code>Snp</code> or <code>Odi</code>. An Open Tool with a name starting with these strings is ignored.</p>

Click **Delete** to remove the selected Open tool.

Click **OK** to add the Open Tool and close this dialog.

Related Topics

"Adding Oracle Data Integrator Tool Steps" in the "Creating and Using Packages" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.3 Change Current ODI User Password Dialog

Use to change the password of the current ODI user.

Properties	Description
Current Password	Enter the current ODI user password.
New Password	Enter the new ODI user password. The new password must adhere to the defined guidelines in the Password Policies Security Settings Dialog .
Confirm New Password	Enter the new ODI user password.

Click **OK** to apply the changes and close the dialog box.

Related Topics

"Enforcing Password Policies" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

4.4 Change Wallet Password Dialog

Use to change the wallet password.

Open this dialog from the menu command **ODI>Change Wallet Password...**

To change the decision of using a wallet password go to the preferences dialog at menu command **Tools>Preferences...** in the preference **ODI>User Interface>Save Login Credentials into Wallet**.

Properties	Description
Current Password	Current wallet password.
Wallet Password	New wallet password. Note: New password must have at least 8 characters with alphanumeric characters combined with numbers or special characters.
Confirm Wallet Password	Confirmation of wallet password. Note: Must match new password.
Expires in (days)	Number of days before wallet password is invalidated.

Related Topics

"Changing the Wallet Password" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

4.5 Clean Stale Sessions Dialog

Use to select agents and work repositories and manually perform a clean stale session operation.

- From the list, select the agents that must execute/clean sessions.
Select **Clean all Agents** if all agents must execute/clean sessions.
- From the list, select the work repositories you want to clean.
Select **Clean all Work Repositories** if you want to clean stale sessions in all work repositories.

Click **OK** to start the cleaning process.

Related Topics

"Cleaning Stale Sessions" in the "Managing your Executions" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

4.6 Clean Stale Sessions Results Dialog

Use to view the results of the clean stale sessions operation.

The Clean Stale Sessions Results Dialog displays the stale sessions that have been detected and cleaned.

The following table lists the stale session details:

Properties	Description
Agent	Agent that has executed the session.
Work Repository	Work repository that contains the stale session.
Session	Session ID of the stale session.
Exceptions	Exception that displays if the session could not be cleaned.

Click **Close** to close the dialog box.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

4.7 Compare with Export File Dialog

Use to compare the repository object with the selected export file.

Properties	Description
Directory	Directory in which the file is created.
Filename	Select a file to compare.

If the export file contains cipher data, you must provide an export key. For information, see [Enter Export Key Dialog to Compare with Export File](#).

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

4.8 Configure Subversion Repository with ODI Dialog

Use this to configure the ODI repository with the Subversion Version Control System.

Properties	Description
SVN Repository Name	Name of the existing Subversion repository.
SVN Project Name	List of the Subversion projects available in the Subversion repository.
Create Default Project Structure	Click to create a default project structure in the Subversion repository.

Properties	Description
Select Trunk or Branch	Select a trunk or an available branch.
New Branch	Click to create a new branch from a tag.
Auto Version	Select to enable auto versioning at the ODI repository level. Note: If Auto Version is enabled, a new version is created for an already versioned object whenever changes are made to the object and the changes are saved.
VCS Key	Select to create a VCS Key. The VCS key can be a minimum of 8 characters and a maximum of 100 characters long.
Enter VCS Key	Click to enter the VCS Key.

Note: After configuring the Subversion repository with ODI, you must disconnect from the repository and connect again.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.9 Connect to Debug Session... Dialog

Use to connect the debugger to a debuggable session on an agent registered with the current repository.

The session that the debugger connects to has to be a debuggable session and started prior to opening this dialog box. It is not possible to connect to sessions running locally without an agent.

A session can be executed with a debuggable flag either through the [Start Debug Session... Dialog](#) with the **Associate to current client** flag deselected, or by starting a session through console, command line, or web services with debug enabled.

Properties	Description
Agent	Select the agent that the debuggable session runs on. The Any Agent choice shows all debuggable sessions from all configured agents in the session list.
Session	Shows all debuggable sessions available for connection from the agent chosen in the agent list.

Related Topics

"Connecting to a Running Debugging Session" in the "Debugging Integration Processes" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

4.10 Contents Of Deployment Archive Dialog

Use to view the details of the objects added, updated, or deleted by the Deployment Archive.

Properties	Description
Object Name	Name of the object.

Properties	Description
Object Type	Type of object.
Path	Location where the object is stored.
Global ID	Deployment archive Global ID.

Related Topics

"Release Management" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.11 Create Branch From Tag Dialog

Use to create a Subversion branch from a Subversion tag.

Properties	Description
Branch Name	Name of the branch.
Tag	List of tags for the ETL project configured in the ODI repository.
Comments	Log message, which describes the changes you are committing.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.12 Create Default SVN Project Structure Dialog

Use to create a default project structure in the existing Subversion repository. Using this structure you can differentiate trunk, branch, and tag folders present in the Subversion project created in the Subversion repository.

Properties	Description
SVN Repository Name	Name of the Subversion repository.
SVN Project Name	Name of the Subversion project.
Comments	Log message, which describes the changes you are committing.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.13 Create Deployment Archive with All Repository Objects Dialog

Use to create an Initial Deployment Archive from a full repository. The deployment archive is created with all the repository objects and this might take some time, depending on the number of objects in the repository.

Note: An Execution Deployment Archive is automatically created when you create an initial deployment archive. The Execution Deployment Archive is named EXEC_<DeploymentArchiveName>.zip.

Properties	Description
Deployment Archive Name	Enter the name you want to assign to the deployment archive.
Deployment Archive Folder Path	Enter or browse for the location where the deployment archive will be stored.
Description	Enter a description of the deployment archive.
Export Key	Enter the Export Key used for cipher text migration. An Export Key is used to cipher sensitive information such as credentials while exporting ODI objects to the deployment archive. The Export Key string is minimum 8 characters long and maximum 100 characters long. It should have at least one special character (@#%\$+/=) or digit, one alphabetic lower or upper case character.
Confirm Export Key	Enter the Export Key again.
Save Export Key	Check to save the Export Key for all future exports.

Related Topics

"Release Management" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.14 Create Deployment Archive from Selected Objects Dialog

Use to create an Initial Deployment Archive or a Patch Deployment Archive using selected objects from a repository.

Note: An Execution Deployment Archive is automatically created when you create an initial or patch deployment archive. The Execution Deployment Archive is named EXEC_<DeploymentArchiveName>.zip.

Properties	Description
Initial Deployment Archive	Select to create an initial deployment archive.
Patch Deployment Archive	Select to create a patch deployment archive.
Deployment Archive Name	Enter the name you want to assign to the deployment archive.
Deployment Archive Folder Path	Enter or browse for the location where the deployment archive will be stored.
Description	Enter a description of the deployment archive.
Include all security objects	Check to include all the security data in the deployment archive.
Export Key	Enter the Export Key used for cipher text migration. An Export Key is used to cipher sensitive information such as credentials while exporting ODI objects to the deployment archive. The Export Key string is minimum 8 characters long and maximum 100 characters long. It should have at least one special character (@#%\$+/=) or digit, one alphabetic lower or upper case character.
Confirm Export Key	Enter the Export Key again.

Properties	Description
Save Export Key	Check to save the Export Key for all future exports.
Objects to be added to Deployment Archive	<ul style="list-style-type: none"> ■ Selected Objects: Drag-and-drop objects or right-click the object in the tree view and click Add to Deployment Archive. ■ Dependent Objects: View the dependent objects of the selected object.

Related Topics

"Release Management" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.15 Create Deployment Archive from VCS Label Dialog

Use to create an Initial Deployment Archive or a Patch Deployment Archive from the selected VCS label.

Note: An Execution Deployment Archive is automatically created when you create an initial or patch deployment archive. The Execution Deployment Archive is named EXEC_<DeploymentArchiveName>.zip.

Properties	Description
Initial Deployment Archive	Select to create an initial deployment archive.
Patch Deployment Archive	Select to create a patch deployment archive.
Deployment Archive Name	Enter the name you want to assign to the deployment archive.
Deployment Archive Folder Path	Enter or browse for the location where the deployment archive will be stored.
Select VCS Label	Select a label from the VCS label list.
Description	Enter a description of the deployment archive.
Export Key	Enter the Export Key used for cipher text migration. An Export Key is used to cipher sensitive information such as credentials while exporting ODI objects to the deployment archive. The Export Key string is minimum 8 characters long and maximum 100 characters long. It should have at least one special character (@#\$\$%+/=) or digit, one alphabetic lower or upper case character.
Confirm Export Key	Enter the Export Key again.
Save Export Key	Check to save the Export Key for all future exports.

Related Topics

"Release Management" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.16 Create Full Tag Dialog

Use to create a Subversion tag from all the ODI objects present in the trunk or branch of the SVN repository. A tag is identification text that you can assign to identify a set of consistent object versions, or the entire repository in Subversion. When a full tag is created, all repository objects are checked in to SVN and the tag is created for all the objects.

Properties	Description
Tag	Name of the tag. Click the Select Tag icon to select an existing tag. Note: The existing full tag must be empty or an error is displayed. If no existing tags are available, you can enter a name for a new tag.
Branch/Trunk	Name of the Subversion branch or trunk to which ODI is configured. This is a read-only field.
Comments	Log message, which describes the changes you are committing.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.17 Create New Model and Topology Objects Wizard

Use to create a new model and other Topology objects.

The required connection details to create a new model are:

- **Login Name:** Name of the login defined to the Master or Work Repository.
- **User:** Oracle Data Integrator user name.
- **Password:** Oracle Data Integrator user password.

These parameters have been specified in the [Repository Connection Information Editor](#).

The Create New Model and Topology Objects wizard has the following steps:

- [Step 1 - Create a new model and associate it with a new or existing logical schema and a new or existing context](#)
- [Step 2 - Associate the new model with a new or existing data server](#)
- [Step 3 - Associate the new model with a new or existing physical schema](#)

4.17.1 Step 1 - Create a new model and associate it with a new or existing logical schema and a new or existing context

Properties	Description
Name	Name of the model used in the user interface.
Technology	Technology of the data model. As a data model is linked to a single technology, it is a physical data model whose datatypes are native to this technology.
Logical Schema	Logical schema of the topology to which this model is attached.

Properties	Description
Context	Context in which the new model is executed.

4.17.2 Step 2 - Associate the new model with a new or existing data server

Data Server

Properties	Description
Name	Name of the data server, as it appears in the user interface. Note: For naming data servers, it is recommended to use the following naming standard: <TECHNOLOGY_NAME>_<SERVER_NAME>.
Technology	Only Appears if the Technology selected for the Model is of type Generic SQL. Technology linked to the data server.
User	User name used for connecting to the data server.
Password	Password linked with the user name. Note: This password is stored encrypted in the repository.
Driver List	Provides a list of available drivers to be used for the data server.
Driver	Name of the driver used for connecting to the data server.
URL	Provides the connection details.

Properties

Properties	Description
Key	Key identifying this property. This key is case-sensitive.
Value	Value for the property.

Note: For data servers that use ODI Special Drivers, the Key is pre-populated and only the Value must be entered. For information on ODI Special Drivers, see the [Data Server Editor](#) section.

Click **Test Connection** to verify that the connection is working.

4.17.3 Step 3 - Associate the new model with a new or existing physical schema

Properties	Description
Name	Name of the physical schema, as it appears in the graphic interface. It is calculated automatically and is read-only.
Datasource (Catalog)	Only appears if the Technology selected supports Catalogs. Name of the catalog in the data server.

Properties	Description
Schema (Schema)	<p>Only appears if the Technology selected supports Schemas.</p> <p>Name of the schema in the data server. Schema, owner, or library where the required data is stored.</p> <p>Caution: Oracle Data Integrator lists all the schemas present in the data server. Sometimes, Oracle Data Integrator cannot draw up this list. In this case, you should enter the schema name, respecting the case.</p>
Datasource (Work Catalog)	<p>Only appears if the Technology selected supports Catalogs.</p> <p>Indicate the catalog in which you want to create these objects. For some data validation or transformation operations, Oracle Data Integrator may require work objects to be created.</p>
Schema (Work Schema)	<p>Only appears if the Technology selected supports Schemas.</p> <p>Indicate the schema in which you want to create these objects. For some data validation or transformation operations, Oracle Data Integrator may require work objects to be created.</p> <p>It is recommended that you create a specific schema dedicated to any work tables. By creating a schema named SAS or ODI in all your data servers, you ensure that all Oracle Data Integrator activity remains totally independent from your applications.</p> <p>Caution: Oracle Data Integrator lists all the schemas present in the data server. Sometimes, Oracle Data Integrator cannot draw up this list. In this case, you should enter the schema name, respecting the case.</p> <p>Note: If you do not set a Work Schema, it defaults to the Schema during the execution.</p>

For more information, see [Context Editor](#), [Data Server Editor](#), [Physical Schema Editor](#).

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

4.18 Big Data Configurations Wizard

Use to configure Big Data technologies, such as Hadoop, HBase, Oozie, Spark, Hive, Pig, etc.

The Big Data Configurations wizard has the following steps:

- [Step 1 - Specify the general settings](#)
- [Step 2 - Configure the data servers for selected Big Data technologies](#)
- [Step 3 - Validate all the settings](#)

4.18.1 Step 1 - Specify the general settings

Properties	Description
Prefix	Prefix to be attached to the data server name, logical schema name, and physical schema name.
Distribution	Type of distribution, either Manual or CDH <version> .

Properties	Description
Base Directory	Only Appears if the Distribution type is other than Manual. Base directory to be used for installation. The base directory that you specify is automatically populated in other panels of the wizard.
Technologies	List of Big Data technologies that can be configured. Select the technologies that you want to configure.

4.18.2 Step 2 - Configure the data servers for selected Big Data technologies

Use to configure data servers for different Big Data technologies and create required infrastructure, physical schema and logical schema, in the ODI Topology.

A data server creation panel for each of the technologies selected in the **General Settings** panel appears. The data server panel lets you configure the properties for creating the data servers, specify a physical and a logical schema, and set a context.

For more information about the data server properties for different technologies, refer to the technologies documentation and *Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator*.

4.18.3 Step 3 - Validate all the settings

Use to validate all the settings that you configured in the data server panels.

Click **Test All Settings** to validate the settings against the data servers.

Related Topics

Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator

4.19 Create Partial Tag Dialog

Use to create a partial tag from ODI objects. You can create a partial tag from the subset of ODI artifacts present in the trunk or branch of the Subversion repository. This will enable you to create a consistent set of ODI artifacts in the Subversion Repository, which can be shared with other users who can create a new repository from a partial tag.

Properties	Description
Tag	Name of the tag. Click the Select Tag icon to select an existing tag.
Branch/Trunk	Name of the Subversion branch or trunk to which ODI is configured. This is a read-only field.
Include all security objects	Select to include all security objects.
Comments	Log message, which describes the changes you are committing.
Selected Objects	Objects to be added. To add objects, drag-and-drop the supported objects from the navigator tree or search for an object using the Search option on the toolbar and add it.
Dependent Objects	Objects dependent on the selected object.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.20 Select Tag Dialog

Displays only empty tags available in the configured Subversion repository.

In the Partial Tag dialog, you can:

- Either type a unique tag name in the Tag text box.
- Or search from a list of empty tag names.

Clicking the Search icon next to the Tag text box opens the Select Tag dialog.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.21 Create Version Dialog

Use to create a version of a version controlled ODI object.

Properties	Description
Name	Name of the ODI object. This is a read-only field.
Type	Type of ODI object. This is a read-only field.
Path	Path of the ODI object present in the ODI repository. This is a read-only field.
Comments	Log message, which describes the changes you are committing.

When creating a version of a version controlled parent object, information regarding the child is also displayed in the dialog. This information includes the Name, Type, Path, and if the child object was added, modified, or deleted.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.22 Datasources Deployment Dialog

Use to deploy datasources in an application server.

You can deploy datasources from the Topology Navigator into an application server (Oracle WebLogic Server) for which a Java EE agent is configured.

Deployment Server: The application server on which you want to deploy the data sources. The datasources can only be deployed on the following JAVA EE server:

- Oracle WebLogic (WLS)

Deployment Server Details

- **Host:** Host name or IP address of the application server.

- **Port:** Port of the application server.
- **User:** Server user name.
- **Password:** Server user password.

Datasource Deployment

Target: The server instance on which the data source will be deployed. For example: odi_server1

Click **OK** to deploy the datasources.

Note: This operation only creates the Datasources definition in WebLogic Server. It does not install the drivers or library files needed for these datasources to work. Additional drivers added to the Studio classpath can be included in the Agent Template.

Related Topics

"Create a Server Template for the Java EE Agent" in the "Setting Up a Topology" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*




4.23 Debugger











Use to execute a session in Oracle Data Integrator debugger by manually performing the steps and tasks of the session.

You can view the values of variables as well as uncommitted data through [Debug Variables](#) and [Debug Data](#) views, only if the editor is displaying a debuggable session to which that instance of studio is connected. You can set breakpoints that are listed in the [Debug Breakpoint](#) view.

Active debug sessions are displayed in the [Debug Session](#) view, while threads within these sessions are listed in the [Debug Session Threads](#) view.

Start the Oracle Data Integrator debugger for a mapping, procedure, package, or scenario. These objects have a Debug... command in the context menu or toolbar. When the debug session is started, the [Start Debug Session... Dialog](#) is first opened. Then the session editor in its Blueprint view opens, if **Associate with this client** option is selected. In the Blueprint view, all steps and tasks of the given scenario are displayed and this allows you to go through individual commands. The debug execution can be controlled by the debug commands in the debug toolbar:

Icon	Command	Description
	Add Breakpoint	Toggles a breakpoint at the currently selected line in the blueprint.
	Start Debug Session...	Starts a debugging session for the editor in focus. In a session editor, this command can be used to restart a session.
	Connect to Debug Session...	Connects to a debugging session running on an agent. This opens the currently running session and allow to step through the execution. It is not possible to connect to a local agent.

Icon	Command	Description
	Disconnect Debug Session	Disconnects the current debugging session. After the debugging session is disconnected, the execution continues and any breakpoint is ignored. It is possible to reconnect to a disconnected session. It is not possible to disconnect from a session running on a local agent.
	Current Cursor	Highlights the current cursor in the blueprint, and opens the session editor, if not already open.
	Get Data	Inserts the SQL code of the currently selected task into the SQL command field of the Debug Data window. Both Source Task Data and Target Task Data windows are populated.
	Step into	Steps to the beginning of the first child node of the currently selected node. If the currently selected node does not have a child, this is disabled.
	Run to Task End	Runs to the end of the currently selected task. If the task has children, execute all children until the end of the task is reached. This is disabled if it is not at the beginning of a task.
	Run to Next Task	Runs to the beginning of the next task. If at the last task in a list of tasks, go to the end of the parent task.
	Run to Step End	Runs to the end of the current step. Executes all tasks until the end of the step is reached. This is disabled if it is not at the beginning of a step.
	Run to Next Step	Runs to the beginning of the next step.
	Pause	Suspends the current execution. The execution can be continued by stepping through the steps/tasks, or by resuming execution.
	Resume	Resumes execution at the current cursor and continues until the session is finished or a breakpoint is reached.

4.23.1 Debug Variables

Use to view all the variables used in the current session.





Variable values can be modified to change the execution of the session.

Properties	Description
Name	Name of the variable.
Value	Value of the variable. Can be modified to change execution.
Type	Type of the variable.

4.23.2 Debug Breakpoint

Use to view the list of all breakpoints.

The breakpoint execution can be controlled by the breakpoint commands in the main menu:

Icon	Command	Description
	Refresh	Updates the window with the latest breakpoint information.
	Edit Breakpoint	Edits a breakpoint.
	Enable	Enables multiple breakpoints.
	Disable	Disables multiple breakpoints.

This list can be used to edit or remove breakpoints throughout the repository. Two lists of breakpoints are shown:

- Runtime Breakpoints: Defined in a blueprint. Runtime breakpoints can be added in the blueprint tab of a session editor.

Properties	Description
Breakpoint	ID of the breakpoint.
Scenario	Name of scenario on which breakpoint is based on, empty if not started from scenario.
Blueprint	ID of the blueprint.
Step	Name of Step on which the breakpoint is located. This is empty if the breakpoint is located on Task.
Task	Name of Task on which the breakpoint is located. This is empty if the breakpoint is located on Step.
Suspend after executing the task	<ul style="list-style-type: none"> ■ No: The breakpoint suspends before executing the task or step. ■ Yes: The breakpoint suspends after executing the task or step. <p>This property can be changed through the Edit Breakpoint context menu command.</p>
Pass count	Number of times the breakpoint has to be passed before suspending execution; 0 suspends the execution on the initial pass, 1 passes the breakpoint once before suspending execution.
Status	<ul style="list-style-type: none"> ■ Enabled: The breakpoint is active and suspends execution. ■ Disabled: The breakpoint is ignored during execution.

- Design Breakpoints: Defined in a design artifact such as package or procedure. When executing, a design breakpoint generates a runtime breakpoint in the blueprint it is associated with. Design breakpoints can be added in [Package Editor](#) or [Procedure Editor](#).

Properties	Description
Breakpoint	ID of breakpoint.
Package	Package where the breakpoint is defined.
Procedure	Procedure where this breakpoint is defined.
Step	Step on which the breakpoint is located. This is empty if the breakpoint is located on Task.
Task	Task on which the breakpoint is located. This is empty if the breakpoint is located on Step.
Suspend after executing the task	<ul style="list-style-type: none"> ■ No: The breakpoint suspends before executing the task or step. ■ Yes: The breakpoint suspends after executing the task or step. <p>This property can be changed through the Edit Breakpoint context menu command.</p>
Pass count	The execution is suspended on the configured value; 1 suspends only on the first pass, 2 suspends only on the second pass, 0 suspends the execution on all passes.
Status	<ul style="list-style-type: none"> ■ Enabled: The breakpoint is active and suspends execution. ■ Disabled: The breakpoint is ignored during execution.

4.23.2.1 Breakpoint Properties

Start Breakpoint properties through Breakpoint view or Edit Breakpoint in the blueprint tab. Edit breakpoint properties on the breakpoint icon on the step/task in the Blueprint view.

Properties	Description
Enabled	<p>If checked the breakpoint is active and execution will be suspended on reaching it.</p> <p>If unchecked the breakpoint will not be active and will be ignored.</p>
Suspend after executing the task	<p>If checked the breakpoint suspends after executing the task or step.</p> <p>If unchecked the breakpoint suspends before executing the task or step.</p>
Pass Count	The execution is suspended on the configured value; 1 suspends only on the first pass, 2 suspends only on the second pass, 0 suspends the execution on all passes.

4.23.3 Debug Session

Use to see the list of all active debug sessions.

It allows you to connect or disconnect the listed sessions.

Session List

Properties	Description
Agent	Agent executing the session.

Properties	Description
Session Number	ID of the session.
Session Name	Name of the session.
Session Connected	Shows whether the session is connected or not connected to the debugger client.

4.23.4 Debug Session Threads

Use to see all threads for the connected session editor in focus.

The icon of the thread shows whether the thread is suspended or executing. The Go To Source context menu command on each thread jumps to the thread location inside the blueprint.

4.23.5 Debug Data

Allows querying of data in the context of the current debugger task.

The data window has two tabs, Source Task Data and Target Task Data, to query the data servers associated with source and target for the current task. The window provides a field to enter a SQL command and execute it by clicking the **Run SQL Code** button. The Get Data command in the main toolbar inserts the SQL code of the current task into both Source Task Data and Target Task Data fields.

All SQL commands use the transaction context of the current session, uncommitted data can be queried.

Related Topics

"About Sessions and Blueprints" in the chapter "Debugging Integration Processes" in *Oracle Fusion Middleware Administering Oracle Data Integrator*

"Stepping through a Blueprint in the Session Editor" in the chapter "Debugging Integration Processes" in *Oracle Fusion Middleware Administering Oracle Data Integrator*

4.24 Deployment Archive Details Dialog

Use to view the details of a Deployment Archive before applying it to an ODI repository. In the Deployment Archive Folder Path field, specify the deployment archive's location and a list of deployment archives is displayed. This includes the following information.

Properties	Description
Name	Name of the deployment archive.
Type	Type of deployment archive.
Source Type	Source type of the deployment archive. The source type can either be VCS or ODI, depending on whether the deployment archive was created from a VCS label or a repository.
Description	Description of the deployment archive.
Creation Date	Date when the deployment archive was created.
Created By User	User who created the deployment archive.
Global ID	Deployment archive Global ID.

Properties	Description
Source Info	Information regarding the source type.

To view information regarding the objects that are updated or deleted when the deployment archive is applied, select the Deployment Archive and click the **View** icon on the toolbar.

Related Topics

"Release Management" *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.25 Display Options Dialog

Use to view or update the options that determine which object or field is displayed in the Version Compare Results dialog.

Reset All Filters

Click to reset all filters. By default, all the check boxes are selected and this results in all the objects and fields being displayed in the Version Compare Results dialog.

To change the color assigned to a particular object or field, click the arrow on the bottom left corner of the color box.

Note: Colors assigned to the objects and fields are retrieved from Preferences. If you want to change the colors permanently, go to Preferences > ODI > User Interface > Versioning. For more information see "[Preferences Dialog - ODI - User Interface - Versioning](#)".

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.26 Drivers Dialog

Use to display the selected driver's details.

Name

Name of the driver used for connecting to the data server.

Syntax

Class name of the driver.

Description

Description of the driver.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

4.27 Enter Export Key Dialog for Multiple Files Import

Use to import multiple files.

Properties	Description
File Name	Name of the file.
Enter Export Key	Export Key used when this file was created during the export.
Import file without cipher data	Select to import a file without an Export Key. The cipher data of this file will not be imported.
Import All files without cipher data	Select to import all files without Export Keys. All files will be imported without cipher data.
Skip file	Select to skip this file for import.

Related Topics

"Export Keys" in the "Exporting and Importing" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.28 Enter Export Key Dialog for Single File Import

Use to import a single file.

Properties	Description
Enter Export Key	Export Key used when this file was created during the export.
Import file without cipher data	Select to import a file without an Export Key. The cipher data of this file will not be imported.

Related Topics

"Export Keys" in the "Exporting and Importing" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.29 Enter Export Key Dialog to Compare with Export File

Use to provide an export key when comparing the repository object with an export file that contains cipher text.

Properties	Description
Enter Export Key	Select to enter the Export Key used for cipher text migration. The Export Key string is minimum 8 characters long and maximum 100 characters long. It should have at least one special character (@#\$\$%+/=) or digit, one alphabetic lower or upper case character.
Compare file without cipher data	Select to nullify the cipher data from the export file.

Related Topics

"Export Keys" in the "Exporting and Importing" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.30 Enter Wallet Password Dialog

Use to enter the wallet password.

This dialog shows up when a user connects to a repository in Oracle Data Integrator Studio and login credentials have been stored in a wallet.

The dialog also shows up when deselecting the preference **ODI>User Interface>Save Login Credentials into Wallet**.

Properties	Description
Wallet Password	Wallet password previously set to protect the wallet.

Related Topics

"Using a Password-Protected Wallet for Storing Login Credentials" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

4.31 Export Key fields in Export Dialog

Use to enter the export key for sensitive encryption data.

Properties	Description
Export Key	Specifies the AES KEY for any sensitive data encryption needed during the export. The export key string is minimum 8 characters long and maximum 100 characters long. It should have at least one special character (@#\$\$%+/=) or digit, one alphabetic lower or upper case character.
Confirm Export Key	Enter your export key again.
Save Export Key	If checked, your Export Key is saved for all future exports.

Related Topics

"Export Keys" in the "Exporting and Importing" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.32 Export of <Object> Dialog

Use to export one single object.

In the Export <Object> Dialog, set the export parameters as follows:

Properties	Description
Export Directory	Directory in which the export file is created.
Export Name	Name given to the export (mandatory).
Export child objects	If this option is checked, the objects linked to the object to be exported are also exported. These objects are those visible under the exported object in the Navigator tree. It is recommended to leave this option checked.
Replace existing files without warning	If this option is checked, the existing file is replaced directly by the ones of the export without any warning (no confirmation dialog is displayed). If a file with the same name as the export file already exists, it will be overwritten by the export file.

Encryption

For more information about Encryption see, [Export Key fields in Export Dialog](#)

Advanced Options

Use these options to customize the XML output file format. It is recommended that you use the default values.

Properties	Description
XML Version	The xml version parameter in the XML file header. <code><?xml version="1.0" encoding="ISO-8859-1"?></code>
XML Character Set	The encoding parameter in the XML file header. <code><?xml version="1.0" encoding="ISO-8859-1"?></code>
Java Character Set	Java character set used to generate the file.

Related Topics

"Exporting an Object with its Child Components" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.33 Export of <Version> Dialog

Use to export a version of an object.

The version is exported to a file without restoring it. The exported version can be imported to another repository.

Note: When you export a version of an object, only the object contained in the version is exported and not version information. This allows you too export an old version without having to restore it in the repository.

In the Export of <Version> Dialog, set the export parameters as follows:

Properties	Description
Export Directory	Directory in which the export file is created. The default is the directory that was specified for the last export.
Export Name	Name given to the export (mandatory).
Replace existing files without warning	If this option is checked, existing files are replaced with the export files without warning, and a confirmation dialog box is not displayed. If a file with the same name as the export file already exists, it will be overwritten by the export file.

Encryption

For more information about Encryption see, [Export Key fields in Export Dialog](#)

Related Topics

"Exporting a Version with the Version Browser" in the "Using Version Control" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.34 Export Multiple Objects Dialog

Use to export one or more ODI objects at the same time to a zip file or a directory.

The objects are either exported as .xml files directly into the directory, or as a zip file containing .xml files.

In the Export Multiple Objects Dialog, set the export parameters as follows:

Properties	Description
Export to directory	Directory in which the export files are created.
Export to zip file	Select to generate a zip file.
Zip File Name	Name of the zip file.
Export child objects	If this option is checked, the objects linked to the object to be exported are also exported. Child objects are displayed under the exported object in the Navigation tree. It is recommended to leave this option checked. See "Exporting an Object with its Child Components" in the <i>Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator</i> for more details.
Replace existing files without warning	If this option is checked, existing files are replaced directly by the export files without any warning, and a confirmation dialog box is not displayed. If a file with the same name as the export file already exists, it is overwritten by the export file.

Encryption

For more information about Encryption see, [Export Key fields in Export Dialog](#)

Advanced Options

Use these options to customize the XML output file format. It is recommended that you use the default values.

Properties	Description
XML Version	XML version specified in the export file. The XML version parameter in the XML file header. <code><?xml version="1.0" encoding="ISO-8859-1" ?></code>
XML Character Set	Encoding specified in the export file. The encoding parameter in the XML file header. <code><?xml version="1.0" encoding="ISO-8859-1" ?></code>
Java Character Set	Java character set used to generate the file.

Export List

Use the export list to specify the list of objects to export:

1. Drag and drop the objects from the Oracle Data Integrator Navigators into the Export list. Note that you can export objects from different Navigators at the same time.
2. Click **Load a list of objects...** to load a previously saved list of objects. This is useful if you regularly export the same list of objects.
3. To save the current list of objects, click **Save Export List** and specify a file name. If the file already exists, it will be overwritten without any warning.

A more powerful mechanism for exporting and importing multiple objects are Solutions.

Related Topics

"Exporting an Object with its Child Components" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

"Working with Solutions" in the "Using Version Control" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.35 Export Selection Dialog

Use to select an export operation.

The following export operations can be launched from this dialog box:

Export Operation	Description
Smart Export	Exports a list of objects with their dependencies. Note: This export option is only available if you are connected to a work repository.
Export Multiple Objects	Exports a list of objects.
Export the Master Repository	Exports the entire content of the master repository. This export is used to create a clone of the master repository or to transfer the content of one master repository to another.
Export the Work Repository	Exports the entire content of the work repository. This export is used to transfer the content of one work repository to another. Note: This export option is only available if you are connected to a Work repository.
Export the Topology	Exports the data servers, physical/logical agents, and schemas as well as the contexts from the master repository.
Export the Logical Topology	Exports the logical architecture (logical schema and agents) from the master repository.
Export the Execution Environment	Exports the physical architecture (data servers, physical schemas, and physical agents) from the master repository.
Export the Log	Archives the content of the Operator log. Perform this operation to archive the content of the log before purging its contents.
Export Security Settings	Exports the users, profiles, and privileges from the master repository.
Export the Technical Environment	Exports a description of your work environment. This export may be required for troubleshooting issues.

From the action list, select the export operation you want to perform and click **OK**.

Related Topics

"Exporting an Object with its Child Components" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.36 Groovy Editor

Use to create, edit, and execute Groovy scripts within the ODI Studio context.

The Groovy Editor contains one tab, the **Source** tab. You can enter or modify the Groovy script on this tab and execute a Groovy script when it is opened in the Groovy Editor.

Each Groovy script execution launches its own [Log Window](#) and you can follow the script execution in the [Log Window](#). The [Log Window](#) is named according to the following format: *Running <script_name>*.

The Groovy Editor provides all standard features of a code editor such as syntax highlighting and common code editor commands except for debugging. The following commands are supported and accessed through the context menu and the main Source tab:

- Show Whitespace
- Text Edits
 - Join Line
 - Delete Current Line
 - Trim Trailing Whitespace
 - Convert Leading Tabs to Spaces
 - Convert Leading Spaces to Tabs
- Indent Block
- Unindent Block
- Move Up
- Move Down

Related Topics

"Introduction to the Groovy Editor" in "Appendix C Using Groovy Scripting with Oracle Data Integrator" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.37 Import Report

Use to view the report of your import operation.

Displays the import mode. Possible values are: Duplication, Synonym Mode INSERT, Synonym Mode UPDATE, and Synonym Mode INSERT_UPDATE.

Imported Objects

Properties	Description
Object Type	Type of imported object.
Object Name	Name of imported object.
Imported As	Specifies whether the imported object has been imported as a copy.
Original ID	ID of the object before the import.
New ID After Import	New ID of the object after the import.
Global ID	Global ID of the object. The global ID is unique for the object type across repositories.
Parent Type After Import	Type of the parent of the imported object after the import.

Properties	Description
Parent Name After Import	Name of the parent of the imported object after the import.

Deleted Objects

Properties	Description
Object Type	Type of deleted object.
Object Name	Name of deleted object.
Original ID	ID of the object before the import and before deleting it.
Global ID	Global ID of the object. The global ID is unique for the object type across repositories.

Not Imported Objects

Properties	Description
Object Type	Type of object that could not be imported.
Object Name	Name of object that could not be imported.
Original ID	ID of the object that could not be imported.
Global ID	Global ID of the object. The global ID is unique for the object type across repositories.

Created Missing References

Properties	Description
Description	Description of the missing reference.
Object Type	Type of object that generates a missing reference.
Object Name	Name of object that generates a missing reference.
Original ID	ID of the object that generates a missing reference.
Global ID	Global ID of the object. The global ID is unique for the object type across repositories.

Fixed Missing References

Properties	Description
Description	Description of the missing reference that could be fixed during the import process.
Object Type	Type of object that generated a missing reference.
Object Name	Name of object that generated a missing reference.
Original ID	ID of the object that generated a missing reference.
Global ID	Global ID of the object. The global ID is unique for the object type across repositories.

Imported Objects With Cipher Text Nullified (due not specifying correct export key)

Properties	Description
Object Name	Name of the imported object.
Object Type	Type of the imported object.
Global ID	ID of the imported object.

Related Topics

"Import Modes" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.38 Import Scenario Dialog

Use to import a scenario.

In the Import Scenario Dialog, set the import parameters as follows:

Properties	Description
Import Type	Select the Import Type. Possible values are: Duplication, Synonym Mode INSERT, Synonym Mode UPDATE, and Synonym Mode INSERT_UPDATE.
File import directory	Enter the directory from which you want to import the object.
Import Schedules	Select to import the schedules exported with the scenarios.
Select the file(s) to import	From the list, select the files you want to import.

Related Topics

"Import Modes" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.39 Import Selection Dialog

Use to select an import operation.

The following import operations can be launched from this dialog:

Import Operations	Description
Smart Import	Imports selectively objects exported during a previous Smart Export.
Import the Master Repository	Imports the entire content of a master repository.
Import the Work Repository	Imports the entire content of a work repository.
Import the Topology	Import the data servers, physical/logical agents and, schemas as well as the contexts from the master repository.
Import the Logical Topology	Imports the logical architecture (logical schema and agents) from the master repository.
Import the Execution Environment	Imports the physical architecture (data servers, physical schemas, and physical agents) from the master repository.

Import Operations	Description
Import the Log	Imports an Operator log archive.
Import Security Settings	Imports the users, profiles, and privileges from the master repository.

From the action list, select the import operation you want to perform and click **OK**.

Related Topics

"Exporting an Object with its Child Components" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.40 Initialize Deployment Repository Dialog

Use to initialize an ODI repository with the contents of the selected Initial Deployment Archive. This can only be done on an empty work repository. For more information, see the "Initializing an ODI Repository Using an Initial Deployment Archive" section in the *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*.

Properties	Description
Deployment Archive Folder Path	Enter or browse for the deployment archive's location. All the deployment archives in the specified folder path are listed under List of Deployment Archives.
Name	Name of the deployment archive.
Type	Type of deployment archive.
Source Type	Source type of the deployment archive. The source type can either be VCS or ODI, depending on whether the deployment archive was created from a VCS label or a repository.
Description	Description of the deployment archive.
Creation Date	Date when the deployment archive was created.
Created By User	User who created the deployment archive.

Note: If the deployment archive contains cipher data, you will be prompted to enter the Export Key entered when first creating the deployment archive.

Related Topics

"Release Management" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.41 Inventory Dialog

Use to view a list of the Deployment Archives applied in an ODI repository.

Properties	Description
Name	Name of the deployment archive.

Properties	Description
Type	Type of deployment archive.
Source Type	Source type of the deployment archive. The source type can either be VCS or ODI, depending on whether the deployment archive was created from a VCS label or a repository.
Applied Successfully	Denotes if the deployment archive was applied successfully.
Rollback Deployment Archive Created	Denotes if the rollback deployment archive was created.
Global ID	Deployment archive Global ID.
Source Info	Information regarding the source type.
Creation Date	Date when the deployment archive was created.
Created By User	User who created the deployment archive.
Applied Date	Date when the deployment archive was applied.
Applied By User	User who applied the deployment archive.
Number of objects added	Number of objects that were added by the deployment archive.
Number of objects updated	Number of objects that were updated by the deployment archive.
Number of objects deleted	Number of objects that were deleted by the deployment archive.
File Path	Location where the deployment archive is stored.
Rollback Deployment Archive File Path	File path of the rollback deployment archive, if created.
Result Message	Result message indicating whether the deployment archive was successfully applied or not. In case of a failure, an error message is displayed in the column.
Rollback Deployment Archive Global ID	Global ID of the rollback deployment archive.

Related Topics

"Release Management" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.42 Master Repository Creation Wizard

Use to create the Oracle Data Integrator Master Repository.

How?

The Master Repository Creation Wizard has the following steps:

- [Step 1 - Repository Connection](#)
- [Step 2 - Authentication](#)
- [Step 3 - Password Storage](#)

4.42.1 Step 1 - Repository Connection

Use to enter the database connection and the repository configuration details.

Database Connection

Properties	Description
Technology	Technology type of the master repository database.
JDBC Driver	Driver used to access the technology that hosts the repository.
JDBC URL	The complete path for the data server that hosts the repository.
Registry Database Name	Existing database name used to create registry information. This field only appears when you select MySQL.
User	The user ID/login of the owner of the tables. Note: The user name should be in uppercase to ensure compliance with the Upgrade Assistant and avoid issues when upgrading the repository.
Password	The user's password.
DBA User	The database administrator's username.
DBA Password	The user's password.

Click **Test Connection** to test the connection to your Master Repository.

Click **Next** to move to the next step of the Master Repository Creation Wizard.

4.42.2 Step 2 - Authentication

Use to select the authentication mode that the Master Repository uses.

Do one of the following:

- Select **Use ODI Authentication** to manage users using ODI's internal security system and enter the following supervisor login information:

Properties	Description
Supervisor User	User name of the ODI supervisor.
Supervisor Password	Password of the supervisor. ODI is installed with a built-in password policy that prevents passwords shorter than 6 characters to be used.
Confirm Password	Password of the supervisor.

- Select **Use External Authentication** to use an external enterprise identity store, such as Oracle Internet Directory, to manage user authentication and enter the following supervisor login information:

Properties	Description
Add principals to Role	Allows the selection of Oracle Platform Security Services (OPSS) principals who have supervisor privileges in ODI master repository. The OPSS principals can be enterprise roles, enterprise users, or both.

Click **Next** to move to the next step of the Master Repository Creation Wizard.

4.42.3 Step 3 - Password Storage

Use to select the password storage mode. Do one of the following:

- Select **Internal Password Storage** to store passwords in the Oracle Data Integrator repository.
- Select **External Password Storage** to use JPS Credential Store Framework (CSF) to store the data server and context passwords in a remote credential store. Indicate the **MBean Server Parameters** to access the credential store.

Enter the following MBean Server Parameters:

Properties	Description
Host	MBeans Server Host, for example: <i>mymachine.oracle.com</i>
Port	MBeans Server Port, for example: <i>7001</i>
User	MBeans Server User Name, for example: <i>weblogic</i>
Password	MBeans Server Password, for example: <i>weblogic</i>

Click **Test Connection** to test the connection to your MBean Server.

Click **Finish** to complete the Master Repository Creation Wizard and create the Master Repository.

Related Topics

"Setting Up External Password Storage" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

4.43 Master Repository Import Wizard

Use to create a new Master Repository by exporting another Master Repository.

How?

The Master Repository Import Wizard has the following steps:

- [Step 1 - Repository Connection](#)
- [Step 2 - Password Storage](#)

4.43.1 Step 1 - Repository Connection

Use to enter the database connection and the repository configuration details.

Database Connection

Properties	Description
Login	User ID/login of the owner of the tables you have created for the Master Repository.
JDBC Driver	Driver used to access the technology, which hosts the repository.
JDBC URL	Complete path for the data server to host the repository. Note: The parameters JDBC Driver and URL are synchronized and the default values are technology dependent.
User	User ID/login of the owner of the tables.
Password	Password of the user.
DBA User	Username of the database administrator.

Properties	Description
DBA Password	Password of the database administrator.

Click **Test Connection** to test the connection to your Master Repository.

Repository Configuration

Properties	Description
ID	A specific ID for the new Master Repository, rather than the default 0. This affects imports and exports between repositories. Note: All master repositories should have distinct identifiers. Check that the identifier you are choosing is not the identifier of an existing repository.
Use a Zip File	If using a compressed export file, select Use a Zip File .
Import Path (File)	If using zip file, enter or select the file containing your Master Repository export. If not using a zip file, enter or select the folder containing the export data of your Master Repository.
Technology	From the list, select the technology your repository must be based on.

Click **Next** to move to the next step of the Master Repository Import Wizard.

4.43.2 Step 2 - Password Storage

Use to select the password storage mode. Do one of the following:

- Select **Use Password Storage Configuration specified in Export** to reuse the same password storage mode as the exported Master Repository.
- Select **Use New Password Storage Configuration** to configure a new password storage mode and do one of the following:
 - Select **Internal Password Storage** to store passwords in the Oracle Data Integrator repository.
 - Select **External Password Storage** to use JPS Credential Store Framework (CSF) to store the data server and context passwords in a remote credential store. Indicate the **MBean Server Parameters** to access the credential store.

Enter the following MBean Server Parameters:

Properties	Description
Host	MBeans Server Host, for example: <i>mymachine.oracle.com</i>
Port	MBeans Server Port, for example: <i>7001</i>
User	MBeans Server User Name, for example: <i>weblogic</i>
Password	MBeans Server Password, for example: <i>weblogic</i>

Click **Test Connection** to test the connection to your MBean Server.

Click **Finish** to complete the Master Repository Import Wizard and create a new Master Repository by exporting another Master Repository.

Related Topics

"Setting Up External Password Storage" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

4.44 Merge Dialog

Use to merge a VCS branch, tag, or trunk with an ODI repository, which is mapped to a VCS branch.

Note: Admin privileges are required to perform a merge and the client must be connected to the VCS system. For information on the privileges required, see "Managing Security in Oracle Data Integrator" in *Oracle Fusion Middleware Administering Oracle Data Integrator*.

Properties	Description
Merge Type	Select Branch, Tag, or Trunk depending on what the ODI repository must be merged with.
Source	Select the name of the Branch, Tag, or Trunk.
Merge Name	Assign a name to the merge.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.45 Merge Results Window

Use to view merge results and resolve merge conflicts.

The Merge Results window has the following tabs:

- [Merge Object Selection](#)
- [Merge Conflict Resolution](#)

To select a branch merge and view the associated merge summary, click **Merge Summary Selection** icon on the top left corner of the window.

4.45.1 Merge Object Selection

Use to select the object to be used to resolve merge conflicts.

Merge Filters

Use to filter results by the name of the merge.

Properties	Description
Merge Status	Select the status of the merge.
Merge Name	Select the name assigned to the merge.

Merge Object Filters

Use to filter results by merge object.

Properties	Description
Merge Object Status	Select the status of the merge object.
Merge Object Owner	Select the user who has conflict objects to be fixed.
Object Type	Select the object type.

Merge Objects

Use to view merge information that is displayed based on the selected filter parameters.

Properties	Description
Merge Status	Displays the icon that denotes the type of object. If there is a merge conflict, a merge conflict indicator (red cross) is displayed on the icon.
Merge Object	Displays the merge object.
Type	Displays the merge object type.
Merge Object Owner	Displays the name of the merge object owner.
Mark Object Merged	Check to resolve conflicts in the selected merge object. When this check box is checked, the merge conflict indicator is removed from the corresponding icon in the Merge Status column.

Toolbar

Name	Description
Select All	Selects all the items in the Mark Object Merged column.
Unselect All	Unselects all the items in the Mark Object Merged column.
Fix Merge Conflicts	Displays the Merge Conflict Resolution tab for the selected merge object.
Save	Saves the selected merge object.
Save All	Saves all the merge objects.

4.45.2 Merge Conflict Resolution

Use to resolve merge conflicts.

Toolbar

Name	Description
Refresh	Refreshes the results.
Go to First Difference	Jumps to the first difference.
Go to Previous Difference	Jumps to the previous difference.
Go to Next Difference	Jumps to the next difference.
Go to Last Difference	Jumps to the last difference.

Name	Description
Coloring and Filtering	Displays the "Display Options Dialog", where colors can be assigned to fields and objects.
Reset all Filters	Resets all filters to the default.
Expand All	Expands all the nodes in the tree.
Collapse All	Collapses all the nodes in the tree.
Generate Report	Generates the report.
Overwrite the repository object with the source version	Overwrites the repository object with the source version and marks the object conflict as resolved.
Use source version as the target repository object	Uses the source version as the target repository object.
Edit repository object	Edits repository objects.
Mark object conflicts as resolved	Marks object conflicts as resolved.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.46 Merge Summary Dialog

Use to view and save the summary of a merge action.

Merge

Properties	Description
Name	Name assigned to the merge.
Start Time	Time when the merge process was initiated.
Source Type	Type of source.
Source Name	Name of the source.
Target Type	Type of target.
Target Name	Name of the target.
Merge Status	Status of the merge process.

Merge Object

Properties	Description
Name	Name of the merged object.
GUID	Unique identifier of the merged object.
Conflict Status	Conflict status, namely, Added, Deleted, In Conflict, or Merged.
Merge Status	Status of the merge.
Type	Type of object.

Properties	Description
Merge Owner	Role of the person who initiated the merge.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.47 Merge Summary Selection Dialog

Use to select a branch merge and view the associated summary.

Properties	Description
Merge Status	Select the status of the merge.
Merge Name	Select the merge from the drop-down list.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.48 New Password Dialog

Use to create a new password for the SUPERVISOR user when you import a file for a master repository, which was exported without an Export Key.

Properties	Description
Supervisor User	User name of the ODI SUPERVISOR. This name is created by ODI and it is not editable.
Supervisor Password	New password of the SUPERVISOR. ODI is installed with a built-in password policy that prevents passwords shorter than 6 characters to be used.
Confirm Password	Confirms new password of the SUPERVISOR.

Related Topics

"Enforcing Password Policies" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

4.49 New Wallet Password Dialog

Use to set a wallet password.

If a wallet password is set, passwords for repository database users and Oracle Data Integrator logins are encrypted and stored in an Oracle secured wallet.

Use the wallet password as the encryption key. This is the recommended approach for a secure environment.

If passwords are stored without a secure wallet, a fixed encryption mechanism is used. This approach is recommended for environments without sensitive data.

Change wallet passwords using the menu command **ODI>Change Wallet Password...**

To change the decision of using a wallet password use the preferences dialog at menu command **Tools>Preferences...** in the preference **ODI>User Interface>Save Login Credentials into Wallet**.

To decide whether to store passwords at all use the preference **ODI>User Interface>Save Login Credentials**.

Properties	Description
Store passwords in secure wallet	Select to store passwords for the master repository database user and Oracle Data Integrator user in an Oracle secure wallet.
Wallet Password	New wallet password. Note: Passwords must have at least 8 characters with alphanumeric characters combined with numbers or special characters.
Confirm Wallet Password	Confirmation of wallet password. Note: Must match new password.
Expires in (days)	Number of days before wallet password is invalidated.
Stores passwords without secure wallet	Select to store passwords for the master repository database user and Oracle Data Integrator user using a fixed encryption mechanism.

Recover Wallet Password

Use to reset your wallet password.

If you forget your Oracle Data Integrator wallet password, remove the wallet directory file on disk. You will lose all the stored passwords.

When you open Oracle Data Integrator Studio again, you will be asked for a new wallet password.

Related Topics

"Using a Password-Protected Wallet for Storing Login Credentials" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

4.50 OPSS Enterprise Roles Dialog

Use to retrieve the Enterprise Roles and Users.

Properties	Description
Retrieval Filter	Enter an optional string to show users or roles from LDAP server. For example, if you want to fetch all the users and roles, type an *. If you want to display all the supervisor roles, type SUPERVISOR . Note: If you specify a string without an *, you get OPSS users or roles whose names match your filter string exactly.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

4.51 Oracle Data Integrator Login Dialog

Use to connect to Oracle Data Integrator.

The required connection details are:

- **Login Name:** Name of the login defined to the Master or Work Repository.
- **User:** Oracle Data Integrator user name.
- **Password:** Oracle Data Integrator user password.

These parameters have been specified in the [Repository Connection Information Editor](#).

Click **New** to create a new Oracle Data Integrator login and **Edit** to modify the existing Oracle Data Integrator login with the [Repository Connection Information Editor](#). Click **Copy** to copy a current login definition.

Click **Delete** to delete the existing Oracle Data Integrator login.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

4.52 Patch Deployment Repository Dialog

Use to apply a Patch Deployment Archive.

Properties	Description
Deployment Archive Folder Path	Enter or browse for the deployment archive's location.
Name	Name of the deployment archive.
Type	Type of deployment archive.
Source Type	Source type of the deployment archive. The source type can either be VCS or ODI, depending on whether the deployment archive was created from a VCS label or a repository.
Description	Description of the deployment archive.
Creation Date	Date when the deployment archive was created.
Created By User	User who created the deployment archive.
Create Rollback Deployment Archive	Check to create a rollback deployment archive.
Rollback Deployment Archive Path	Enter or browse for the file path of the rollback deployment archive.

Note: If the deployment archive contains cipher data, you will be prompted to enter the Export Key entered during the creating of the deployment archive.

Related Topics

"Release Management" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.53 Preferences Dialog

Use to configure the Oracle Data Integrator user parameters.

Open from **Tools** in the menu bar.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

4.54 Recover Password Storage Wizard

Use to start the password recovery process.

Oracle Data Integrator offers a password recovery service that should be used only in case of an external password storage crash. Using this procedure, password storage is forced to Internal Password Storage as the external storage is no longer available. This operation should be performed by a Supervisor user.

Caution: When performing a password storage recovery, context and data server passwords are lost and need to be reentered manually in the Topology Navigator.

Step 1 - Login

Use to specify the JDBC connectivity details of your Oracle Data Integrator Master Repository as defined when connecting to the Master Repository.

Properties	Description
Login	User ID/login of the owner of the tables you have created for the Master Repository.
JDBC Driver	Driver used to access the technology, which hosts the repository.
JDBC URL	Complete path of the data server to host the repository. Note: The parameters JDBC Driver and URL parameters are synchronized and the default values are technology dependent.
User	User ID/login of the owner of the tables.
Password	Password of the user.

Click **Finish** to complete the Recover Password Storage Wizard.

Related Topics

"Recovering the Password Storage" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

4.55 Release Tag Selection Dialog

Use to select a release tag to add to the Smart Export.

From the release tag list, select a release tag. All objects of the selected release tag are added to the Smart Export and do not have to be added individually.

The Release Tag Selection dialog displays the list of release tags that have been already added to the Smart Export.

Click **OK** to add the objects of the selected release tag to the Smart Export.

Related Topics

"Using Release Tags" in the "Using Shortcuts" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.56 Release Tag Wizard

Use to manage release tags.

You can add release tags to folders and model folders.

The Release Tag Wizard has the following steps:

- [Step 1](#)
- [Step 2](#)

4.56.1 Step 1

Select an existing release tag or enter a new release tag name from the **Release Tag Name** list that you want to add to a given folder.

Properties	Description
Available	Displays the available folders. From the list, select the folder to which you want to add the release tag and use the arrows to move it to the selected list.
Selected	Displays the selected folders.

Click **Next** to move to the next step of the Release Tag Wizard.

Click **Finish** if you do not want to add the release tag to a model.

4.56.2 Step 2

Properties	Description
Available	Displays the available models and model folders. From the list, select the model or model folder to which you want to add the release tag and use the arrows to move it to the selected list.
Selected	Displays the selected models and model folders.

Click **Finish** to add the release tag to the selected models and project folders and to close the Release Tag Wizard.

Related Topics

"Using Release Tags" in the "Using Shortcuts" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.57 Repository Connection Information Editor

Use to view and define the connection details of the following components:

- **Oracle Data Integrator Connection:** Specified by the Oracle Data Integrator Login name, user name, and user password.
- **Database Connection (Master Repository):** Specified by the database user name, user password, driver list, driver name, and driver URL to connect to the master repository.
- **Work Repository:** Select **Master Repository Only** if this connection addresses only a master repository. If not, select a work repository to which you want to connect.

Default Connection

If this check box is selected, the displayed connection details are used by default.
Click **Test** to check a connection when creating it.

You can connect to the master repository and test this connection. How?

You can also connect to a work repository and test this connection. How?

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

4.58 Restart Debug Session Dialog

Use to restart a debug session for mapping, procedure, package, or scenario.

Properties	Description
Use the previous agent	If checked, uses agent from previous execution.
Choose another agent	If checked, chooses an agent to execute debug session.
Log Level	Specifies the level of logging information that should be retained. All session tasks with a defined log level lower than or equal to this value are kept in the Session log when the session completes. However, if the object execution ends abnormally, all tasks will be kept, regardless of this setting. Note: Log level 6 has the same behavior as log level 5, but with the addition of variable and sequence tracking. More

Debug Properties

Properties	Description
Suspend Before First Task	If checked, the debug session suspends the execution before the first task, even if no client is connected to the Debuggable session. If unchecked, the debug session proceeds until it hits a breakpoint or finishes. Checked by default.
Associate to Current Client	If checked, debugger starts a session in debug mode and open the blueprint to debug the session. If unchecked, the debugger starts a session in debug mode but not open the blueprint for debugging. The session can later be associated by this or other clients. Checked by default.
Delay Before Connection (Sec)	Seconds before debugger attempts a connection to the started session. Default is 5.
Debug Descendant Sessions	If checked, it will ask to start a new debugging session whenever a Startscen command is executed. This only applies to packages or procedures, not mappings. If unchecked, any descendant session is executed without debugger. Unchecked by default.

Properties	Description
Break on Error	<p>If checked, the debugger suspends execution at the task where an error happened. This enables the user to review data within the transaction.</p> <p>If unchecked, session execution engine reacts to any errors as usual. Debugger client does not provide any notification to the user.</p> <p>Unchecked by default.</p>

Related Topics

"Restart Execution" in the "Debugging Integration Processes" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

4.59 Restart Session Dialog

Use to select the restart mode of a session.

- To select the agent to execute the session, select one of the following options:
 - Select **Use the previous agent: <agent name>** to use the agent that was used to execute a previous session.
 - Select **Choose another agent** to select the agent that you want to use to execute the session.

Note: Select **Internal** to use the ODI Studio built-in agent.

- Select the **Log Level**.

Note: Log level 6 has the same behavior as log level 5, but with the addition of variable and sequence tracking. More

- Click **OK** to restart the indicated session and to close the dialog.

Click **Cancel** if you do not want to restart session.

Related Topics

"Restarting a Session from ODI Studio" in the "Running Integration Processes" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

4.60 Restore Deleted ODI Object

Use to restore a deleted ODI object from the Subversion repository.

Version Search Criteria

Enter Version Search Criteria to search for the deleted ODI object.

Properties	Description
Date Range	Select to enter the range of dates.

Properties	Description
Version Range	Enter the first and last version to view all the versions within this range. Note: If the Version Range fields are left empty, all versions of the object are displayed. If the lower limit of the version range is not specified, all the versions that are lesser than the specified upper limit are displayed. Similarly, if the upper limit of the version range is not specified, all the versions that are greater than the specified lower limit are displayed.
Path	Path of the ODI object in the ODI repository. Note: You can use * as a wildcard character.
Comments	Enter the log message, which describes the changes that were committed. Note: You can use * as a wildcard character.
Name	Name of the ODI object. Note: You can use * as a wildcard character.
Apply	Click to view a list of deleted objects.

Versions

After you enter Version Search Criteria, search results are filtered to display relevant versions. Select the deleted ODI object that you want to restore and click the **Restore...** icon on the toolbar.

Columns	Description
Name	Name of the deleted ODI object.
Path	Path of the ODI object.
Version	Version of the deleted ODI object.
SVN User	Subversion user who created the version.
ODI User	ODI user who created the version.
Date	Date when the version was created.
Comments	Log message, which describes the changes that were committed.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.61 Restore Object from Subversion Repository Dialog

Use to restore an object from the Subversion repository to the ODI repository.

Properties	Description
Name	Name of the ODI object. This is a read-only field.
Type	Type of ODI object to be restored from the Subversion repository. This is a read-only field.
Path	Path of the ODI object present in the ODI repository. This is a read-only field.

Properties	Description
Select Version	Click to access the Version Selection dialog and select a particular version of the ODI object present in the Subversion repository.
Restore with Merge	Select to restore an object using the Merge option.
Restore child objects with Merge	Select to restore an object along with its child objects using the Merge option.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.62 Rollback the Patch Dialog

Use to rollback the last applied Patch Deployment Archive.

Note: The Create Rollback Deployment Archive check box in the [Patch Deployment Repository Dialog](#) must be checked for you to rollback a patch deployment archive.

Properties	Description
Deployment Archive Folder Path	Enter or browse for the deployment archive's location.
Name	Name of the deployment archive.
Type	Type of deployment archive.
Source Type	Source type of the deployment archive. The source type can either be VCS or ODI, depending on whether the deployment archive was created from a VCS label or a repository.
Description	Description of the deployment archive.
Creation Date	Date when the deployment archive was created.
Created By User	User who created the deployment archive.

Note: If the deployment archive contains cipher data, you will be prompted to enter the Export Key entered during the creating of the deployment archive.

Related Topics

"Release Management" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.63 Search Messages Log Dialog

Use to search for text in the Message Log.

The first match is highlighted in blue, and the subsequent matches are highlighted in yellow. Use the **Previous** and **Next** buttons to browse through the occurrences.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

4.64 Select Version Control Management System Dialog

Use to select the Version Control System (VCS) for versioning an Oracle Data Integrator object.

From the list of VCS supported by ODI, select the VCS you want to use and click **OK**. After you select a VCS, you must disconnect and connect to the ODI repository again. Also, you can disable VCS and select the legacy version management system from this dialog. For more information, see *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.65 Simulation Report

Use to view the report of your simulated execution.

Simulating an execution generates and displays the code corresponding to the execution without running the code. Execution simulation provides this report.

Level of detail: Shows the level of details to display in the simulation report.

- **Full:** If selected, all the fields are displayed in the simulation report.
- **No generated code:** If selected, the generated code is not displayed in the simulation report.
- **Only generated code:** If selected, only the generated code along with fewer important fields is displayed in the simulation report.

Hide empty fields: If selected, the empty fields are hidden in the simulation report.

Use the search box to search for text in the Simulation Report.

Note: No session is created in the log when the execution is started in simulation mode.

Properties	Description
Session Name	Name of the session whose execution is simulated.
Scenario Name	Name of the simulated scenario.
Context Code	Context code used for the simulated execution.

Step

Properties	Description
Step Name	Name of the step whose execution is simulated.
Step No	Number of the step.
Step Type	Type of the step.

Properties	Description
Ok Next Step	Next step on success.
Ko Next Step	Next step on failure.
Ok Exit	Indicator for exiting the package if step is OK.
Ko Exit	Indicator for exiting the package if step is not OK (Error).

Task

Properties	Description
Task Type	Type of the task whose execution is simulated.
Task Name 1	Name of task 1.
Task Name 2	Name of task 2.
Task Name 3	Name of task 3.
Exe Channel	Execution channel.
Source Context Code	Forced context for execution on source. Null means that the execution context would be used.
Source Logical Schema Name	Logical schema name on the source technology.
Source Connection Name	Physical data server name for source.
Source Commit Indicator	Flag indicating whether to commit on source after execution of tasks.
Source Isolation Level	Isolation level on source technology.
Source Technology Internal Name	Internal code of source technology
Target Context Code	Forced context for execution on target. Null means that execution context would be used.
Target Logical Schema Name	Logical schema name on the target technology.
Target Connection Name	Physical data server name for target.
Target Commit Indicator	Flag to indicate whether this transaction ID is committed on target.
Target Isolation Level	Isolation level on target technology.
Target Technology Internal Name	Internal code for target technology.
Default Command (Destination)	Command text for the target.

Related Topics

"Simulating an Execution" in the "Running Integration Processes" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

4.66 Smart Export Dialog

Use to perform a Smart Export.

Note: You must be connected to a work repository to be able to use the Smart Export feature.

In the Smart Export Dialog, set the export parameters as follows:

Properties	Description
Export Name	Name given to the export (mandatory). Default is <code>SmartExport.xml</code> .
Export as zip file	Select if you want to generate a zip file. Default is <code><Export Name>.zip</code> .

Encryption

For more information about Encryption see, [Export Key fields in Export Dialog](#)

Encoding Options

Use these options to customize the XML output file format. It is recommended that you use the default values.

Properties	Description
XML Character Set	Encoding specified in the export file. The encoding parameter in the XML file header. <code><?xml version="1.0" encoding="ISO-8859-1"?></code>
Java Character Set	Java character set used to generate the file.

Objects to be Exported

Use to specify the list of objects to be exported to the Selected Objects list and to manage dependencies.

1. Drag and drop the objects from the Oracle Data Integrator Navigators into the Selected Objects list on the left.

Note: You can export objects from different Navigators at once. Repeat this step if required.

2. The object to be exported appears in a tree with its related parent and child objects.
Notes:

- A bold object name indicates that the object has been specifically added to the Smart Export. Only objects that appear in bold can be removed. Removing an object also removes its child objects and the dependent objects. Child objects of a specifically added object also appear in bold and can be removed. To remove an object from the export tree, right-click the object and select **Remove Object**. If the removed object is dependent on another object, remains in the tree but the font changed to a standard font.
- A grayed out object name indicates that the object is a dependent object that is not exported, as for example, a technology.

3. If there are any cross-referenced objects, they are displayed in the **Dependencies** list on the right side of the dialog. Parent objects will not be shown under the *Uses* node and child objects will not be shown under *User by* node.

The Smart Export toolbar provides tools for managing the objects to export and for viewing dependencies. [Table 4-1](#) details the different toolbar components.

Table 4–1 Smart Export Toolbar

Name	Description
Search	Searches for a object in the Selected Objects or Dependencies list.
Expand All	Expands all tree nodes in the Selected Objects or Dependencies list.
Collapse All	Collapses all tree nodes in the Selected Objects or Dependencies list.
Clear All	Deletes all objects from the list. Warning: This also deletes Release Tags and Materialization selections.
Add Objects by Release Tag	Adds all objects that have the same release tag as the object that already exists in the Selected Objects list.

4. Click **Export** to start the export process.

Related Topics

"Exporting an Object with its Child Components" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.67 Smart Export Report

Use to view the report of your Smart Export operation.

Displays the objects that have been exported from the Master Repository.

Note: A smart export exports an object with all its related parent and child objects that are required to support.

Level of detail: Change detail of report. Only the default level of reporting is available.

Use the search bar to search for text in the Smart Export Report.

User Added Objects

Properties	Description
Object Type	Type of object that you have added to the export.
Object Name	Name of object that you have added to the export.
Object ID	ID of the object that you have added to the export.

Dependent Objects

Properties	Description
Object Type	Type of the exported object.
Object Name	Name of the exported object.
Object ID	ID of the exported object.
Used by Object Type	Type of dependent object which is used by this object.
Used By	Dependent object is used bu this object.
Uses Object Type	Type of dependent object which uses this object type.
Uses	Dependent object uses this object.

Displays the objects that have been exported from the work repository.

Note: A smart export exports an object with all its related parent and child objects that are required to support.

User added Objects

Properties	Description
Object Type	Type of object that you have added to the export.
Object Name	Name of object that you have added to the export.
Object ID	ID of the object that you have added to the export.

Dependent Objects

Properties	Description
Object Type	Type of the exported object.
Object Name	Name of the exported object.
Object ID	ID of the exported object.
Used by Object Type	Type of dependent object which is used by this object.
Used By	Dependent object is used by this object.
Uses Object Type	Type of dependent object which uses this object type.
Uses	Dependent object uses this object.

Related Topics

"Exporting an Object with its Child Components" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.68 Smart Import Report

Use to view the report of your Smart Import operation.

Displays the objects that have been imported into both the master and work repositories.

Note: A Smart Import imports an object with all its related parent and child objects that are required to support.

Create Copy Imported Objects

Properties	Description
Object Type	Type of imported object.
Object Name	Name of imported object.
Object ID	ID of the object before the import.
Parent Object Type	Type of the parent of the imported object.
Parent Object Name	Name of the parent of the imported object.
Parent Object ID	ID of the parent of the imported object.

Properties	Description
Duplicated Object Type	Type of the duplicated imported object.
Duplicated Object Name	Name of the duplicated imported object.
Duplicated Object ID	ID of the duplicated imported object.

Merge Imported Objects

Properties	Description
Object Type	Type of imported object.
Object Name	Name of imported object.
Object ID	ID of the object before the import.
Parent Object Type	Type of the parent of the imported object.
Parent Object Name	Name of the parent of the imported object.
Parent Object ID	ID of the parent of the imported object.
Duplicated Object Type	Type of the duplicated imported object.
Duplicated Object Name	Name of the duplicated imported object.
Duplicated Object ID	ID of the duplicated imported object.

Overwrite Imported Objects

Properties	Description
Object Type	Type of imported object.
Object Name	Name of imported object.
Object ID	ID of the object before the import.
Parent Object Type	Type of the parent of the imported object.
Parent Object Name	Name of the parent of the imported object.
Parent Object ID	ID of the parent of the imported object.
Duplicated Object Type	Type of the duplicated imported object.
Duplicated Object Name	Name of the duplicated imported object.
Duplicated Object ID	ID of the duplicated imported object.

Deleted Objects

Properties	Description
Object Type	Type of the deleted object.
Object Name	Name of the deleted object.
Original ID	ID of the deleted object before the import.

Reused Objects

Properties	Description
Object Type	Type of object that could not be imported.
Object Name	Name of the object that could not be imported.
Original ID	ID of the object that could not be imported.

Ignored Objects

Properties	Description
Object Type	Type of the object that could not be imported.
Object Name	Name of the object that could not be imported.
Original ID	ID of the object that could not be imported.

Created Missing References

Properties	Description
Description	Description of the missing reference.
Object Type	Type of the object that generates a missing reference.
Object Name	Name of the object that generates a missing reference.
Original ID	ID of the object that generates a missing reference.

Deleted Missing References

Properties	Description
Description	Description of the missing reference was fixed during the import process.
Object Type	Type of the object that generated a missing reference.
Object Name	Name of the object that generated a missing reference.
Original ID	ID of the object that generated a missing reference.

Imported Objects With Cipher Text Nullified (due not specifying correct export key)

Properties	Description
Object Name	Name of the imported object.
Object Type	Type of the imported object.
Global ID	ID of the imported object.

Related Topics

"Exporting an Object with its Child Components" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.69 Smart Import Wizard

Use to selectively import objects exported using the Smart Export.

The Smart Import wizard guides you through the process of importing objects into Oracle Data Integrator.

The Smart Import wizard has the following steps:

- [Step 1 - File Selection](#)
- [Step 2 - Import Actions](#)
- [Step 3 - Summary](#)

4.69.1 Step 1 - File Selection

Use to specify the import settings.

Properties	Description
File Selection	Location of the Smart Export file to be imported.
Response File	Optionally, select a Response File to replay a previous Smart Import wizard execution by presetting all fields.

Click **Next** to move to the next step of the Smart Import Wizard.

Oracle Data Integrator (ODI) starts a matching process that verifies whether the repository contains matching objects for each of the potential objects being imported. ODI also performs an issue detection process to check for any issues that would prevent a successful import.

4.69.2 Step 2 - Import Actions

Use to verify the result of the matching process and to fix potential issues.

Object Match Details

Properties	Description
Import Object	Displays the objects to be imported and their dependencies in the navigator trees. Expand the nodes to navigate to the objects to import and to display the possible actions.
Action	Select the action to perform on the object during the import operation. Possible values are: <ul style="list-style-type: none"> ▪ Merge ▪ Overwrite ▪ Create Copy ▪ Reuse ▪ Ignore
Repository Object	Displays the repository objects that match the import object. From the list, select the required repository object.
Issues	Displays an icon indicating if issues such as broken links and name or code conflicts, have been detected during the matching process. If an issue is detected, details are displayed in the Issue Details table. Next button is disabled until all critical issues are fixed.

Issue Details

This table lists the issues that have been detected during the matching process. To fix an issue, select the action to perform in the Action column.

Note: Oracle Data Integrator provides a default working solution for every issue. However, missing references may still occur during the actual import process depending on the selected import actions.

Properties	Description
Issue	Displays a warning or error icon depending on the encountered issue.
Type	Displays the type of the encountered issue. Possible values are: <ul style="list-style-type: none"> ▪ Collision (for example, Name, Code, ID). ▪ Broken Link (for example, when referenced object cannot be found). ▪ Value Changed (for example, Project Code will be changed to a different value).
Description	Description of the encountered issue.
Action	Select the action to perform to fix the issue. Possible values are: <ul style="list-style-type: none"> ▪ Ignore (Not possible on critical issues). ▪ Change (If the collision is on an ID, the new value is always NEW_ID. If a name or code collision is detected, specify the new value in the Fix field). ▪ Do not change (For value changed issues, the value in the matching target object is kept). ▪ Fix Link (For broken links, click Search in the Fix field).
Fix	If you have selected to fix the issue in the Action column, specify the fix in this column. Click Search to select the target object in the Broken Link Target Object Selection dialog to fix the broken link.

Click **Next** to move to the next step of the Smart Import Wizard.

4.69.3 Step 3 - Summary

Use to review the import file name and eventual issues. You can also create a response file that you can reuse in another import to replay the same selections.

Properties	Description
File Selection	Location of the Smart Export file to be imported.
Issue Details	Displays the unresolved warnings. Note: Critical issues are not displayed here.
Save Response File	Select this to create a response file. You can use the response file in another import to replay this Smart Execution wizard execution, by presetting all fields.

Click **Finish** to run the Smart Import and to complete the Smart Import Wizard.

The [Smart Import Report](#) is displayed at the end of the Smart Import process.

Related Topics

"Exporting an Object with its Child Components" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.70 Start Debug Session... Dialog

Use to start a debug session for mapping, procedure, package, or scenario. Use it also to rerun a completed session.

Properties	Description
Context	Specifies the context in which the session is started. Only the contexts for which you have the appropriate rights are listed here.
Physical Mapping Design	Specifies the deployment information defined in the mapping.
Logical Agent	Specifies the agent that executes the object. The object can also be directly executed by Oracle Data Integrator Studio, by selecting Local (No Agent).
Log Level	Specifies the level of logging information to retain. All session tasks with a defined log level lower than or equal to this value are kept in the Session log when the session completes. However, if the object execution ends abnormally, all tasks are kept, regardless of this setting. Note: Log level 6 has the same behavior as log level 5, but with the addition of variable and sequence tracking. More.

Debug Properties

Properties	Description
Suspend before the first task	If checked, the debug session suspends the execution before the first task, even if no client is connected to the Debuggable session. If unchecked, the debug session proceeds until it hits a breakpoint or finishes. Checked by default.
Debug descendant sessions	If checked, it will ask to start a new debugging session whenever a Startscen command is executed. This only applies to packages or procedures, not mappings. If unchecked, any descendant session is executed without debugger. Unchecked by default.
Break On Error	If checked, the debugger suspends execution at the task where an error happened. This enables the user to review data within the transaction. If unchecked, session execution engine reacts to any errors as usual. Debugger client does not provide any notification to the user. Unchecked by default.

Properties	Description
Associate to the current client	<p>If checked, debugger starts a session in debug mode and open the blueprint to debug the session.</p> <p>If unchecked, the debugger starts a session in debug mode but not open the blueprint for debugging. The session can later be associated by this or other clients.</p> <p>Checked by default.</p>
Delay before Connection (sec)	<p>Seconds before debugger attempts a connection to the started session.</p> <p>Default is 5.</p>

Related Topics

"Starting a Debugging Session" in the "Debugging Integration Processes" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

4.71 Stop Session Dialog

Use to confirm that you want to stop a specific running session.

Click **OK** to stop the indicated session and to close the dialog.

Click **Cancel** if you do not want to stop the indicated session.

Related Topics

"Stopping a Session" in the "Running Integration Processes" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

4.72 Structure Pane

Use to view components from the mapping editor in focus in a tree view.

Container components such as the [Dataset Component](#) show enclosed components as children. All [Projector Components](#) show their [Component Attributes](#) as children.

Selection of a node in the structure pane changes the selection in the Property Inspector. This makes it possible to modify components without using the mapping diagram.

Related Topics

"Using the Structure Panel" in the "Creating and Using Mappings" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.73 SVN Connection Dialog

Use to create and edit a Subversion repository connection from ODI Studio. ODI provides the following types of authentication options when connecting to the Subversion repository:

- HTTP Basic Authentication
- Subversion Basic Authentication
- SSH Authentication
- SSL Authentication

- File Based Authentication

Based on the selected authentication option, different properties are displayed in the SVN Connection Dialog. The configuration properties that appear for more than one authentication option are described below:

Properties	Description
Authentication Type	Authentication option to connect to the Subversion repository.
SVN URL	Subversion repository URL.
User	User name for authentication.
Password	Password to connect to the Subversion repository.

Properties specific to authentication options are described in the following sections:

HTTP Basic Authentication

Properties	Description
Use Proxy Server	Select to use an HTTP or HTTPS proxy server. Note: If this check box is selected, the Host and Port properties are enabled.
Host	Proxy server to connect to the Subversion repository.
Port	Proxy server port.
Proxy Server Requires Authentication	Select to authenticate the proxy server. Note: If this check box is selected, the User and Password properties are enabled.
User	User name to connect to the proxy server.
Password	Password to connect to the proxy server.

SSH Authentication

Properties	Description
Private Key File	Select to establish an SSH connection using SSH keys. Note: If this check box is selected, the Key File and Passphrase properties are enabled.
Key File	Path of the Private Key File.
Passphrase	Passphrase for the selected Private Key File.

SSL Authentication

Properties	Description
Enable Client Authentication	Select to enable client authentication. Note: If this check box is selected, the Certificate File and Passphrase properties are enabled.
Certificate File	Path of the Certificate File.
Passphrase	Passphrase for the selected Certificate File.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.74 Switch Authentication Mode Wizard

Use to switch the user authentication mode for a Master Repository.

Switching the authentication mode of the Oracle Data Integrator Master Repository changes the way users authenticate. This operation must be performed by a Supervisor user.

WARNING: When switching from an external to internal authentication, user passwords are not copied from the identity store to the repository. The passwords are nullified. All the user accounts are marked as expired and must be reactivated by a SUPERVISOR that is created during the switch.

When switching from internal to external authentication, the users who exist in the repository and match a user in the identity store are automatically mapped. Users who do not match a user in the identity store are disabled. A Supervisor must edit the users so that their name has a match in the identity store.

Before launching the Switch Authentication Mode wizard perform the following tasks:

- Disconnect Oracle Data Integrator Studio from the repository.
- Shut down every component using the Oracle Data Integrator repository.

The Switch Authentication Mode wizard has the following steps:

- [Step 1 - Login](#)
- [Step 2 - Credentials](#)

4.74.1 Step 1 - Login

Use to specify the JDBC connectivity details of your Oracle Data Integrator Master Repository as defined when connecting to the Master Repository.

Properties	Description
Login	User ID/login of the owner of the tables you have created for the Master Repository.
JDBC Driver	Driver used to access the technology, which hosts the repository.
JDBC URL	Complete path for the data server to host the repository. Note: The JDBC Driver and URL parameters are synchronized and the default values are technology dependent.
User	User ID/login of the owner of the tables.
Password	Password of the user.

Click **Next** to move to the next step of the Switch Authentication Mode Wizard.

4.74.2 Step 2 - Credentials

Use to review the changes that are made when switching the authentication mode.

The next action varies depending on the current Authentication Mode in use:

- If currently using Internal Authentication, you are prompted to switch to external authentication.
- If currently using External Authentication, you are prompted to switch to internal authentication. You must provide and confirm a password for the SUPERVISOR user that the wizard automatically creates in the repository.

Click **Finish** to complete the Switch Authentication Mode Wizard.

Related Topics

"Switching an Existing Master Repository to External Authentication Mode" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

4.75 Password Storage Wizard

Use to switch to change the password storage options of the data server passwords.

Switching the password storage of the Oracle Data Integrator repository changes how data servers and contexts passwords are stored. This operation must be performed by a supervisor user.

Before launching the Password Storage wizard perform the following tasks:

- Disconnect Oracle Data Integrator Studio from the repository.
- Shut down every component using the Oracle Data Integrator repository.

The Password Storage wizard has the following steps:

- [Step 1 - Login](#)
- [Step 2 - Password Storage](#)

4.75.1 Step 1 - Login

Use to specify the JDBC connectivity details of your Oracle Data Integrator Master Repository as defined when connecting to the Master Repository.

Properties	Description
Login	User ID/login of the owner of the tables you have created for the Master Repository
JDBC Driver	The driver used to access the technology, which will host the repository.
JDBC Url	The complete path for the data server to host the repository. Note that the parameters JDBC Driver and URL are synchronized and the default values are technology dependant.
User	The user id/login of the owner of the tables.
Password	This user's password.

Click **Next** to move to the next step of the Password Storage wizard.

4.75.2 Step 2 - Password Storage

Use to select the password storage mode. Do one of the following:

- Select **Internal Password Storage** to store passwords in the Oracle Data Integrator Repository.
- Select **External Password Storage** to use JPS Credential Store Framework (CSF) to store the data server and context passwords in a remote credential store. Indicate the **MBean Server Parameters** to access the credential store.

Enter the following MBean Server Parameters:

Properties	Description
Host	MBeans Server Host, for example: <i>mymachine.oracle.com</i>
Port	MBeans Server Port, for example: <i>7001</i>
User	MBeans Server User Name, for example: <i>weblogic</i>
Password	MBeans Server Password, for example: <i>weblogic</i>

Click **Test Connection** to test the connection to your MBean Server.

Click **Finish** to finalize the Password Storage wizard.

Related Topics

"Import Modes" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

"Switching the Password Storage" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

4.76 Switch Repository Compatibility Mode Wizard

Use to upgrade your version to a compatibility mode.

In the 12.1.2 version you started using Global Unique Identifiers (GUIDs) instead of numeric identifiers. The compatibility mode allows you to continue using numeric identifiers.

The compatibility mode allows the customers to migrate the Knowledge Modules, procedures, and any code using `odiRef` substitution APIs to the new APIs taking GUIDs as arguments. After the migration is completed, you can switch off the repository compatibility mode, in order to take full advantage of the global identifiers. This is a one time operation, once the repository compatibility mode is switched off, there is no turning back as the menu option becomes disabled. From that point on the repository will only function in full GUID mode.

Related Topics

"About Compatibility Mode" in the "Using Compatibility Mode" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.77 Target Object Selection Dialog

Use to select the target object for a broken link.

From the tree, select the target object to fix the broken link.

Click **OK** to establish the link to the target object.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

4.78 URL Examples Dialog

Use to select a valid URL for your driver.

Name

Name of the template for your URL.

Description

Description of the URL and its parameters.

Related Topics

"Driver Configuration" in "Appendix B Oracle Data Integrator Driver for XML Reference" in *Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator*

4.79 Version Browser Dialog

Use to browse for versions stored in the repository.

Use the Type and Name drop-down lists to filter the objects for which you want to display the list of versions.

From the Version Browser dialog, you can:

- Restore a version.
- Export a version as an XML file.
- Delete an existing version.
- Compare two versions for identical objects.
- Refresh the list.

Note: The Version Browser displays the versions that existed when you opened it. Click **Refresh** to view all new versions created since then.

Related Topics

"Browsing Versions" in the "Using Version Control" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.80 Version Compare Results Window

Use to determine the differences between two versions of an object or a selected version and repository object and merge the changes, if required.

Toolbar

Name	Description
Refresh	Refreshes the results.
Go to First Difference	Jumps to the first difference.
Go to Previous Difference	Jumps to the previous difference.
Go to Next Difference	Jumps to the next difference.
Go to Last Difference	Jumps to the last difference.
Coloring and Filtering	Displays the " Display Options Dialog ", from where colors can be assigned to fields and objects.
Reset all Filters	Resets all filters to the default.
Generate Report	Generates the report.
Expand All	Expands all the nodes in the tree.
Collapse All	Collapses all the nodes in the tree.
Perform Merge	Displays the " Merge Dialog " from where you can perform merge operations.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.81 Version History Dialog

Use to view the Version History of an ODI object selected in the Subversion repository.

Properties	Description
Name	Name of the selected ODI object. This is a read-only field.
Current Version	Current version of the selected ODI object. This is a read-only field.

Version Search Criteria

Enter Version Search Criteria to search for a particular version of the selected ODI object.

Properties	Description
Branch/Tag/Trunk	Select Branch, Tag, or Trunk depending on what the ODI repository is mapped to.
Branch/Tag	Select the appropriate Branch or Tag from the list. Note: This list is populated only if the Branch or Tag check box is selected.
Date Range	Select to enter the range of dates within which the version was created.

Properties	Description
Version Range	Enter the first and last version to view all the versions within this range. Note: If the Version Range fields are left empty, all versions of the object are displayed. If the lower limit of the version range is not specified, all the versions that are lesser than the specified upper limit are displayed. Similarly, if the upper limit of the version range is not specified, all the versions that are greater than the specified lower limit are displayed.
Path	Path of the ODI object in the ODI repository. Note: You can use * as a wildcard character.
Comments	Enter the log message, which describes the changes that were committed in the selected version of the object. Note: You can use * as a wildcard character.
Apply	Click to view version history.

Versions

View the Version History of the selected ODI object. If you have entered Version Search Criteria, search results are filtered to display relevant versions. To compare selected versions or a selected version with a repository object, click the **Compare Versions** icon on the toolbar. Similarly, to restore an ODI object from an existing version to the repository, click the **Restore...** icon.

Columns	Description
Version	Version(s) of the selected ODI object.
Action	Action(s) performed on the selected ODI object.
Path	Path of the ODI object in the ODI repository.
Date	Date when the version was created.
SVN User	Subversion user who created the version.
ODI User	ODI user who created the version.
Comments	Log message, which describes the changes that were committed in each version.
Copy Source	Source location from where this version was created. Note: This is applicable to the Copy, Move, and Rename actions only.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.82 Version Selection Dialog

Use to view the versions of an ODI object and select the version that must be restored. When you select a version and click **OK**, the [Restore Object from Subversion Repository Dialog](#) is populated with the version number.

Columns	Description
Version	Version(s) of the ODI object.
Action	Action performed that resulted in the particular version.

Columns	Description
Date	Date when the version was created.
SVN User	Subversion user who created the version.
ODI User	ODI user who created the version.
Comments	Log message, which describes the changes that were committed.
Version Move Details	Information regarding the action performed in a particular version of the ODI object. If the ODI object was renamed, moved, or copied, information regarding the path and the path to the location from where it was copied is displayed.

Version Move Details

This section is populated when you select a version.

Columns	Description
Action	Action performed in the selected version.
Path	Path of the ODI object.
Copied From Path	Path to the location from where the object was copied. This information is displayed if the ODI object was renamed or moved in the selected version.

Note: The OK button on this dialog is enabled only after you select a version in the Version Move Details section.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.83 Version Tree Editor

Use to view the version history of a version controlled ODI artifact across multiple branches, tags, or trunk present in the Subversion repository. To generate the graph, ODI fetches all the log messages from the integration project root and each revision graph node in the version tree graph represents a revision in the repository. The different nodes are distinguished by different colors, for example, the items that were deleted are red in color. Right-click a version to view the corresponding log in the Version Properties section.

Toolbar

Name	Description
Refresh	Refreshes the version history.
Hide Insignificant Versions	Hides the nodes to which no changes were made.
Reset Layout	Resets the version history.
Zoom In	Zooms in to the version tree display.
Zoom Out	Zooms out of the version tree display.

Name	Description
Zoom 100%	Zooms in a 100% of the version tree display.
Fit to Window	Fits the version tree to window.
Hand Tool	Moves the version tree graph up and down or from left to right.
Version Properties	Displays version properties.
Version Tree Filter	Displays the "Version Tree Filter Dialog".
Version Compare	Enables you to compare selected versions or compare a selected version with a repository object.

Version Properties

Use to view the properties of the version and the corresponding value. For example, the Property column displays Version Creation Date and the corresponding Value column displays the date and time when the version was created.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

4.84 Version Tree Filter Dialog

Use to filter the versions from the version tree based on the selected criteria.

Properties	Description
Date Range	Select to enter the range of dates within which the version was created.
Version Range	Enter the first and last version to view all the versions within this range. Note: If the Version Range fields are left empty, all versions of the object are displayed. If the lower limit of the version range is not specified, all the versions that are lesser than the specified upper limit are displayed. Similarly, if the upper limit of the version range is not specified, all the versions that are greater than the specified lower limit are displayed.

Related Topics

"Integrating ODI with Version Control Systems" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

Operator Navigator Editors

This chapter lists the Oracle Data Integrator help topics for the Operator Navigator.

This chapter includes the following topics:

- [Define Filter Dialog](#)
- [Export the Log Dialog](#)
- [Import of the Log Dialog](#)
- [Load Plan and Scenario Folder](#)
- [Load Plan Run Editor](#)
- [Operator Navigator](#)
- [Purge Log Dialog](#)
- [Remove Temporary Objects Dialog](#)
- [Regenerate Scenario Dialog](#)
- [Restart Load Plan Dialog](#)
- [Scenario Variable Editor](#)
- [Session Editor](#)
- [Session Folder Editor](#)
- [Session Step Editor](#)
- [Session Task Editor](#)
- [Session Variable Editor](#)
- [Start Load Plan Dialog](#)
- [Stop Load Plan Dialog](#)

5.1 Define Filter Dialog

Use to define the filters for the sessions that are displayed in Operator Navigator.

In the Define Filter Dialog, set the filter parameters as follows:

Properties	Description
Session Number	Number of the session.
Session Name	Name of the session. Use % as a wildcard. For example, <i>DWH%</i> matches any session whose name begins with <i>DWH</i> .

Properties	Description
Context	Session's execution context. Use the default All Contexts value, if you do not want to filter based on a context.
Agent	Agent used to execute the session. Use the default All Agents value, if you do not want to filter based on a given agent.
User	User who launched the session. Use the default All Users value, if you do not want to filter based on a given user.
Status	Status of the session. The possible states are Done, Error, Queued, Running, Waiting, Warning, and All States. Use the default All States value, if you do not want to filter based on a given session state.
From / To	Date of execution. Specify either a date From or a date To, or both.
Duration greater than	Only sessions whose execution duration exceeds the time (in seconds) indicated in this field are displayed.

Related Topics

"Filtering Sessions" in the "Monitoring Integration Processes" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

5.2 Export the Log Dialog

Use to export log data for archiving purposes.

In the Export the Log Dialog, set the export parameters as follows:

Properties	Description
Export to directory	Directory in which the export files are created.
Export to zip file	Select to generate a zip file.
Zip File Name	Name of the zip file.

Filters

This set of options allows you to filter the log files to export, according to the specified parameters.

Properties	Description
Log Type	From the list, select whether you want to export the log of: <ul style="list-style-type: none"> ■ Load Plan runs and attached sessions: All the Load Plan runs that match the export criteria are exported, including all the sessions launched by the Load Plan runs along the child session's hierarchy. ■ Sessions: All the session logs that match the export filter criteria will be exported. All the Load Plan sessions are excluded when exporting the session logs. ■ All: All the Load Plan runs as well as the session logs matching the filter criteria are exported.
From / To	Date of execution. Specify either a date From or a date To, or both.
Agent	Agent used to execute the session. Use the default All Agents value, if you do not want to filter based on a given agent.
Context	Session's execution context. Use the default All Contexts value, if you do not want to filter based on a context.

Properties	Description
Status	Status of the session. The possible states are Done, Error, Queued, Running, Waiting, Warning, and All States. Use the default All States value, if you do not want to filter based on a given session state.
User	Name of the session. User who launched the session. Leave the default All Users value, if you do not want to filter based on a given user.
Session Name	Use % as a wildcard. For example, <i>DWH%</i> matches any session whose name begins with <i>DWH</i> .

Advanced Options

Use these options to customize the XML output file format. It is recommended that you leave the default values.

Properties	Description
XML Character Set	Encoding specified in the export file. The encoding parameter in the XML file header. <?xml version="1.0" encoding="ISO-8859-1"?>
Java Character Set	Java character set used to generate the file.

Related Topics

"Exporting and Importing Log Data" in the "Monitoring Integration Processes" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

5.3 Import of the Log Dialog

Use to import log data.

Importing log data allows you to import into your work repository log files that have been exported for archiving purposes.

In the Import of the Log Dialog, set the import parameters as follows:

Properties	Description
Import Mode	Displays the import mode. The log import supports Synonym Mode INSERT.
Import From a Folder	Select to import the log data from a folder.
Import From a Zip File	Select to import the log data from a zip file.

In the text field, enter the file import folder or zip file. You can browse for the import folder or zip file.

Related Topics

"Exporting and Importing Log Data" in the "Monitoring Integration Processes" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

5.4 Load Plan and Scenario Folder

Use to group scenarios.

Scenario folders can contain other scenario folders.

The Scenario Folder Editor has the following tabs:

- [Definition](#)
- [Version](#)
- [Privileges](#)

5.4.1 Definition

Properties	Description
Name	Name of the scenario.
Description	Description of the scenario.

Related Topics

"Using Scenarios" and "Using Load Plans" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

5.5 Load Plan Run Editor

Use to monitor Load Plan runs and to view the results of a Load Plan run.

Executing a Load Plan creates a Load Plan instance and the first Load Plan run for the instance. Every time a Load Plan instance is restarted, a Load Plan run is created for the Load Plan instance. A Load Plan run corresponds to an attempt to execute the instance.

When running, a Load Plan run starts sessions corresponding to the scenarios sequenced in the Load Plan.

The Load Plan Run Editor has the following tabs:

- [Execution](#)
- [Load Plan](#)
- [Steps](#)
- [Variables](#)
- [Privileges](#)

5.5.1 Execution

Properties	Description
Instance ID	Load Plan instance identifier. Every time a Load Plan is executed, a new Load Plan instance with a unique identifier is created.
Global ID	Global unique instance identified for the Load Plan.
Run #	Number of execution attempts of this Load Plan instance. It starts at 1 when the Load Plan is first executed, and is incremented each time the Load Plan instance is restarted.
Load Plan Name	Name of the Load Plan at the time the Load Plan instance was created.
Started By	User who initiated the execution.

Properties	Description
Physical Agent	Agent that started the Load Plan run. Executing a Load Plan may involve multiple agents for running the different steps. The agent displayed here, is the one that started the root step.
Context	Context in which the Load Plan starts.
Start	Date and time when the Load Plan run starts.
End	Date and time when the Load Plan run ends. If the Load Plan is still running, this field is blank.
Duration	Duration of the Load Plan run (format: hh:mm:ss).
Status	Load Plan status. The possible states are Done, Error, Running, and Warning.
Return Code	Return code for the Load Plan run.
Invocation Log Level	Load Plan log level specified when starting the Load Plan.
Error Message	Error message, if any.

5.5.2 Load Plan

Use to view the Load Plan settings.

Properties	Description
Name	Name of the Load Plan.
Folder Name	Name of the Load Plan and Scenario folder where this Load Plan is located. If the Load Plan is not created within a folder, this field is disabled.
Keep Log History (days)	Number of days for which log history will be kept for this Load Plan.
Log Sessions	Indicates how logs are preserved for the sessions started by the Load Plan. Possible values are: <ul style="list-style-type: none"> ■ Always: Session logs are always preserved. ■ Never: Session logs are not preserved. This has no effect on Run Scenario steps that are configured as <i>Restart from Failed Step</i> or <i>Restart From Failed Task</i>, as the whole session needs to be preserved for restartability. ■ Errors: Session log is preserved only if the session completed in an error state.
Log Session Steps	Indicates how logs are maintained by default for the session steps of each of the sessions started by the Load Plan. This applies only when the session log is preserved. Possible values are: <ul style="list-style-type: none"> ■ By Scenario Settings: Session step logs are preserved depending on the scenario settings. Scenario created from packages can preserve the steps depending on the advanced step property called Log Steps in the Journal. All steps can be preserved in other scenarios. ■ Never: Session step logs are not preserved. This has no effect on Run Scenario steps that are configured as <i>Restart from Failed Step</i> or <i>Restart From Failed Task</i>, as the whole session needs to be preserved for restartability. ■ Errors: Session step log is preserved only if the step is in an error state.

Properties	Description
Session Tasks Log Level	Default task level logging (from 1 to 5). This value corresponds to the Log Level value when starting unitary scenarios. Default is 5. When Run Scenario steps are configured as <i>Restart from Failed Step</i> or <i>Restart From Failed Task</i> , this parameter is ignored as the whole session must be preserved for restartability.
Keywords	Comma separated list of keywords that are set on the sessions started from this Load Plan. It shows the effective keywords for the step at a time of execution. These keywords improve the organization of ODI logs by session folders and automatic classification. You can override these keywords at the level of the child steps.
Description	Description of the Load Plan.

Concurrent Execution Controller

Properties	Description
Limit Concurrent Executions	If checked, enables the Concurrent Execution Controller for the Load Plan.
Violation Behavior	Specifies the violation behavior action. Default is <i>Raise Error and End Execution</i> .
Wait Polling Interval	Specifies the polling frequency (in seconds) for the wait behavior to check for its turn to run. This field accepts only a positive integer, and it is enabled only when Violation Behavior is selected as Wait . Note: A newly generated scenario or a newly created load plan has blank as the default value, meaning that the runtime agent determines the actual value used for the polling (currently the default is 30 seconds in the runtime agent).

5.5.3 Steps

Use to view the Load Plan execution details at step level.

The Steps Hierarchy table shows the organization of the steps in the Load Plan. Each row in this table represents a step.

The Steps Hierarchy table contains the following columns:

- **Steps Hierarchy:** Displays the steps hierarchy in a tree showing the name of each step.
- **Status:** Depending on the step type:
 - Displays the session status, which may be *Queued* or *Running*, for a running Run Scenario step
 - Displays the step status, which may be *Error*, *Warning*, *Done*, or *Done in previous run*, for other step types

Note: You can change the status of a failed Load Plan step from *Error* to *Done* to ignore this particular Load Plan step the next time the Load Plan run is restarted. This might be useful, for example, when the error causing this Load Plan step to fail cannot be fixed at the moment, and you want to execute the rest of the Load Plan regardless of this Load Plan step.

- **Start:** Depending on the step type:
 - Displays the start session for a Run Scenario step. The session start will be *after* the step start if the step was Queued and it will be before the step start if the step was Done in previous run.
 - Displays the start step for other step types.
- **End:** Depending on the step type:
 - Displays the session end for a Run Scenario step.
 - Displays the step end for other step types.
- **Duration:** Depending on the step type:
 - Displays the duration of the session for a Run Scenario step.
 - Displays the duration of the step for other step types.
- **Scenario/Variable:** Displays the name of the scenario for Run Scenario steps or the name of the variable evaluated in a Case step.
- **Session ID:** For Run Scenario steps only: Click the Session ID to open the session started by the Load Plan for this step.
- **Return Code:** Return code for step.
- **Error Message:** Error message, if the step encountered an error.
- **Row Count:** Overall number of rows processed.
- **Insert Count:** Number of inserts processed.
- **Update Count:** Number of updates processed.
- **Delete Count:** Number of deletes processed
- **Error Count:** Number of errors encountered.
- **Step Status:** Status of the step. The possible states are Done, Done in previous run, Error, Running, and Warning.
- **Step Start:** Start time of the execution of the step.
- **Step End:** End time of the execution of the step.
- **Step Duration:** Time taken for the execution of the step, showing the number of minutes and seconds between Step Start and Step End.
- **Step ID:** ID of the step. Useful for debugging purposes.
- **Session Status:** Status of the session. The possible states are Done, Error, Running, Waiting, Queued, and Warning.
- **Session Start:** Start time of the execution of the session.
- **Session End:** End time of the execution of the session.
- **Session Duration:** Time taken for the execution of the session, showing the number of minutes and seconds between Session Start and Session End.

In the Editor toolbar, click **Refresh** to update the content of the Steps Hierarchy table.

If you want to show or hide columns in the table, click **Select Columns...** in the Steps toolbar.

Using the Steps toolbar, you can also perform the following:

- Click **Expand All** or **Collapse All** to expand or collapse the nodes in the Steps Hierarchy table.
- Click **Show Exceptions** to view exception steps of this Load Plan run. The minimum number of tree nodes are expanded to display all exception steps that were invoked during this Load Plan run. This feature is only enabled if at least one exception step exists.

The Property Inspector displays the step properties for the step that is selected in the Steps Hierarchy table. See [Steps](#) for more information on the step properties.

Note that the properties are read-only.

5.5.4 Variables

Use to view the startup values and settings of the variables used in the Load Plan.

Properties	Description
Data Type	Data type of the variable.
Description	Description of the variable.
Logical Schema	Logical schema of the variable where the refresh query to be executed.
Select Statement	Query to be executed for refreshing variable value from the logical schema.
Startup Value	Value of the variable when the Load Plan is started.
Variable Name	Name of the variable.

Related Topics

"Introduction to the Load Plan Editor" in the "Using Load Plans" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

5.6 Operator Navigator

Use to work with your load plans, scenarios, sessions, and schedules in Oracle Data Integrator.

The Operator Navigator stores this information in a work repository.

The Operator Navigator has the following navigation trees:

- [Session List](#)
- [Hierarchical Sessions](#)
- [Scheduling](#)
- [Load Plans and Scenarios](#)
- [Solutions](#)

Session List

The Session List navigation tree displays all sessions organized per date, physical agent, sessions, status, keywords, user, and all executions.

Hierarchical Sessions

The Hierarchical Sessions navigation tree displays the execution sessions organized in a hierarchy with their child sessions.

Load Plan Executions

The Load Plan Executions navigation tree displays the Load Plan runs of the Load Plan instances.

Scheduling

The Scheduling navigation tree displays the list of physical agents and schedules.

Load Plans and Scenarios

The Scenarios navigation tree displays the list of scenarios and available Load Plans.

Solutions

The Solutions navigation tree contains the Solutions that have been created when working with version management.

Related Topics

"Running and Monitoring Integration Processes" in *Oracle Fusion Middleware Administering Oracle Data Integrator*

"Developing Integration Projects" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

5.7 Purge Log Dialog

Use to specify the purge parameters. Sessions and/or Load Plan runs are deleted based on the following parameters.

Parameter	Description
Purge Type	Select the objects to purge.
From ... To	Sessions and/or Load Plan runs during this time range will be deleted. When you choose to purge session logs only, then the sessions launched as part of the Load Plan runs are not purged even if they match the filter criteria. When you purge Load Plan runs, the Load Plan run which matched the filter criteria and the sessions launched directly as part of the Load Plan run and its child/grand sessions will be deleted.
Context	Sessions and/or Load Plan runs executed in this context are deleted.
Agent	Sessions and/or Load Plan runs executed by this agent are deleted.
Status	Session and/or Load Plan runs in this status are deleted.
User	Sessions and/or Load Plan runs executed by this user are deleted.
Name	Sessions and/or Load Plan runs matching this session name are deleted. You can specify session name masks using % as a wildcard.
Purge scenario reports	If you select Purge scenario reports , the scenario reports (appearing under the execution node of each scenario) are also purged.
Remove temporary objects on Session Deletion	If selected, removes any temporary objects left behind while purging the session.

Related Topics

"Purging the Log" in the "Monitoring Integration Processes" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

5.8 Remove Temporary Objects Dialog

Use to remove temporary objects that could remain between executions.

Properties	Description
Session count	Number of sessions for which to skip cleanup. If the count is zero, all sessions that match the filter criteria are cleaned up.
From	Start date for the cleanup. All sessions started after this date are cleaned up.
To	End date for the cleanup. All sessions started before this date are cleaned up.
Context	Cleans up only those sessions executed in this context.
Agent	Cleans up only those sessions executed by this agent.
User	Cleans up only those sessions launched by this user.
Session Name	Name of the session.

Related Topics

"Removing Temporary Objects" in the "Monitoring Integration Processes" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

5.9 Regenerate Scenario Dialog

Use to regenerate an existing scenario with the same name and version number.

The existing scenario is replaced by a scenario generated from the source object contents. Schedules attached to this scenario are preserved.

Caution: Regenerating a scenario cannot be undone. For important scenarios, it is better to generate a scenario with a new version number.

Properties	Description
Name	Name of the scenario.
Version	Version of the scenario.

Select **Generate scenario as if all underlying objects are materialized** to generate the scenario as if the shortcuts were real objects.

Note: The logical schema of the model containing the shortcut(s) is used for the scenario generation.

Click **OK** to regenerate the scenario.

Related Topics

"Using Scenarios" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

5.10 Restart Load Plan Dialog

Use to select the Physical Agent and a Log Level to restart the Load Plan.

Properties	Description
Physical Agent	From the list, select the agent that restarts the Load Plan.
Log Level	<p>Level of logging information to retain. All sessions with a defined log level lower than or equal to this value are kept in the Session log when the session completes. However, if the object execution ends abnormally, all tasks are kept, regardless of this setting.</p> <p>Note: The log level 6 has the same behavior as log level 5, but with the addition of variable and sequence tracking. More</p> <p>Select Use Previous Run Log Level (default) to use the same log level as the previous Run.</p>

Click **OK** to restart the Load Plan.

Related Topics

"Restarting a Load Plan from ODI Studio" in the "Running Integration Processes" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

5.11 Scenario Variable Editor

Use to view the scenario variable settings.

Scenario Variables are variables used in the scenario that should be set when executing the scenario to parameterize its behavior.

The Scenario Variable Editor has the following tabs:

- [Definition](#)
- [History](#)
- [Version](#)
- [Privileges](#)

5.11.1 Definition

Properties	Description
Name	Name of the scenario variable.
Datatype	Type of variable: Alphanumeric (255 characters), Date, Numeric (Maximum 10 digits), or Text (unlimited length).

Properties	Description
Keep History	Length of time the value of a variable is kept for: <ul style="list-style-type: none"> ■ No History: The value of the variable is kept in memory for a whole session. ■ Latest value: Oracle Data Integrator stores in its repository the latest value held by the variable. ■ All values: Oracle Data Integrator keeps the history of all the values held by this variable.
Secure Value	Select Secure Value if you do not want the variable to be recorded. This is useful when the variable contains passwords or other sensitive data. If Secure Value is selected: <ul style="list-style-type: none"> ■ The variable is not tracked: it is displayed unresolved in the source or target code, it will not be tracked in the repository, and it will not be historized. ■ The Keep History parameter is automatically set to No History and cannot be edited.
Default Value	Value assigned to the variable by default.
Description	Detailed description of the variable.

Related Topics

"Executing a Scenario" in the "Running Integration Processes" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

"Using Scenarios" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

5.12 Session Editor

Use to view the running or completed sessions.

A session is an execution (of a scenario, a mapping, a package, or a procedure) undertaken by an execution agent. A session is made up of steps which are made up of tasks.

The Session Editor has the following tabs:

- [Definition](#)
- [Steps](#)
- [Blueprint](#)
- [Privileges](#)
- [Details](#)

5.12.1 Definition

Properties	Description
Instance ID	Internal ID of the session within the work repository.
Global ID	Global unique identifier for the session specific across work repositories.
Session Name	Name of the session.

Properties	Description
Status	The session status when the Session Editor was opened. The possible states are Done, Error, Running, Waiting, Queued, and Warning.
Scenario Name	Name of the scenario, package, mapping, procedure, and so forth, executed in this session.
Context	Context in which the execution was run.
Scenario Version	Version (for a scenario).
Physical Agent	Agent responsible for the execution of the tasks in this session.
Start	Start date and time of execution of the session.
End	End date and time of execution of the session.
Duration (seconds)	Time taken for execution of the session.
Return code	Return code for the session.
URL	An optional URL link that enables you to view details such as execution structure, metrics, error messages, and logs using a UI specific to the execution. For example, Oozie Web Console or MapReduce JobTracker.

Concurrent Execution Controller

Properties	Description
Limit Concurrent Executions	If checked, enables the Concurrent Execution Controller for the Session.
Violation Behavior	Specifies the violation behavior action. Default is Raise Error and End Execution.
Wait Polling Interval	Specifies the polling frequency (in seconds) for the wait behavior to check for its turn to run. This field accepts only a positive integer, and it is enabled only when Violation Behavior is selected as Wait . Note: A newly generated scenario or a newly created load plan has blank as the default value, meaning that the runtime agent determines the actual value used for the polling (currently the default is 30 seconds in the runtime agent).

Record Statistics

Properties	Description
No. of Inserts	Number of rows inserted during the session.
No. of Updates	Number of rows updated during the session.
No. of Deletes	Number of rows deleted during the session.
No. of Errors	Number of rows in error in the session.
No. of Rows	Total number of rows handled during this session.

Message and Execution Details

Properties	Description
Error Message	Session execution error message, if any.

Properties	Description
Execution information details	Number of running child sessions, successful child sessions, and failed child sessions.

5.12.2 Steps

Use to view steps and executions statistics of a given session.

In the *Steps Hierarchy table*, each row represents a step. For more details on the properties displayed in this table and the actions you can perform, see the [Steps](#) section in the Load Plan Run Editor.

To bring up the Session Task Editor, double-click on a task. Select a task, to show the **Source Code** and **Target Code** fields. A Code Type field appears. Select the **Executed Code** value when you want to see the code that was executed as part of the task execution. As this is historic information of a particular task, execution cannot be edited.

Pre-Executed Code Dialog

Select the **Pre-execution Code** value to set the code that would be used by the agent/runtime at the next execution of the task. You can edit this information.

Click **Edit and use as Pre-execution Code** to open the Pre-executed Code dialog.

In the Pre-execution Code dialog, enter or edit code in the Query box. Click **Save Pre-executed Code** to save code you have entered. Click **Cancel** to reject all changes and return to the Property Inspector.

In the Property Inspector, click **Revert to original Pre-execution Code** to undo any changes or new code you have entered and restore the original Pre-execution code.

5.12.3 Blueprint

Blueprints show all the steps and task hierarchy of the selected session. You can also view and edit the source and target code for any task within the session.

You can only see a blueprint on the tab of the properties of a session. You see the same blueprint in different sessions based on a given mapping. If you make edits to a blueprint, the edited code is used for all sessions run from a mapping. Even if you delete all of the sessions, the cached blueprint survives and is reused when you next start a session from the unchanged mapping.

After you edit a mapping, new sessions that run based on it, do not use a previously cached blueprint; instead, a new blueprint is generated. If you want to preserve a blueprint, you must either create a scenario from the current mapping, or make a copy of your mapping before editing it (to preserve the blueprint for sessions run on that mapping).

In the Steps Hierarchy table, each row represents a step. You can set break points on blueprint steps and tasks for debugging purposes. See [Debugger](#) for more details.

For more details on the properties displayed in this table and the actions you can perform, see the [Steps](#) section in the Load Plan Run editor.

Select a task to show the Property Inspector with **Source Code** and **Target Code** fields.

Click **Edit and use as Pre-execution Code** to open the Pre-executed Code dialog.

In the Pre-execution Code dialog, enter or edit code in the Query box. Click **Save Pre-executed Code** to save code you have entered. Click **Cancel** to reject all changes and return to the Property Inspector.

In the Property Inspector, click **Revert to original Pre-execution Code** to undo any changes or new code you have entered and restore the original Pre-execution code.

5.12.4 Details

Use to view a summary of the Oozie or MapReduce job details when a session is executed within a Hadoop cluster.

Properties	Description
External ID	ID representing the execution in the external system, for example, an Oozie Job ID when the execution is in Oozie.
Path	Reference representing the object being executed in an external system, for example, an Oozie workflow when the execution is in Oozie.
Last Refresh	Last time the execution log details were refreshed.
Runs	Number of times this execution was performed.

Execution Details

Properties	Description
Execution Details	Displays optional text summarizing the execution with details from the external system. It might also contain embedded links to native UIs for the external system, such as the Oozie Web Console and MapReduce JobTracker.

Related Topics

Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

5.13 Session Folder Editor

Use to group sessions that were started with certain keywords.

You can use session folders to organize the log. Session folders automatically group sessions that were started with certain keywords. Session folders are created under the Keywords node on the Session List navigation tree.

Each session folder has one or more keywords associated with it. Any session launched with all the keywords of a session folder is automatically categorized beneath it.

The Session Folder Editor has the following tabs:

- [Definition](#)
- [Version](#)
- [Privileges](#)

5.13.1 Definition

Properties	Description
Folder Name	Name of the session folder.

From the Session Folder toolbar menu:

- Select **Add** to add a keyword to the list. Repeat this step for every keyword you wish to add.
- Select **Delete** to remove the selected keyword from the list.

Related Topics

"Organizing the Log with Session Folders" in the "Monitoring Integration Processes" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

5.14 Session Step Editor

Use to view the session step details.

A session step is the unit of execution found between a task and a session. It corresponds to a step in a package or in a scenario. When executing a mapping or a single variable, for example, a session has only one session step.

The Session Step Editor has the following tabs:

- [Definition](#)
- [Details](#)
- [Privileges](#)

5.14.1 Definition

Properties	Description
Step Name	Name of the step executed.
Status	Status when the Session Step Editor was opened. The possible states are Done, Error, Running, Waiting, and Warning.
Step Type	Type of step executed. More
Context	Context in which the step was executed.
Order Number	Order number linked to the step for this session.
Number of Executions	Execution number, in the event of successive executions of the step (loops).
URL	An optional URL link that enables you to view details such as execution structure, metrics, error messages, and logs using a UI specific to the execution. For example, Oozie Web Console or MapReduce JobTracker.

Record Statistics

Properties	Description
No. of Inserts	Number of rows inserted during the step.

Properties	Description
No. of Updates	Number of rows updated during the step.
No. of Deletes	Number of rows deleted during the step.
No. of Errors	Number of rows in error in the step.
No. of Rows	Total number of rows handled during this step.
Maximum number of errors allowed	<ul style="list-style-type: none"> ■ If the % check box is not selected, this is the maximum number of errors allowed (as total number of errors). ■ If the % check box is selected, this is the maximum percentage of errors allowed (based on the total number of errors). <p>When this field is left empty, then an infinite number of errors is allowed.</p> <p>When this number or percentage of errors is exceeded, the step goes into error status.</p>

Execution Statistics

Properties	Description
Start	Start date and time of execution of the step.
End	End date and time of execution of the step.
Duration (seconds)	Time taken for execution of the step.
Return code	Return code for the step.

Target Table Details

Note: The target table details are only provided for session steps in which a mapping has been executed or a datastore check has been performed.

Properties	Description
Table Name	Name of the target datastore.
Model Code	Code of the model in which the target datastore is stored.
Resource Name	Resource name of the target datastore.
Logical Schema	Logical schema of this datastore.
Forced Context Code	Context of the target datastore.

Variable and Sequence Values

Note: If tracking was not enabled for this session, this section will be empty.

Properties	Description
Name	Full name of the variable or sequence.
Type	Indicates whether this is a variable or a sequence.
Value	Value associated with the variable or sequence at the end of the session step. For sequences, the value shows the start and end value of the sequence used in this step.

Knowledge Module Details

Properties	Description
Filter By	Filters Knowledge Modules by LKM, IKM and CKM, which are used in Session Step.
Knowledge Module	Displays all Knowledge Modules used in Session Step depending on the selection in the Filter By field.
Options	Displays all options available for the selected Knowledge Module.
Name	Name of the option.
Value	Value of the option.
Use Default	Indicates whether the content in the Value field is default value or not.

Error Message

Step execution error message, if any.

5.14.2 Details

Use to view a summary of the Oozie or MapReduce job details when a session step is executed within a Hadoop cluster.

Properties	Description
External ID	ID representing the execution in the external system, for example, an Oozie Job ID when the execution is in Oozie.
Last Refresh	Last time the execution log details were refreshed.

Execution Details

Properties	Description
Execution Details	Displays optional text summarizing the execution with details from the external system. It might also contain embedded links to native UIs for the external system, such as the Oozie Web Console and MapReduce JobTracker.

Related Topics

"Monitoring Integration Processes" in *Oracle Fusion Middleware Administering Oracle Data Integrator*

Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator

5.15 Session Task Editor

Use to view the session task details.

A session task is the smallest execution unit. It corresponds for example to a procedure command in a KM, a procedure task, assignment of a variable, or a connect/disconnect command.

The Session Task Editor has the following tabs:

- [Definition](#)
- [Code](#)

- [Connection](#)
- [Details](#)
- [Privileges](#)

5.15.1 Definition

Properties	Description
Task Name	Name of the task.
Status	The status of the task when the Session Task Editor was opened. The possible states are Done, Error, Running, Waiting, and Warning. See "Monitoring Integration Processes" in <i>Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator</i> for more information.
Type	Type of task executed.
Ignore Errors	Shows if this task sets up an error tolerance, and therefore, if errors for this task are blocking.
Order	Detail the order of the task in the session (unique for the session), and the task order in the KM or the procedure that contributed to its creation.
Log Level	Log level defined for the task, representing the level of the logs that the task generates. The task log is retained if the session's execution level is the same or higher than this value. The task log level is related to the session's execution log level, but that they are different properties. Note: Log level 6 has the same behavior as log level 5, but with the addition of variable tracking. More
Invocation Log Level	Session Log Level specified in the Run Dialog or Session start.
URL	An optional URL link that enables you to view details such as execution structure, metrics, error messages, and logs using a UI specific to the execution. For example, Oozie Web Console or MapReduce JobTracker.

Record Statistics

Properties	Description
No. of Inserts	Number of rows inserted during the task.
No. of Updates	Number of rows updated during the task.
No. of Deletes	Number of rows deleted during the task.
No. of Errors	Number of rows with errors in the task.
No. of Rows	Total number of rows handled during this task.

Execution Statistics

Properties	Description
Start	Start date and time of execution of the task.
End	End date and time of execution of the task.
Duration (seconds)	Time required to execute the task.

Properties	Description
Return code	Return code for the task.

Execution Details

Properties	Description
Execution Details	Displays metrics about the performance of the loading task and any configuration data that have an impact on performance.

Variable and Sequence Values

Note: If tracking was not enabled for this session, this section will be empty.

Properties	Description
Name	Full name of the variable or sequence.
Type	Indicates whether this is a variable or a sequence.
Value	Value associated with the variable or sequence at the end of the session task. For sequences, the value shows the start and end value of the sequence used in this task.

Error Message

Task execution error message, if any.

5.15.2 Code

This tab contains the code of the commands launched on the target connection and in the source connection, if any. You can edit this code and save changes before restarting a session. Changes performed in this tab do not alter the original object which needs to be edited from its own editor.

Properties	Description
Code Type	Select Executed Code to display the code that was executed for the task. The code cannot be edited as it has been executed. Select Pre-executed Code to edit the code to be used to execute this task.

Source Code/Target Code

Properties	Description
Edit and use as Pre-execution Code	Click to open a dialog box to enter or edit Pre-execution code associated with this task. Click Save Pre-executed Code to save the code you have entered. Click Cancel to reject all changes and return to the code tab.
Revert to original Pre-execution Code	Click to delete the updates to the Pre-execution code and restore the original code.

Properties	Description
Query/Execution Plan	Click to display the execution plan used by the database to execute the SQL query of this task. Note: This feature is only provided for an Oracle database.
Show/Hide Values	Click to display the code with resolved variable and sequence values. Variables that are marked as Secure Value are not tracked and are not resolved in the code text. If the variable values are shown, the code becomes read-only.

5.15.3 Connection

This tab displays the details of source and target connections on which the code is being executed.

Properties	Description
Connection Name	Data server executing the task.
Context	Execution context of the task.
Logical Schema	Logical schema onto which the task is executed.
Transaction Isolation	Transaction isolation level for the command.
Commit	Commit management for the transaction.
Transaction ID	Number of the transaction onto which the command is executed.

5.15.4 Details

Use to view a summary of the Oozie or MapReduce job details when a session task is executed within a Hadoop cluster.

Properties	Description
External ID	ID representing the execution in the external system, for example, an Oozie Job ID when the execution is in Oozie.
Last Refresh	Last time the execution log details were refreshed.
Progress	A number between 0 and 1 that is calculated by the external execution system representing the progress of the execution.

Execution Details

Properties	Description
Execution Details	Displays optional text summarizing the execution with details from the external system. It might also contain embedded links to native UIs for the external system, such as the Oozie Web Console and MapReduce JobTracker.

Related Topics

"Monitoring Integration Processes" in *Oracle Fusion Middleware Administering Oracle Data Integrator*

Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator

5.16 Session Variable Editor

Use to view the session variable details.

A session variable displays the values taken by a variable during the execution sessions.

The Session Variable Editor has the following tabs:

- [Definition](#)
- [History](#)
- [Version](#)
- [Privileges](#)

5.16.1 Definition

Properties	Description
Name	Name of the variable. It is prefixed by GLOBAL or the code of the project containing this variable.
Datatype	Type of variable: Alphanumeric, Date, or Numeric.
Keep History	Length of time the value of a variable is kept for: <ul style="list-style-type: none"> ■ No History: The value of the variable is kept in memory for a whole session. ■ Latest value: Oracle Data Integrator stores in its repository the latest value held by the variable. ■ All values: Oracle Data Integrator keeps the history of all the values held by this variable.
Secure Value	Select Secure Value if you do not want the variable to be recorded. This is useful when the variable contains passwords or other sensitive data. If Secure Value is selected: <ul style="list-style-type: none"> ■ The variable is displayed as unresolved in the source or target code and is not tracked in the repository or historized. ■ The Keep History parameter is automatically set to No History and cannot be edited.
Default Value	Default value assigned to the variable.
Description	Detailed description of the variable.

5.16.2 History

This tab displays the history of the values of the variable.

Note that each value is attached to the context in which the session was executed:

- Click **Delete** to remove a line from the history.
- Click **Context Filter** field to filter the entries for a given context.

Related Topics

"Working with Variables" in the "Creating and Using Procedures, Variables, Sequences, and User Functions" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

5.17 Start Load Plan Dialog

Use to specify the execution parameters of the Load Plan run.

Properties	Description
Context	Specifies the context in which the Load Plan is started. Only the contexts for which you have the appropriate rights are listed here.
Logical Agent	Agent that runs the Load Plan.
Log Level	<p>Level of logging information to retain. All sessions with a defined log level lower than or equal to this value are kept in the Session log when the session completes. However, if the object execution ends abnormally, all tasks are kept, regardless of this setting.</p> <p>Note: Log level 6 has the same behavior as log level 5, but with the addition of variable and sequence tracking. More</p> <p>Select Use Session Task Log Level (default) to use the Session Tasks Log Level value defined in the Load Plan to start the Load Plan and the Scenarios in the Load Plan. If that value is not defined, a default value of 5 will be used.</p>

In the Variables table, enter the **Startup values** for the variables used in this Load Plan.

Click **OK** to start the Load Plan.

Related Topics

"Executing a Load Plan from ODI Studio" in the "Running Integration Processes" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

5.18 Stop Load Plan Dialog

Use to select the agent that stops the Load Plan.

From the Physical Agent list, select the agent that stops the Load Plan.

Click **OK** to stop the Load Plan.

Related Topics

"Stopping a Load Plan from ODI Studio" in the "Running Integration Processes" in *Oracle Fusion Middleware Administering Oracle Data Integrator*

Designer Navigator Editors

Oracle Data Integrator objects are created and edited in object editors. Oracle Data Integrator Studio provides several navigator-specific object Editors. This section lists the Oracle Data Integrator help topics for the object editors displayed in Designer Navigator.

This chapter includes the following topics:

- [Add Step Wizard](#)
- [Attribute Editor](#)
- [Attribute Matching Dialog](#)
- [Condition Editor](#)
- [Connection Path Selection Dialog](#)
- [Create a Lookup or a Join Dialog](#)
- [Create <Version> Dialog](#)
- [Cross Reference Editor](#)
- [Datastore Editor](#)
- [Designer Navigator](#)
- [Diagram Editor](#)
- [Edit Choice Dialog](#)
- [Export All Scenarios Dialog](#)
- [Export Work Repository Dialog](#)
- [Expression Editor](#)
- [Fetch Data for Reverse Engineering Dialog](#)
- [Filter Editor](#)
- [Folder Editor](#)
- [Generate DDL Editor](#)
- [Generate Mappings IN Editor](#)
- [Generate Mappings OUT Editor](#)
- [New <Object> Dialog](#)
- [Import <Object> Dialog](#)
- [Import Work Repository Dialog](#)

-
- Key Editor
 - Knowledge Module Editor
 - Load Plan Editor
 - Locked Objects Dialog
 - Lookup Scenario Dialog
 - Lookup Variable Dialog
 - Mapping Editor
 - Reusable Mapping Editor
 - 11g-Compatible Mapping Editor
 - Mapping Editor Overview Tab
 - Mapping Editor Concepts
 - Mapping Editor Validation Results Tab
 - Mapping Editor Logical Tab
 - Mapping Editor Physical Tab
 - Marker Group Editor
 - Model Editor
 - Model Folder Editor
 - Modify Case Step
 - Modify Run Scenario Step 1
 - Modify Run Scenario Step 2
 - New Mapping Dialog
 - New Scenario Dialog
 - Package Editor
 - Procedure Editor
 - Project Editor
 - Run <Object> Dialog
 - Reference Editor
 - Replace Object Dialog
 - Sequence Editor
 - Scheduling Editor
 - Run... Dialog
 - Scenario Editor
 - Scenario Generation Editor
 - Scenario Execution Report Editor
 - Scenario Variable Dialog
 - Shortcut Conflicts Dialog
 - Shortcut Editor

- Sign In Dialog
- Label Editor
- Path Selection Dialog
- Step Report Editor
- Sub-Model Editor
- Synchronizing Label Dialog
- Target Load Order Dialog
- User Function Editor
- Variable Editor
- HTTP Analyzer
- HTTP Analyzer Test Window
- HTTP Analyzer Instances Window
- HTTP Analyzer Dialog
- Proxy Settings Dialog
- Rule Settings Dialog
- HTTP Keystore Dialog
- Rename Credentials Dialog
- Select Web Service Description Dialog
- Certificate Error Dialog
- Dimension and Cube Accordion
- Dimension Editor
- Cube Editor
- Dimension Component Properties
- Cube Component Properties

6.1 Add Step Wizard

Use to add a step to a Load Plan.

How?

The Add Step Wizard has the following steps that vary depending on the type of step you are adding:

- Step 1 - Select a type and location of the step
- Step 2 - Select Step Name (for Serial, Parallel, Else, and Exception steps)
- Step 2 - Select Scenario (for Run Scenario steps)
- Step 2 - Select Variable to Evaluate (for Case steps)
- Step 2 - When Clause Evaluation (for When steps)
- Step 3 - Select Scenario Variables (for Run Scenario steps)

6.1.1 Step 1 - Select a type and location of the step

Use to specify the step type and the location of the step.

Properties	Description
Step Type	Type of step that is added. Possible step types are: Serial, Parallel, Run Scenario, Case, When, Else, and Exception if you start the Add Step Wizard from the Exception tab. See "Load Plan Structure" for more information on the different step types.
Step Location	This parameter defines where the step is added. <ul style="list-style-type: none"> ■ Add a child step to selection: The step is added under the selected step. ■ Add a sibling step after selection: The step is added on the same level after the selected step. ■ Add a sibling step before selection: The step is added on the same level before the selected step.

Note: Only values that are valid for the current selection are displayed for **Step Type** and **Step Location**.

Click **Next** to move to the next step of the Add Step Wizard.

6.1.2 Step 2 - Select Step Name (for Serial, Parallel, Else, and Exception steps)

Use to specify the name of the Load Plan step.

Enter a **Step Name** for the new Load Plan step.

Click **Finish** to complete the Add Step Wizard and add a new Load Plan step.

6.1.3 Step 2 - Select Scenario (for Run Scenario steps)

Use to select the scenario you want to add to your Load Plan.

Click **Lookup Scenario** to select the scenario.

The **Step Name** is automatically populated with the name of the scenario and the **Version** field with the version number of the scenario. The Step Name is editable and you can change it.

Tip: At design time, you may want to create a Run Scenario step using a scenario that does not exist. In this case, instead of selecting an existing scenario, enter a Scenario Name and Version directly and click **Finish**. Later, you can select the scenario using the [Modify Run Scenario Step 1](#) wizard.

Note that when you use the version number -1, the latest version of the scenario is used.

Click **Next** to move to the next step of the Add Step Wizard.

6.1.4 Step 2 - Select Variable to Evaluate (for Case steps)

Use to select the variable to evaluate.

Properties	Description
Variable	Variable to use for the conditional branching. Note: You can either select one of the Load Plan variables from the list or click Lookup Variable to add a new variable to the load plan and use it for this case step. See "Working with Variables" for more information.
Step Name	The Step Name is automatically populated with the step type and name of the variable. The Step Name is editable and you can change it.

Click **Finish** to complete the Add Step Wizard and add a new Load Plan step.

6.1.5 Step 2 - When Clause Evaluation (for When steps)

Use to define the When clause.

Properties	Description
Operator	Operator to use in the WHEN clause evaluation. Possible values are: <ul style="list-style-type: none"> ▪ Less Than (<) ▪ Less Than or Equal (<=) ▪ Different (<>) ▪ Equals (=) ▪ Greater Than (>) ▪ Greater Than or Equal (>=) ▪ Is not Null ▪ Is Null
Value	Value to use in the WHEN clause evaluation.
Step Name	The Step Name is automatically populated with the operator that is used. The Step Name is editable and you can change it.

Click **Finish** to complete the Add Step Wizard and add a new Load Plan step.

6.1.6 Step 3 - Select Scenario Variables (for Run Scenario steps)

Use to select variables that will be added to the Load Plan step.

In the Add to Load Plan column, select the scenario variables that you want to add to the Load Plan variables. If the scenario uses certain variables as its startup parameters, they are automatically added to the Load Plan variables.

Click **Finish** to complete the Add Step Wizard and add a new Load Plan step.

Related Topics

"Adding Load Plan Steps" in the "Creating a Load Plan" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.2 Attribute Editor

Use to create and configure attributes of your datastores.

Datstores describe data accessed by Oracle Data Integrator through two-dimensional tabular structures. Oracle Data Integrator contains the metadata related to the attributes. The data contained in the applications is stored in rows.

Attribute editors can be invoked by double-clicking the individual attributes or on the attributes tab from the corresponding data store.

The Attribute Editor has the following tabs:

- [Definition](#)
- [Description](#)
- [Control](#)
- [Markers](#)
- [Memo](#)
- [Version](#)
- [Privileges](#)
- [Flexfields](#)

6.2.1 Definition

Properties	Description
Name	The name of the attribute, as recognized by the data server.
Datatype	The datatype, corresponding to a type recognized by the data server.
Storage Type	This field is only displayed for HBase attributes. The storage type, using which you can define how a data type is physically stored in HBase. Possible values are <code>String</code> (default) and <code>Binary</code> .
Short Description	The descriptive label of the attribute. This label makes the user interface easier to read if the attribute names are not explicit.

Physical Format

The description of the physical storage format of the data.

Properties	Description
Order	The order number or physical rank of the attribute in the datastore.
Start	The physical position of the attribute. This information is to be given for fixed format files. The first position is position 1.
Length	The physical length in number of bytes. An attribute with start position '1' and length '4' occupies bytes 1,2,3,4 in each row. Note: For delimited files, the maximum length taken by the attribute should be shown. If the maximum length of the delimited file is not known, try to ensure that the length is greater than the maximum length. This length is used to reserve sufficient memory space for reading the information.

Properties	Description
Record codes	<p>The list of record codes that allows files with several record patterns to be processed.</p> <p>For example, for a file INVOICE containing different header and invoices rows marked by a record code CODE_REC with the value 'HEAD' or 'LINE', you must create a datastore for each of these conceptual entities. Specify HEAD in the list of record codes for the attribute CODE_REC of the datastore INVOICE, and LINE in the corresponding attribute for the datastore INVOICE_LINE.</p> <p>You can specify several values separated by the semicolon (;) character.</p>
Exclude the Record Codes	If this box is selected, the rows respecting the record codes are excluded (instead of included) when loading the file.

Logical Format

The description of the data representation.

Properties	Description
Logical Length	<p>The logical length of the attribute, as it is functionally seen by the user. For a number, this is the total number of numerals (Precision).</p> <p>Specify this length even if the physical length is given.</p>
Format	The format used for dates. This format is a Java date format that matches your computer's local parameters.
Scale	The number of digits that occur after the decimal.
Decimal separator	The decimal separator (by default, a period (.)).

Hive Column Type

This section is only displayed for Hive attributes.

Properties	Description
Used for Partitioning	<p>Select to use Hive columns for partitioning.</p> <p>Note: This value is filled in by reverse-engineering a Hive table and should not be changed manually.</p>
Used for Clustering	<p>Select to use Hive columns for clustering.</p> <p>Note: This value is filled in by reverse-engineering a Hive table and should not be changed manually.</p>
Used for Sorting	<p>Select to use Hive columns for sorting.</p> <p>Note: This value is filled in by reverse-engineering a Hive table and should not be changed manually.</p>

Data Service Operations

Use to configure the generation and the deployment of data services for this attribute.
More

Properties	Description
Allowed operations	Select the box that corresponds to the type of actions (INSERT, UPDATE, and SELECT) you want to allow data services to perform on this attribute. One important use of this tab is to lock an attribute against being written to by data services.

6.2.2 Description

Properties	Description
Default Value	The value inserted in this attribute if no value is specified.
Read only	Select this box if this attribute cannot be used in an INSERT or UPDATE command. This is the case, for example, for attributes whose values are automatically set and cannot be modified, such as IDENTITY attributes.
Slowly Changing Dimensions Behavior	<p>The attribute behavior when loading a slowly changing dimension table for OLAP. The following attribute behaviors are supported:</p> <ul style="list-style-type: none"> ■ Surrogate Key: The attribute is the unique (technical) identifier for a record version. This attribute is usually loaded by an automatically incremented value (sequence), and is referenced by the fact table. ■ Natural Key: The attribute is part of the key that identifies one record regardless of the versions. This key usually corresponds to the primary key of the source table. ■ Overwrite on Change: If this attribute's value changes, the current record version is updated, and this attribute is overwritten. ■ Add Row on Change: If this attribute's value changes, a new record version is created. ■ Current Record Flag: This attribute is the flag that identifies the current record version. It is usually set to 1 for the current version and to 0 for older versions. ■ Starting Timestamp: The valid starting date and time for the record version. ■ Ending Timestamp: The valid end date and time for the record version. <p>Some of the Knowledge Modules support slowly changing dimensions (SCDs). See the <i>Oracle Fusion Middleware Connectivity and Knowledge Modules Guide</i> for Oracle Data Integrator for more information.</p>
Description	The detailed description of the attribute.

6.2.3 Control

Use to display and configure the quality control properties.

The quality control properties are taken into account during a flow control, during a static control, or while loading a file.

Properties	Description
Mandatory	If selected, the attribute must be completed.
Control	Type of quality control for which the mandatory status is checked.
Flow	During a flow control for data integrated into this datastore, the mandatory status of the attribute is verified if this box is selected. This information is a default value that can be modified when the mapping is designed. It is recommended that you activate this type of check to ensure data quality, even if the target technology also verifies the information.
Static	During a static control, that is, when data already present in this datastore is checked, the mandatory status of this attribute is also verified if the box is selected.

Properties	Description
On Error	Data in the file may be inconsistent with the datastore definition. When reading the file, if one value from the row is inconsistent with the attribute description, the On Error option defines the action Oracle Data Integrator will perform.
Reject Error	The row containing the error is moved into a file with the BAD extension, and an explanation of the error is placed in a file with the ERROR extension. The BAD and ERROR files are located in the same directory as the file being read.
Null if error (inactive trace)	The row is kept in the flow and the erroneous value is replaced by null.
Null if error (active trace)	The row is kept in the flow, the erroneous value is replaced by null, and an explanation of the error is placed in a file with the ERROR extension.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator

6.3 Attribute Matching Dialog

The Attribute Matching Dialog is displayed when a connector is drawn to a projector component (see: [Projector Components](#)) in the Mapping Editor. The Attribute Matching Dialog gives you an option to automatically create expressions to map attributes from the source to the target (or in some cases, from the target to the source) component based on a matching mechanism. It also gives the option to create new attributes on the target based on the source, or new attributes on the source based on the target.

Connection Path

Use when the source and/or target connection have multiple connection points.

Properties	Description
From	Source connection point to be used in this connection. Select New to create a new connection point.
To	Target connection point to be used in this connection. Select New to create a new connection point.

Match Options

Properties	Description
By Name	Matches attribute names by exactly matching the name.
By Position	Matches attributes by matching the position: The first source attribute is mapped to the first target attribute, and so on.
Ignore Case	Ignores the case when matching attribute names.

Create Actions

Properties	Description
Create Attributes On Source	Creates every target attribute on the source if it does not exist already.
Create Attributes On Target	Creates every source attribute on the target if it does not exist already.
Auto Map	Creates map expressions on the target using the matching options. Newly created attributes are also mapped.
Preserve Existing Expressions	If checked, do not overwrite existing expressions during Auto Map.

Related Topics

"Attribute Matching" in the "Creating and Using Mappings" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator

6.4 Condition Editor

Use to define and configure conditions.

A *condition* is a WHERE-type SQL expression attached to a datastore based on an RDBMS that supports SQL. The purpose of this condition is to filter or check the data in the datastore in question.

The Condition Editor has the following tabs:

- [Definition](#)
- [Control](#)
- [Markers](#)
- [Memo](#)
- [Version](#)
- [Privileges](#)
- [Flexfields](#)

6.4.1 Definition

Properties	Description
Name	Name of the condition.
Type	<ul style="list-style-type: none"> ■ Database condition: A condition that exists in the database engine. ■ Oracle Data Integrator condition: A condition defined by the user that does not necessarily exist in the database engine.

Properties	Description
Where (Use the table alias: <table alias>)	<p>WHERE-type expression defining the condition. This expression must use the table alias (defined in the Datastore Editor) before each attribute.</p> <p>An example of an expression on the datastore CLIENT with the alias CLI: CLI.TYPE_CLIENT like 'A%'</p> <p>Use the Expression Editor to define the WHERE expression. Click Launch the Expression Editor to open it.</p> <p>Verify your expression by clicking Testing query on the DBMS.</p>
Message	Error message listed in the error table for rows in a datastore that contain an error.

6.4.2 Control

Use to display and configure the quality control properties.

The quality control properties are taken into account during a flow control or during a static control. Quality control information is not available for filter conditions, because filter conditions are not checked.

Properties	Description
Defined in the database	Shows if this condition is defined in the database engine. A condition that has been reversed has been defined in the database.
Active	Shows if the condition is active in the database. In some technologies, the dictionary may contain active (Enable) or inactive (Disable) conditions, that is, where the database does not check data consistency.
Control	Shows the type of control for which this condition will be checked.
Flow	During a flow control for this datastore, the condition will be verified if this box is selected. This information is a default value that can be modified when the mapping is designed. It is recommended to activate this type of control to ensure data quality, even if the target technology also checks the information.
Static	During a static quality control, that is, a quality control of data already present in this datastore, the condition will be verified if the box is selected.
Synchronous Control	Retrieves and displays the number of rows that do not respect the condition.
Rows not satisfying condition	Number of rows that do not respect the condition. This number is returned when the Check button is clicked.

Click the **Check** button to retrieve the number of records that do not respect this constraint. To obtain a list of the rows in error, a static control must be launched from either the Datastore or the Model Editor.

Related Topics

"Conditions" in the "Creating and Using Data Models and Datastores" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.5 Connection Path Selection Dialog

Use to map a selector component, or when the source is not connected to any projector component.

These properties are only shown if multiple connection points are possible.

Properties	Description
From	Source connection point to be used in this connection. Select one of the listed attributes to create a new connector point. This property is not applicable for connecting a datastore attribute to set.
To	Target connection point to be used in this connection. Select one of the listed attributes to create a new connector point.

Related Topics

"Connecting and Configuring Components" in the "Creating and Using Mappings" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.6 Create a Lookup or a Join Dialog

Use to specify if the relationship in the dataset will be a lookup or a join. More See [Lookups](#) and [Joins](#) for more information on these components.

This dialog can be restored by using the Bypass Create Lookup or Join user parameter.

Note: Without this dialog, dragging between attributes creates a join by default. To create a lookup, hold the Shift key while dragging between attributes.

Related Topics

"Create a Join or a Lookup" in the "Creating Joins and Lookups" section in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.7 Create <Version> Dialog

Use to check in a version for the selected object.

Version: Displays the version number of this object. A version number is automatically generated in the Version field. Modify this version number if necessary.

Description: The details for this version.

Click **Hide/Unhide Previous Versions** next to the Version field to expand the list of versions already checked in for this object.

Related Topics

"Working with Versions" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.8 Cross Reference Editor

Use to view a cross-reference.

Objects in Oracle Data Integrator are interlinked by relationships varying from simple usage associations to complex ones such as code-interpretation relationships. These relationships are implemented as *cross-references*. They are used to check and maintain consistency between related objects within a work repository.

The Cross Reference Editor has the following tabs:

- [Definition](#)
- [Version](#)
- [Privileges](#)

6.8.1 Definition

Properties	Description
Usage Mechanism	Type of cross-reference.
Object Type	ODI object, for example a datastore, a model, or an mapping.
Object Name	Name of the ODI object.
Reference Text	Text of the cross-reference.

Click **Display the original object** to display the Object Editor of the referenced object.

Related Topics

"Using Cross-References" in the "Organizing and Documenting Integration Processes" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.9 Datastore Editor

Use to create and configure your datastores in Oracle Data Integrator.

A datastore describes data as a table structure. Datastores are composed of attributes.

The Datastore Editor toolbar has the following tabs:

- **Datastore Static Control:** Executes Static Control on the current datastore.

The Datastore Editor has the following tabs:

- [Definition](#)
- [Attributes](#)
- [Files](#) (this tab appears only for File datastores)
- [Journalizing](#)
- [Partitions](#)
- [Markers](#)
- [Services](#)
- [Memo](#)
- [Version](#)
- [Privileges](#)
- [Flexfields](#)

The data of a datastore can be viewed or edited by using two specific editors. Right-click the datastore and selecting the appropriate editor:

- [View Data Editor:](#) This Editor is for viewing purposes only.

- [Data... Editor](#): This Editor is for viewing and editing purposes.

6.9.1 Definition

Properties	Description
Name	<p>The name of the datastore. This is the name that appears in the navigation trees and refers to the datastore from a Project (Mapping, and so forth). Do not use ODI reserved names, for example, <code>JRN_FLAG</code>, <code>JRN_SUBSCRIBER</code>, and <code>JRN_DATE</code> for the datastore name. These names cause Duplicate Attribute name SQL errors in ODI intermediate tables such as error tables. This name should be different from the resource name, if the latter is unknown or not very clear.</p> <p>For example, if the model contains tables named <code>C3556F</code> or <code>C5677D</code>, you can give them more explicit names, such as <code>CUSTOMER</code> or <code>ORDER</code>, leaving <code>C3556F</code> and <code>C5677D</code> as resource names.</p>
Datastore Type	The type of object the datastore represents.
Alias	<p>The short name used in check and filter expressions. This name must be used as a prefix for each attribute name to allow cross-references to be compiled.</p> <p>For example, for the table <code>CUSTOMER</code>, the alias <code>CUS</code> can be given. In this case, the expression of the condition <code>CLIENT_TYPE</code> could be <code>CUS.CLIENT_TYPE</code> in ('A', 'S', 'D').</p> <p>The alias name is not necessarily unique for all tables in a certain model. Its only purpose is to make writing expressions easier and to make managing cross-references possible.</p>
OLAP Type	<p>The table type in a multidimensional model for online analytic processing (OLAP):</p> <ul style="list-style-type: none"> ■ Fact table ■ Dimension ■ Slowly Changing Dimension
Resource Name	The name of the object in the form recognized by the data server that stores it. This may be a table name or a file name.
Buckets	<p>This field is only displayed for Hive datastores.</p> <p>Number of buckets to be used for clustering.</p>
Number of Rows	The number of datastore rows obtained by clicking Refresh .
Description	The detailed description of the datastore.

6.9.2 Attributes

Use to provide an overall view of the datastore attributes to be displayed, added, or deleted.

The **Reverse Engineer** and **Reverse COBOL Copybook** buttons allow you to reverse-engineer the attributes of a file in the following conditions:

Reverse Engineer

This button launches a standard reverse-engineering process for the current datastore. This reverse-engineering process is also possible for a delimited file. The attribute names are retrieved on the header line of the file, or are automatically generated. The length of the attributes are set with default values, and must be redefined manually.

Attribute Setup Wizard

If the file is a fixed file, the reverse button opens a wizard for defining the attributes.

Properties	Description
Fields	Shows the attributes and their values. Select a column to edit the attribute metadata.

Definition

Properties	Description
Attribute Name	Name of the selected attribute.
Datatype	Name of the datatype.
Heading	Name of the heading.

Physical Format

Properties	Description
Length	<p>The physical length in number of bytes. An attribute with start position '1' and length '4' occupies bytes 1,2,3,4 in each row.</p> <p>Note: For delimited files, the maximum length taken by the attribute should be shown. If the maximum length of the delimited file is not known, try to ensure that the length is greater than the maximum length. This length is used to reserve sufficient memory space for reading the information.</p>
Record Codes	<p>The list of record codes that allows files with several record patterns to be processed.</p> <p>For example, for a file INVOICE containing different header and invoices rows marked by a record code CODE_REC with the value 'HEAD' or 'LINE', you must create a datastore for each of these conceptual entities. Specify HEAD in the list of record codes for the attribute CODE_REC of the datastore INVOICE, and LINE in the corresponding attribute for the datastore INVOICE_LINE.</p> <p>You can specify several values separated by the semicolon (;) character.</p>
Exclude Record Codes	If this box is selected, the rows respecting the record codes are excluded (instead of included) when loading the file.

Logical Format

Properties	Description
Scale	The number of digits that occur after the decimal.
Format	The format used for dates. This format is a Java date format that matches your computer's local parameters.
Decimal Separator	The decimal separator (by default, a period (.)).

Miscellaneous

Properties	Description
Default Value	Default value of the attribute.

Properties	Description
SCD Behavior	Specifies the changing dimension behavior of the attribute.
Description	Detailed description of the attribute.

Check

Properties	Description
Mandatory	If selected, the attribute must be completed.

Version

Properties	Description
Created	Date and time when the object was created.
Created By	User who created the object.
Updated	Date and time when the object was updated.
Updated By	User who updated the object.

Reverse COBOL Copybook

This button launches a reverse-engineering process for a Fixed file when you have its description file as a COBOL Copybook format.

Automatic Adjustment

For a fixed file only, toggle to adjust starting positions automatically, as a function of column widths, to avoid gaps, and overlaps. It is not possible to manually specify the starting position of a column when Automatic Adjustment is enabled.

More

Click **Add Attributes** to add an attribute to the end of the list.

Click **Delete Attribute** to delete a selected attribute from the list.

You can use the arrows to move the range forward or backward.

This properties might change depending on the Source Technology selection.

Properties	Description
Order	Order in which attributes to be sequenced.
Name	Name of the attribute.
Type	Data type of the attribute.
Physical Length	Physical length of the attribute.
Logical Length	Logical length of the attribute.
Scale	Scale of the numeric attribute.
Decimal Separator	Decimal separator for the numeric attribute.
Rec. Code	Record flags to retrieve data from a file. Acts as a filter applied by the ODI File JDBC Driver.
Format	Logical format of the column (ex: dd/MM/yyyy).

Properties	Description
SCD Behavior	Specifies the changing dimension behavior of the attribute
Not Null	Specifies if the attribute can be null or not.

6.9.3 Files

Use to describe the datastores attached to a File technology.

This tab appears only for datastores attached to a File technology.

Properties	Description
File Format	Format of the file datastore: <ul style="list-style-type: none"> ▪ Delimited: The fields of a record are separated by a field separator. ▪ Fixed: The fields of a record are not separated, but their length is fixed.
Header (number of lines)	Number of records at the beginning of the file that are not data. These lines are ignored.
Record Separator	One or several characters separating records (or lines) in the file: <ul style="list-style-type: none"> ▪ MS-DOS: DOS carriage return ▪ Unix: UNIX carriage return ▪ Other: Free text you can enter as characters or hexadecimal codes
Field Separator	One ore several characters separating the fields in a record. <ul style="list-style-type: none"> ▪ Tabulation ▪ Space ▪ Other: Free text you can enter as characters or hexadecimal codes
Text Delimiter	Pair of characters delimiting a field. This delimiter escapes the field and record separators. If a separator is in a field - between the delimiters - it is not taken into account. The two left and right delimiter characters may be similar or different.
Decimal Separator	Character separating the integer and decimal part of a <i>numeric</i> character.

6.9.4 Journalizing

Use to display the journalizing status of the datastore and the list of subscribers tracking changes on this datastore, along with their subscription creation dates.

See "Introduction to Changed Data Capture" in the "Using Journalizing" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator* for more information.

6.9.5 Partitions

Properties	Description
Partition by	Partitioning method. This list displays the partitioning methods supported by the technology on which the model relies. If you want to define sub-partitions in addition to partitions, select the sub-partitioning method.
Sub-partition by	Sub-partitioning method. This list displays the sub-partitioning methods supported by the technology on which the model relies.

Toolbar Items

- **Add Partition:** Adds a partition at the end of the list
- **Add Sub-Partition:** Adds a sub-partition at the end of the list
- **Delete Partition:** Deletes the selected partition or sub-partition from the list

6.9.6 Services

Use to configure the generation and the deployment of data services for this datastore.
More

Properties	Description
Data Service Name	Name of the web service generated for this datastore.
Published entity	Name that is used to generate all operations names for the data service. For example, if you specify customer as the published entity, the generated operations are named addcustomer, getcustomer, and so forth.

6.9.7 View Data Editor

Use to view the data of a specific datastore.

The data is displayed in the form of a table.

If you want to narrow the displayed data, modify the SQL query by clicking **New Query** in the toolbar.

You can use the arrows in the toolbar to navigate through the records.

6.9.8 Data... Editor

Use to view and edit the data of a specific datastore.

The data is displayed in the form of a table.

If you want to narrow the displayed data, modify the SQL query by clicking **New Query** in the toolbar.

Using the toolbar, you can perform the following:

- Use the arrows to navigate through the records.
- Click **Insert new row** to insert a new row and **Delete current row** to delete the selected row.
- Click **Post changes to current row** to apply changes made to the current row and **Cancel the changes to current row** to cancel them.
- Click **Copy current row** to duplicate the selected row.
- Click **Save changes** to apply the changes you made to the data server.
- Click **Refresh data** to update the datastore's data from the data server.

Related Topics

"Reverse-engineering a File Model" in the "Creating and Reverse-Engineering a File Model" chapter in *Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator*

"Creating a Model" in the "Creating and Reverse-Engineering a Model" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.10 Designer Navigator

Use to work with projects, models, and labels in Oracle Data Integrator.

The Designer Navigator stores this information in a work repository.

The Designer Navigator has the following accordions:

- [Projects](#)
- [Models](#)
- [Load Plans and Scenarios](#)
- [Global Objects](#)
- [Labels](#)

Projects

Contains the developments made with Designer Navigator.

Models

Contains the descriptions of the data and applications structures.

Load Plans and Scenarios

Contains the Load Plans and Scenarios.

Global Objects

Contains the Global Knowledge Modules, User Functions, Variables, Markers, Sequences, and Global Reusable Mappings.

Labels

Contains the Labels that have been created when working with version management.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

6.11 Diagram Editor

Use to create and configure the diagrams.

A *diagram* is a graphical view of a subset of the datastores contained in a sub-model (or data model). A data model may have several diagrams attached to it.

A diagram is built:

- By assembling datastores from models and sub-models.
- By creating blank datastores into which you either create new attributes or assemble attributes from other datastores.

The Diagram Editor has the following tabs:

- Overview
 - [Definition](#)
 - [Markers](#)

- [Memo](#)
- [Version](#)
- [Privileges](#)
- [Diagram](#)

6.11.1 Definition

Properties	Description
Name	Name of the diagram.
Description	Description of the diagram.

6.11.2 Diagram

Use to edit all the model elements (datastores, attributes, references, filters, and so forth) visible in this diagram. Double-click the element you wish to edit to open the corresponding editor. Changes performed in the diagram immediately apply to the model. More

Related Topics

"Creating a New Diagram" in the "Using the Diagram" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.12 Edit Choice Dialog

Use to edit the default value of the Choice type in the Options tab of the Knowledge Module Editor and the Procedure Editor.

Properties	Description
Name	Name of the Choice type. Click the Name field to edit it.
Default	Select the default Choice type.

Tasks Toolbar

Name	Description
Add	Adds new items.
Delete	Deletes the selected item.
Move Up	Moves the selected item up in the list.
Move Down	Moves the selected item down in the list.
Move to top	Moves the selected item to the top of the list.
Move to bottom	Moves the selected item to the bottom of the list.

Related Topics

Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator

Oracle Fusion Middleware Developing Knowledge Modules with Oracle Data Integrator

6.13 Export All Scenarios Dialog

Use to export a group of scenarios.

Properties	Description
Export Directory	Directory in which the export file is created.
Child components export	If this option is checked, the objects linked to the object to be exported are also exported. These objects are those visible under the exported object in the Navigator tree. It is recommended to leave this option checked.
Replace existing files without warning	If this option is checked, the existing file is replaced directly by the ones of the export without any warning (no confirmation dialog will be displayed). If a file with the same name as the export file already exists, it is overwritten by the export file.

Objects to Export

Select the type of objects (packages, mappings, procedures, or variables) whose scenarios you want to export.

Encryption

For more information about Encryption see, [Export Key fields in Export Dialog](#)

Advanced Options

Use these options to customize the XML output file format. It is recommended that you use the default values.

Properties	Description
XML Version	XML Version specified in the export file. The xml version parameter in the XML file header. <code><?xml version="1.0" encoding="ISO-8859-1"?></code>
XML Character Set	Encoding specified in the export file. The encoding parameter in the XML file header. <code><?xml version="1.0" encoding="ISO-8859-1"?></code>
Java Character Set	Java character set used to generate the file.

Related Topics

"Exporting an Object with its Child Components" in the "Exporting and Importing" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.14 Export Work Repository Dialog

Use to export the work repository with its models and projects either as .xml files into a directory or as a zip file that contains the .xml files.

In the Export Work Repository Dialog set the export parameters as follows:

Properties	Description
Export to directory	Directory in which the export files are created.
Export to zip file	Select if you want to generate a zip file.
Zip File Name	Name of the zip file.

Encryption

For more information about Encryption see, [Export Key fields in Export Dialog](#)

Advanced Options

Use these options to customize the XML output file format. It is recommended that you use the default values.

Properties	Description
XML Character Set	Encoding specified in the export file. The encoding parameter in the XML file header. <pre><?xml version="1.0" encoding="ISO-8859-1" ?></pre>
Java Character Set	Java character set used to generate the file.

Related Topics

"Exporting an Object with its Child Components" in the "Exporting and Importing Objects" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.15 Expression Editor

Use to write expressions for mappings, filters, joins, or other queries.

By using the Expression Editor, you can avoid common syntax errors. It also contains a valuable reference for functions and other objects.

Syntax elements available in the Expression Editor are defined by the technology that runs this expression. This is the technology, for example of the source, target, or staging area data server when editing a mapping. The name of the technology currently used is indicated in the toolbar.

The Expression Editor is made up of the following panels:

- **Source datastores:** This panel appears on the left of the Expression Editor. When editing an expression for a mapping, this panel contains the names of source datastores and attributes.
- **Expression:** This panel appears in the middle of the Expression Editor. It displays the current text of the expression. You can directly type code here, or drag and drop elements from the other panels.
- **Technology functions:** This panel appears below the expression. Lists the language elements and functions appropriate for the given technology.
- **Variable, Sequences, User Functions and odiRef API:** This panel appears at the right of the technology functions and contains:
 - Project and Global Variables.
 - Project and Global Sequences, with the NEXTVAL and CURRVAL operators for each sequence.

Note: To use the native sequence's current value, you must select the **Supports native sequences** check box and configure **Local Sequence Current Value Mask** and **Remote Sequence Current Value Mask** in the Definition tab of the Technology Editor. You must also configure the **Sequence Current Value In Non-binded Mode** in the SQL tab of the Technology Editor.

- Project and Global User-Defined Functions.
- OdiRef Substitution Methods.

The toolbar buttons include standard editing functions:

- **Cut**
- **Copy**
- **Paste**
- **Undo**
- **Redo**
- **Highlighter drop-down menu:** Highlights the code part that matches the selected option.

Related Topics

"The Expression Editor" in the "Using Mapping Components" section in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.16 Fetch Data for Reverse Engineering Dialog

Use to retrieve data sample for reverse-engineering.

Properties	Description
Physical Agent	Name of the Physical Agent to retrieve data sample for reverse-engineering.
Number of Rows to Fetch	Value that indicates the number of rows to be fetched for reverse engineering the File data store.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

6.17 Filter Editor

Use to define and configure filters.

Filters can be placed on the source datastores of mappings to restrict the data to be used as source by the mapping.

Note: Filters defined in datastores will usually be used only if the filter is inside a Dataset component of the mapping.

The Filter Editor has the following tabs:

- [Definition](#)
- [Markers](#)

- [Memo](#)
- [Version](#)
- [Privileges](#)
- [Flexfields](#)

6.17.1 Definition

Properties	Description
Name	Name of the filter, as it appears in the Oracle Data Integrator user interface.
Filter Active for Static Control	Select Filter Active for Static Control to activate this filter during the static control.
Where	<p>WHERE-type expression defining the filter. This expression must use the table alias (defined in the Datastore Editor) before each column.</p> <p>An example of an expression on the datastore CLIENT with the alias CLI: <code>CLI.TYPE_CLIENT like 'A%'</code></p> <p>Use the Expression Editor to define the WHERE expression. Click Launch the Expression Editor to open it.</p> <p>Verify your expression by clicking Testing query on the DBMS.</p>

Related Topics

"Creating Filters" in the "Using Mappings Components" section in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.18 Folder Editor

Use to create and configure folders.

A *folder* is a group of packages, mappings, and specific procedures. Folders and sub-folders allow these objects to be grouped and organized according to criteria specific to the project. Sub-folders can be created to an unlimited number of levels.

Note: To create a sub-folder, create a folder, then drag and drop it into the parent folder. Alternatively, click on the **New Sub-Folder** context menu item of a Folder or Sub-Folder.

The Folder Editor has the following tabs:

- [Definition](#)
- [Markers](#)
- [Memo](#)
- [Version](#)
- [Privileges](#)

6.18.1 Definition

Properties	Description
Name	Name of the folder or sub-folder, as it appears in the Oracle Data Integrator user interface.

6.18.2 Version

Use to view and manage versions of your object.

A version is a backup copy of an object and is saved in the master repository. A version is checked in at a given time and may be restored later.

The version management system allows flags on developed objects (such as projects, models, and so forth) to be set automatically to indicate their status (such as new or modified). The version management system also allows these objects to be backed up as stable checkpoints, and later restored from these checkpoints. These checkpoints are created for individual objects in the form of versions.

Creation and Update Information

Properties	Description
Created by	User who created the object.
Created on	Date and time when the object was created.
Updated by	User who updated the object.
Updated on	Date and time when the object was updated.

Information

Properties	Description
Internal ID	Internal ID that enables object identification. <ul style="list-style-type: none"> ■ The internal ID is unique for the object type within a repository. ■ The internal ID for an object type contained in more than one repository is unique because it also contains the unique repository ID.
Global ID	Global ID of the object. The global ID is unique for the object type across repositories.
Release Tag Name	Release Tag name to be associated with a folder.

Versions

Properties	Description
Current Version: N/A (created)	<p>This table lists the version number, the creation date of the version, the name of the user who created the version, and a version description.</p> <p>You can perform the following actions by clicking the corresponding button, or by right clicking the Version and select a menu item under a version:</p> <ul style="list-style-type: none"> ■ Create a new version ■ Delete a version ■ Restore a version ■ Export a version ■ Compare a version <p>See "Working with Versions" for more information.</p>

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

6.19 Generate DDL Editor

Use to generate DDL scripts for your data models.

When a diagram or data model is designed or modified in Oracle Data Integrator, it may be necessary to implement the data model or the changes in the data server that contains the model implementation. This operation can be performed using DDL scripts generated by Oracle Data Integrator. The DDL scripts are generated in the form of procedures that contain DDL commands (create table, alter table, and so forth). This procedure may be executed on the data server to impact the changes.

Note: DDL scripts are generated based on the Action Group templates. Check that you have the appropriate action group for the technology of the model before starting DDL scripts generation. The action groups appear in the Topology Navigator.

Specify the following parameters in the Generate DDL Editor:

Action Group

The action group to be used to generate the DDL script.

Procedure Name

The name of the generated procedure.

Generation Folder

Oracle Data Integrator project folder in which the procedure is created.

You can click **Search**, to search for the project folder or click **Clear Folder Selection** if you want to remove the folder specified in this field.

Hierarchical/Linear View

The differences (new, deleted or modified) that exist between the data model defined in Oracle Data Integrator and the data server appear in a hierarchical or flattened view. In the hierarchical view, for example, a new column appears under a table node, and this table appears with a modified marker. In the flatten view, all the changes appear at the same level.

The differences that appear in these views depend on the filters that are checked in the **Filters** option group.

Select the changes you want to apply by checking the **Synchronization** column check box. You can also select or unselect all the differences currently displayed by using the buttons on top of the views.

Filters

Check boxes to filter the type of changes you want to display

For more information on how to create DDL Scripts, see "Generating DDL Scripts" in the "Creating Data Models with Common Format Designer" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*.

Related Topics

"Generating DDL Scripts" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.20 Generate Mappings IN Editor

Use to generate mappings to load a data model created using Common Format Designer.

Mappings IN are mappings that are used to load the model's datastores assembled from other datastores or attributes in a model diagram. They are the integration processes merging data from the original datastores into the composite datastores.

Specify the following parameters in the Generate Mappings IN editor:

Generation

- **Optimization Context:** The context used for creating the mappings. This context is used to automatically generate the flow of the mappings.
- **Generation Folder:** Oracle Data Integrator project folder into which the mappings are generated.

You can click **Search**, to search for the project folder or click **Clear Folder Selection** if you want to remove the folder specified in this field.

Candidate Datastores

- Select the **Generate Mapping** check box for the datastores that you want to load. Click **Select all the suggested mappings** if you want to select all mappings.
- Edit the **Mapping Name** attribute to rename the generated mappings.

Related Topics

"Generating Mapping IN/OUT" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.21 Generate Mappings OUT Editor

Use to automatically generate mappings to extract data from the datastores of a given data model. The automatically generated mappings are based on the existing mappings used to load data into these datastores. These mappings invert the existing data flows targeting this model.

Mappings OUT are generated using the mappings (including the mappings IN) that are already loading the model's datastores. They invert the integration process to propagate the data from the composite datastore to the original datastores.

For a given model or datastore assembled using Common Format Designer, Oracle Data Integrator is able to generate mappings OUT.

Note: The Generate Mappings OUT command does not take shortcuts into account.

Specify the following parameters in the Generate Mappings OUT editor:

Generation

- **Optimization Context:** The context used for creating the mappings. This context is used to automatically generate the flow of the mappings.
- **Generation Folder:** Oracle Data Integrator project folder into which the mappings are generated.

You can click **Search**, to search for the project folder or click **Clear Folder Selection** if you want to remove the folder specified in this field.

Candidate Datastores

This list displays the datastores from the current model as the **Target Tables**. For each of these datastores, if Oracle Data Integrator is able to find a datastore for which the data flow can be inverted, it presents it as a **Candidate Datastore**.

- Select the **Generation/Generate Mapping** check box to generate mappings for all or some of the candidate datastores attached to a given target table. Click **Select all the suggested mappings** if you want to select all mappings.
- Edit the **Mapping Name** attribute to rename the generated mappings.

Related Topics

"Generating Mapping IN/OUT" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.22 New <Object> Dialog

Use to create a new object.

Properties	Description
Name	Name of the object to be created.
Description	Detailed description of the object.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

6.23 Import <Object> Dialog

Use to import an object.

In the Import <Object> Dialog, set the import parameters as follows:

Properties	Description
Import Type	Select the Import Type. Possible values are: Duplication, Synonym Mode INSERT, Synonym Mode UPDATE, and Synonym Mode INSERT_UPDATE.
File import directory	Enter the directory from which you want to import the object.
Select the file(s) to import	From the list, select the files you want to import. Shows the build number and description for each KM.

Global ID Upgrade Key Dialog

Use to convert old IDs to a global unique identifier.

Properties	Description
Global ID Upgrade Key	Enter an upgrade key for converting object ID to a unique global ID.

Related Topics

"Importing Objects" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.24 Import Work Repository Dialog

Use to import a work repository.

In the Import Work Repository Dialog, set the import parameters as follows:

Properties	Description
Import Mode	Select the Import Mode. Possible values are: Synonym Mode INSERT, Synonym Mode UPDATE, and Synonym Mode INSERT_UPDATE.
Import From a Folder	Select if you want to import the work repository from a folder.
Import From a Zip File	Select if you want to import the work repository from a Zip file.

In the text field, enter the file import folder or zip file. Note that you can browse for the import folder or zip file.

Related Topics

"Importing Objects" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.25 Key Editor

Use to create and configure keys.

A *key* is a set of datastore attributes that enables each datastore row to be uniquely identified. If the key is also an index, it may also allow row access to be optimized. Some drivers retrieve key descriptions during the reverse-engineering process. It is also possible to define keys directly in the repository.

The Key Editor has the following tabs:

- [Description](#)

- [Attributes](#)
- [Control](#)
- [Markers](#)
- [Memo](#)
- [Version](#)
- [Privileges](#)
- [Flexfields](#)

6.25.1 Description

Properties	Description
Name	Name of the key.
Key or Index Type	Type of key or index: <ul style="list-style-type: none"> ■ Primary key: A unique key, preferred for all criteria, objective (all these columns must be completed) and subjective (conciseness, performance, and so forth). Oracle Data Integrator uses the primary key as the default update key when the datastore is a mapping target. ■ Alternate key: A unique key not used as the primary key. ■ Not unique index: An index used to improve access performance.

6.25.2 Attributes

Use to select the attributes of the table participating in the key.

6.25.3 Control

Use to display and configure the quality control properties.

The quality control properties are taken into account during a flow control, a static control, or while loading a file.

Defined in the Database: Shows if this key has been defined in the database dictionary (or in the described application). A key that has been reverse-engineered has been defined in the database. Datastore keys are generated **ONLY** when **Defined in the Database** is selected.

Active: Shows if the key is active in the database. In some technologies, the dictionary may contain active keys (Enable) or inactive keys (Disable), that is, where the data server does not check data consistency.

Control

Shows the type of quality control for which this key is checked.

Properties	Description
Flow	During a flow control in this datastore, the key is verified if this check box is selected. This information is a default value that can be modified when the mapping is designed. It is recommended to activate this type of control to ensure data quality, even if the target technology also checks the information.

Properties	Description
Static	During a static control, that is, during a check of the data already present in this datastore, the key is verified if the box is selected.

Synchronous Control

Properties	Description
Duplicated rows	Displays the number of duplicated rows. Click Check , to retrieve the number of records that respect or do not respect this constraint. To obtain a trace of the rows in error, a static control must be launched from either the Datastore or the Model Editor.

Related Topics

"Managing Security Settings" in *Oracle Fusion Middleware Administering Oracle Data Integrator*

6.26 Knowledge Module Editor

Use to create and customize your knowledge modules.

Knowledge Modules (KMs) are components of Oracle Data Integrator Open Connector technology. KMs contain the knowledge required by Oracle Data Integrator to perform a specific set of tasks against a specific technology or set of technologies.

The Knowledge Modules Editor has the following tabs:

- [Definition](#)
- [Tasks](#)
- [Options](#) (not for SKM)
- [Markers](#)
- [Memo](#)
- [Version](#)
- [Privileges](#)
- [Flexfields](#)

6.26.1 Definition

For an LKM, IKM, RKM, JKM, and CKM, the definition tab appears with the following fields.

Properties	Description
Name	Name of the knowledge module (KM), as it appears in the user interface.
Type	Type of KM
Consistent Set Journalizing / Simple Journalizing	This field appears only for a journalizing KM and defines which journalizing mode is managed by this knowledge module.
Default KM for this pair of technologies	Shows that the KM will be selected by default for the data servers with technologies shown in Source technology and Target Technology

Properties	Description
Multi-Connections	Shows if the KM has access to paired data servers. If this box is selected, it becomes possible to exchange data loading commands between two data servers, of different technologies if necessary. For example, on procedure commands, you would be able to specify a command "Select" in one database and a command "Insert into another database" to transfer the whole result.
Oracle GoldenGate JKM with JAgent	Marks the knowledge module as Oracle GoldenGate Knowledge Module to make visible JAgent configuration options in physical schema, logical schemas, and models. This field only appears for the Journalizing Knowledge Module.
Source Technology	Source technology used by default on the procedure commands. This information is only available if the procedure is "Multi-connections"-type.
Target Technology	The default technology to which the commands of this procedure will be addressed.
Description	Detailed description of the KM.

For an SKM, the tab appears with the following fields:

Properties	Description
Name	Name of the KM, as it appears in the user interface.
Command Text	Code of the SKM.
Description	Detailed description of the KM.
Source Technology	Source technology used by default on the procedure commands. This information is only available if the procedure is "Multi-connections"-type.
Target Technology	The default technology to which the commands of this procedure will be addressed.

There is no Details or Options tab for an SKM.

6.26.2 Tasks

A Knowledge Module task is the unit of command in the execution in the Knowledge Module task list. Each task contains information about the operation to be performed on the source and target.

General

Properties	Description
Name	Name of the command, as it appears in the user interface.
Type	Specifies the type of the task.
Log Counter	Shows which counter (Insert, Update, Delete or Errors) records the number of rows processed by this command.
Log Level	Logging level of the command. At execution time, commands can be kept in the log based on their log level.

Properties	Description
Ignore Errors	Shows that the procedure is not interrupted in case of an invalid return code. If this box is selected, the procedure command goes into "warning" instead of "error", and the procedure is not be stopped.
Log Final Command	Select to log final code for source and target commands. This improves the readability of logs which have a lot of runtime parameters. Use this option only for tasks that do not contain passwords or other sensitive data.nad

Journalizing

Properties	Description
Journalized Table in current source schema	If checked, the task is generated only if one of the source tables is journalized in the current source schema.

Create Temporary Indexes

Properties	Description
Create Temporary Indexes	If checked, the task is generated only for temporary indexes.

Target Command

Properties	Description
Technology	Technology on which the command is executed. Note: To use Oracle Data Integrator Tools (commands) in KM procedure commands or procedures commands, you must set the technology to ODI Tools. To use OS commands or to execute external programs, you must set the technology to Operating System
Transaction Isolation	Transaction isolation level for Select orders. The isolation levels shown are those of the SQL ISO. They are not supported by all data servers. The levels of isolation are: Default: The transaction keeps the connection's isolation level. Read Uncommitted: The transaction can read data not committed by another transaction. Read Committed: The transaction can only read data committed by other transactions (in general, this is the default mode of many data servers). Repeatable Read: The transaction is certain to read the same information if it executes the same SQL query several times, even if the rows have been modified and committed by another transaction in the meantime. Serializable: The transaction is certain to read the same information if it executes the same SQL query several times, even if the rows have been modified, deleted, or created and committed by another transaction in the meantime.
Context	Execution context of the query. If "Execution context" is left, the command will be executed in the execution context chosen on launching or on the step.

Properties	Description
Schema	Logical schema for execution. The logical schema, linked to the context, allows the physical execution schema to be defined.
Transaction	You can execute commands on several concurrent transactions (numbered from 0 to 9) or work off-transaction by choosing the option Autocommit .
Commit	<p>If your command is being executed in a transaction (numbered 0 to 9), you can decide to continue or to commit the current transaction according to the following modes:</p> <p>No Commit: The transaction is not committed. In this case, it can be committed in a later command. If a session ends normally, all transactions are committed by default.</p> <p>Commit: The transaction is committed.</p> <p>Commit 1000 rows: Oracle Data Integrator commits every 1000 records processed. This choice is only possible on a loading procedure command, that is, one containing a source command returning a result set and a destination command that carries out inserts.</p>
Command	<p>Text of the command expressed in a native language or in a standard language (such as SQL, PL/SQL, Transact-SQL, shell, and so forth). You can use the Expression Editor, accessible through the button to the right of the text.</p> <p>Warning: If the command is made of random characters, then the procedure is encrypted. For more details about encryption and decryption, see "Encrypting and Decrypting Procedures" in <i>Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator</i>.</p>

Source Command

Properties	Description
Technology	<p>Technology on which the command is executed.</p> <p>Note: To use Oracle Data Integrator Tools (commands) in KM procedure commands or procedures commands, you must set the technology to ODI Tools. To use OS commands or to execute external programs, you must set the technology to Operating System</p>
Transaction Isolation	<p>Transaction isolation level for Select orders. The isolation levels shown are those of the SQL ISO. They are not supported by all data servers.</p> <p>The levels of isolation are:</p> <p>Default: The transaction keeps the connection's isolation level.</p> <p>Read Uncommitted: The transaction can read data not committed by another transaction.</p> <p>Read Committed: The transaction can only read data committed by other transactions (in general, this is the default mode of many data servers).</p> <p>Repeatable Read: The transaction is certain to read the same information if it executes the same SQL query several times, even if the rows have been modified and committed by another transaction in the meantime.</p> <p>Serializable: The transaction is certain to read the same information if it executes the same SQL query several times, even if the rows have been modified, deleted, or created and committed by another transaction in the meantime.</p>

Properties	Description
Context	Execution context of the query. If "Execution context" is left, the command will be executed in the execution context chosen on launching or on the step.
Schema	Logical schema for execution. The logical schema, linked to the context, allows the physical execution schema to be defined.
Transaction	You can execute commands on several concurrent transactions (numbered from 0 to 9) or work off-transaction by choosing the option Autocommit .
Commit	<p>If your command is being executed in a transaction (numbered 0 to 9), you can decide to continue or to commit the current transaction according to the following modes:</p> <p>No Commit: The transaction is not committed. In this case, it can be committed in a later command. If a session ends normally, all transactions are committed by default.</p> <p>Commit: The transaction is committed.</p> <p>Commit 1000 rows: Oracle Data Integrator commits every 1000 records processed. This choice is only possible on a loading procedure command, that is, one containing a source command returning a result set and a destination command that carries out inserts.</p>
Command	<p>Text of the command expressed in a native language or in a standard language (such as SQL, PL/SQL, Transact-SQL, shell, and so forth). You can use the Expression Editor, accessible through the button to the right of the text.</p> <p>Warning: If the command is made of random characters, then the procedure is encrypted. For more details about encryption and decryption, see "Encrypting and Decrypting Procedures" in <i>Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator</i>.</p>

Options

Properties	Description
Always Execute	<p>If checked, the Task executes under any conditions. If not checked, the Task executes under the conditions set in the Options tab.</p> <p>Note: Only Boolean Knowledge Module options are shown in the Options tab.</p>
Is Executed	If checked, the Task executes under this condition individually.
Name	Name of the boolean option.
Description	Description of the boolean option.

Tasks Toolbar

Name	Description
Expand All	Expands all the task groups.
Collapse All	Collapses all the task groups in task hierarchy.
Add	Adds a new task.
Delete	Deletes the selected task.
Up	Moves the selected task up in the list.

Name	Description
Down	Moves the selected task down in the list.
Move task to top	Moves the selected task to the top.
Move task to bottom	Moves the selected task to the bottom.
Select Columns	Adds a task property to the task table.

6.26.3 Options

A KM can be parameterized when executed with options. The Options tab allows all options to be displayed, as well as their default values. The options values specified on this tab are only used during execution of the KM.

Properties	Description
Name	Name of the option. It is a common practice to use upper-case names with underscores ("_") such as DETECTION_STRATEGY.
Type	Datatype of the option. Possible values are Boolean, Choice, Text, or Value. Boolean options can be true or false. A choice option allows you to choose the option value from a pre-defined list of option values. These values can be edited using the Edit Choice Dialog . Value options can hold an alphanumeric value up to 250 characters, and Text options can hold an alphanumeric value of unlimited length and are parsed for variable and sequence references.
Default Value	Value of the option that is set by default.
Condition Expression	Click the condition expression cell against a particular option to enter or edit the groovy script, which will determine if a Knowledge Module option should be disabled, enabled, shown, or hidden.
Description	Text that will be shown in the Oracle Data Integrator UI next to the option. Text should be short to be readable in a single line inside a table.

Options Toolbar

Name	Description
Expand All	Expands all the option groups.
Collapse All	Collapses all the option groups.
Add	Adds a new option or option group.
Delete	Deletes the selected option or option group.
Move Up	Moves the selected option up within the option group or an entire option group up in the list of option groups. If an option is not within an option group, it will be moved up in the list of ungrouped options.
Move Down	Moves the selected option down within the option group or an entire option group down in the list of option groups. If an option is not within an option group, it will be moved down in the list of ungrouped options.
Move Out	Moves an option out of an option group. Note: This option is disabled if an ungrouped option or an option group is selected.

Name	Description
Move In	Moves an ungrouped option into an option group. Note: This option is disabled if an option within a group or an option group is selected.
Move to top	Moves the selected option to the top of the option group or an entire option group to the top of the list of option groups. If an option is not within an option group, it will be moved to the top of the list of ungrouped options.
Move to bottom	Moves the selected option to the bottom of the option group or an entire option group to the bottom of the list of option groups. If an option is not within an option group, it will be moved to the bottom of the list of ungrouped options.
Select Column	Selects or deselects the columns to be displayed in the table.

6.26.4 Version

Use to view and manage versions of your Knowledge Module.

A version is a backup copy of an object and is checked in at a given time and may be restored later. Versions are saved in the Master Repository.

The version management system allows flags on developed objects (such as projects, models, and so forth) to be set automatically to indicate their status (such as new or modified). The version management system also allows these objects to be backed up as stable checkpoints, and later restored from these checkpoints. These checkpoints are created for individual objects in the form of versions.

Creation and Update Information

Properties	Description
Created by	User who created the knowledge module.
Created on	Date and time when the object was created.
Updated by	User who updated the knowledge module.
Updated on	Date and time when the object was updated.

Information

Properties	Description
Internal ID	Internal ID that enables object identification. <ul style="list-style-type: none"> The internal ID is unique for the object type within a repository. The internal ID for an object type contained in more than one repository is unique because it also contains the unique repository ID.
Build Number	Internal build number to track the release.
Customization Status	Can take the following values: <ul style="list-style-type: none"> Oracle Genuine: The KM is a non-modified original KM provided by Oracle. Modified by User: The KM is an original KM provided by Oracle that has been modified by the user. Non-Oracle: The KM is a non-Oracle KM.

Versions

Properties	Description
Current Version: N/A (created)	<p>This table lists the version number, the creation date of the version, the name of the user who created the version, and a version description.</p> <p>You can perform the following actions by clicking the corresponding button:</p> <ul style="list-style-type: none"> ■ Create a new version ■ Delete a version ■ Restore a version ■ Export a version <p>See "Working with Version" for more information.</p>

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator

Oracle Fusion Middleware Developing Knowledge Modules with Oracle Data Integrator

Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator

6.27 Load Plan Editor

Use to create and configure a Load Plan.

The Load Plan is the largest execution unit in Oracle Data Integrator. It contains a hierarchy of steps that can be executed conditionally, in parallel or in series. The leaves of this hierarchy are scenarios. Packages, mappings, variables, and procedures are started by Load Plans in the form of scenarios. More

In the Load Plan editor toolbar, click **Validate** to verify whether the Load PLAN contains any errors and click **Execute** to start the Load Plan.

The Load Plan Editor has the following tabs:

- [Definition](#)
- [Steps](#)
- [Exceptions](#)
- [Variables](#)
- [Privileges](#)
- [Version](#)
- [Flexfields](#)

6.27.1 Definition

Properties	Description
Name	Name of the Load Plan.

Properties	Description
Folder Name	Name of the Load Plan and Scenario folder where this Load Plan is located. If the Load Plan is not created within a folder, this field is empty.
Keep Log History (days)	Number of days for which log history is kept for this Load Plan.
Log Sessions	Indicates how logs are preserved for the sessions started by the Load Plan. Possible values are: <ul style="list-style-type: none"> ■ Always Keep: Always keep session logs (default). ■ Never: Never Keep session logs. <p>Note: This has no effect on Run Scenario steps that are configured as <i>Restart from Failed Step</i> or <i>Restart From Failed Task</i> as the whole session needs to be preserved for restartability.</p> <ul style="list-style-type: none"> ■ Error: Only keep the session log if the session completed in an error state.
Log Session Step	Indicates how logs are maintained by default for the session steps of each of the sessions started by the Load Plan. Note that this applies only when the session log is preserved. Possible values are: <ul style="list-style-type: none"> ■ By Scenario Settings: Session step logs are preserved depending on the scenario settings. Scenario created from packages can preserve or not the steps depending on the advanced step property called Log Steps in the Journal. Other scenarios preserve all the step (Default). ■ Never: Never keep session step logs. <p>Note: This has no effect on Run Scenario steps that are configured as <i>Restart from Failed Step</i> or <i>Restart from Failed Task</i> as the whole session needs to be preserved for restartability.</p> <ul style="list-style-type: none"> ■ Errors: Only keep session step log if the step is in an error state.
Session Tasks Level Logging	Indicates the task level logging (from 1 to 5). This value corresponds to the <i>Log Level</i> value when starting unitary scenarios. Default is 5. Note that when Run Scenario steps are configured as <i>Restart from Failed Step</i> or <i>Restart From Failed Task</i> , this parameter is ignored as the whole session needs to be preserved for restartability.
Keywords	Comma separated list of keywords that are set on the sessions started from this Load Plan. These keywords improve the organization of ODI logs by session folders and automatic classification. <p>Note: You can override these keywords at the level of the child steps.</p>
Description	Description of the Load Plan.

Concurrent Execution Controller

Properties	Description
Limit Concurrent Executions	If checked, enables the Concurrent Execution Controller for the Load Plan.
Violation Behavior	Specifies the violation behavior action. Default is <code>Raise Execution Error</code> and <code>Wait to Execute</code> .

Properties	Description
Wait Polling Interval	<p>Specifies the polling frequency (in seconds) for the wait behavior to check for its turn to run. This field accepts only a positive integer, and it is enabled only when Violation Behavior is selected as Wait.</p> <p>Note: A newly generated scenario or a newly created load plan has blank as the default value, meaning that the runtime agent determines the actual value used for the polling (currently the default is 30 seconds in the runtime agent).</p>

6.27.2 Steps

Use to define and view the steps hierarchy and step properties.

The *Steps Hierarchy* table defines the organization of the steps in the Load Plan. Each row in this table represents a step. You can drag components such as packages, mappings, variables, procedures, or scenarios from the Designer Navigator into the Steps Hierarchy table for creating Run Scenario steps for these components.

The Steps Hierarchy table contains the following columns:

- **Steps Hierarchy:** Displays the steps hierarchy in a tree showing the name of each step.
- **Enabled:** Defines whether a step is enabled or disabled. Disabling a step also disables all its child steps. Disabled steps and all their child steps are not executed when you run the Load Plan.
- **Scenario/Variable:** Displays the name of the scenario for Run Scenario steps or the name of the variable evaluated in a Case step.
- **Restart:** Defines the restart behavior of a step when the Load Plan is restarted. More
- **Context:** Displays the context that is used for the step execution. Default context is the Load Plan context that is defined in the Start Load Plan Dialog when executing a Load Plan.
- **Logical Agent:** Displays the logical agent that is used for the step execution. By default, the logical agent, which is defined in the Start Load Plan Dialog when executing a Load Plan, is used.

The Load Plan Editor toolbar, located on top of the Steps Hierarchy table, provides tools for creating, organizing, and sequencing the steps in the Load Plan. [Table 6-1](#) details the different toolbar components.

Table 6-1 Load Plan Editor Toolbar

Name	Description
Search	Searches for a step in the Steps Hierarchy table.
Expand All	Expands all tree nodes in the Steps Hierarchy table.
Collapse All	Collapses all tree nodes in the Steps Hierarchy table.
Add Step	Opens an Add Step menu. You can either select the Add Step Wizard or a Quick Step tool to add a step. See "Adding Load Plan Steps" for more information.
Remove Step	Removes the selected step and all its child steps.

Table 6–1 (Cont.) Load Plan Editor Toolbar

Name	Description
Navigation arrows: Move Up, Move Down, Move Out, Move In	Use the navigation arrows to move the selected step to the required position.

The *Property Inspector* displays the step properties for the step that is selected in the Steps Hierarchy table.

The following sections describe the Load Plan Step properties displayed in the Property Inspector:

- [Common Properties](#)
- [Serial Step Properties](#)
- [Parallel Step Properties](#)
- [Run Scenario Step Properties](#)
- [Case Step Properties](#)
- [When Step Properties](#)
- [Else Step Properties](#)

6.27.2.1 Common Properties

This section describes the following properties common to all Load Plan steps:

- [Steps Properties](#)
- [Exception Handling](#)
- [Advanced](#)
- [Variables](#)

Steps Properties

Properties	Description
Name	Displays the name of the Load Plan step
Step Type	Displays the type of the step. You cannot modify the step type. If you want to change the step type, you have to delete the step and create a new one.
Enabled	Select Enabled to execute the step and its subsequent steps. Note: Disabled steps are not executed in the Load Plan regardless of the enabled/disabled property of their sub-elements. When a step is created, it is enabled by default.
Keywords	Displays the keywords that are attached to the sessions of the scenario execution. The keywords are passed to the steps under this step in the hierarchy. These keywords improve the organization of ODI logs by session folders and automatic classification. If you use several keywords, use a comma (,) to separate the keywords. If no keywords are specified, the keywords are inherited from the parent step.

Exception Handling

Properties	Description
Timeout(s)	<p>Defines a maximum time that this step takes before it is aborted by the Load Plan. When a timeout is reached, the step is marked in error and the Exception step (if defined) is executed. In this case, the exception step never times out. If needed, a timeout can be set on a parent step to safeguard such a potential long running situation.</p> <p>If the step fails before the timeout and an exception step is executed, then the execution time of the step plus the execution time of the exception step should not exceed the timeout, otherwise the exception step will fail when the timeout is reached.</p> <p>Note: The default value of zero (0) indicates an infinite timeout.</p>
Exception Step	<p>Displays the name of the step that is executed if this Load Plan step fails. See "Handling Load Plan Exceptions and Restartability" for more information about exception steps.</p>
Exception Behavior	<p>Defines how this step behaves in case an exception is encountered. Possible values are:</p> <p>Run Exception and Raise: Runs the Exception step of the Load Plan and raises the exception to its parent. If no exception step is defined, it only raises the exception to its parent. This is the default value.</p> <p>Run Exception and Ignore: Runs the Exception step of the Load Plan and ignores the exception. The parent step is notified of a successful run. If no exception step is defined, it notifies the parent step of the successful run. If an exception is caused by the exception step itself, the parent step is notified of the failure.</p> <p>See "Handling Load Plan Exceptions and Restartability", for more information on how to create Exception steps and how to define the exception flow.</p>

Advanced

- **Order:** Displays the execution order of this step within the parent step.

Variables

This section is only displayed for Serial, Parallel, Run Scenario, and Case steps. The Variables section displays and defines how the variables are handled in this step.

Note: When starting a Run Scenario step, all the variables that are common to the closest parent step and the scenario are passed to the scenario and overwrite the variable default values of this scenario.

Properties	Description
Variable	Displays the fully qualified name of the variable.
Data Type	Displays the datatype of the variable.
Overwrite	Select Overwrite , if you want to specify a variable value for this step and all its children. Otherwise the step inherits the variable value from its parents.
Value	Displays the variable value used for this step and all its children, if Overwrite is selected.
Refresh	Select Refresh to refresh this variable prior to executing the step.

See "Working with Variables in Load Plans" in the "Using Load Plans" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator* for more information.

6.27.2.2 Serial Step Properties

This section describes the step properties that are specific to Serial steps.

Exception Handling

Properties	Description
Restart Type	<p>Defines the restart behavior of this step when the Load Plan is restarted. Possible values are:</p> <ul style="list-style-type: none"> ▪ Restart all children: When the Load Plan is restarted and if this step is in error, the sequence of steps restarts from the first one. ▪ Restart from failure: When the Load Plan is restarted and if this step is in error, the sequence of child steps starts from the one that has failed. This is the default value.

6.27.2.3 Parallel Step Properties

This section describes the step properties that are specific to Parallel steps.

Exception Handling

Properties	Description
Restart Type	<p>Defines the restart behavior of this step when the Load Plan is restarted. Possible values are:</p> <ul style="list-style-type: none"> ▪ Restart all children: When the Load Plan is restarted and if this step is in error, all the child steps are restarted regardless of their status. This is the default value. ▪ Restart from failed children: When the Load Plan is restarted and if this step is in error, only the failed child steps are restarted in parallel.
Max Error Child Count	<p>Displays the maximum number of sub-elements in error that is accepted before this step is to be considered in error. Default is 0. When the Max Error Child Count is reached, all errors are buffered until all pending sessions are stopped or completed. The final behavior depends on the selected Restart Type:</p> <ul style="list-style-type: none"> ▪ If the Restart type is Restart from failed children, it waits for all sessions (these are the currently running sessions and the ones waiting to be executed) to run and complete before it raises the error to the parent step. ▪ If the Restart Type is Restart all children, it stops all running sessions and does not start any new ones before it raises the error to the parent.

6.27.2.4 Run Scenario Step Properties

This section describes the step properties that are specific to Run Scenario steps.

Step Properties

Scenario: Displays the scenario name and version that is executed in this step.

Click **Lookup Scenario** to launch the [Modify Run Scenario Step 1 Wizard](#) and change the scenario.

Exception Handling

Properties	Description
Restart Type	<p>Defines the restart behavior of this step when the Load Plan is restarted. Possible values are:</p> <ul style="list-style-type: none"> ▪ Restart from new session: When restarting the Load Plan and this scenario step is in error, start the scenario and create a new session. This is the default value. ▪ Restart from failed step: When restarting the Load Plan and this scenario step is in error, restart the session from the step in error. All the tasks under this step are restarted. This option is restricted by the settings of the Session Level Logging, Session Step Logging, and Task Level Logging defined in Definition tab of the Load Plan Editor. ▪ Restart from failed task: When restarting the Load Plan and this scenario step is in error, restart the session from the task in error. This option is restricted by the settings of the Session Level Logging, Session Step Logging, and Task Level Logging defined in the Definition tab of the Load Plan Editor. <p>When restarting from a failed step or task, all the database sessions are reopened and any uncommitted changes are lost.</p> <p>Caution: When restarting from a failed step or task, all the database sessions are reopened and any uncommitted changes are lost. These options should be used with caution as they may result in data loss.</p> <p>The same limitation as those described in "Restarting a Session" apply to the sessions restarted from a failed step or failed task.</p>

Advanced

Properties	Description
Priority	<p>Priority for this step when its parent is a parallel step. The integer value range is from 0 to 100 (100 being the highest priority). The default value is 0. The priority of a Run Scenario step is evaluated each time Load Plan has a set of runnable scenarios to be executed during the advancing of a Load Plan. Runnable scenarios are the ones that are available for running after a prior scenario completes its execution. The Run Scenario step with the highest priority is executed first, however, once a scenario is submitted or running for execution based on the priority evaluation during Load Plan advancing, it will not be unseated in the future by a Run Scenario with a higher priority.</p>
Context	<p>Context that is used for the step execution. The default context is the Load Plan context that is defined in the Start Load Plan Dialog when executing a Load Plan.</p>
Logical Agent	<p>Logical agent that is used for the step execution. By default, the logical agent, which is defined in the Start Load Plan Dialog when executing a Load Plan, is used.</p>

Scenario Variables

This section displays the scenario variables used in this step.

Properties	Description
Variable	Displays the fully qualified name of the variable.
Data Type	Displays the datatype of the variable.
Overwrite	Select Overwrite , if you want to specify a variable value for this step and all its children. Otherwise the step inherits the variable value from its parents.
Value	Displays the variable value used for this step and all its children, if Overwrite is selected.
Refresh	Select Refresh to refresh this variable prior to executing the step.

Other Load Plan Variables

This section displays the variables that are available for this step but that are not used in this step.

The properties are the same as for Scenario Variables.

6.27.2.5 Case Step Properties

This section describes the step properties that are specific to Case steps.

Test Variable: Displays the fully qualified name of the variable that is used for evaluating the tests defined in the WHEN statements.

Click **Lookup Variable** to launch the [Modify Case Step](#) Wizard and change the test variable.

6.27.2.6 When Step Properties

This section describes the step properties that are specific to WHEN steps.

Case Statement Variable: Displays the fully qualified name of the variable that is used in the WHEN clause evaluation. This variable is the one defined in the parent Case step.

Select the **Operator** to use in the WHEN clause evaluation. Possible values are:

- Less Than (<)
- Less Than or Equal (<=)
- Different (<>)
- Equals (=)
- Greater Than (>)
- Greater Than or Equal (>=)
- Is not Null
- Is Null

Value: Displays the value of the variable that is used in the WHEN clause evaluation.

6.27.2.7 Else Step Properties

This section describes the step properties that are specific to Else steps.

Case Statement Variable: Displays the fully qualified name of the variable that is used in the WHEN clause evaluation. This variable is the one defined in the parent Case step.

6.27.3 Exceptions

Use to define and view the exception steps.

This tab displays the list of *Exception Steps* in the Steps Hierarchy table. Each Exception Step consists of a hierarchy of Load Plan steps. When a step of the Load Plans results in an error, the associated exception step is executed automatically. You can associate a given exception step to one or more steps in the Load Plan Editor.

The Exceptions tab is similar to the Steps tab in the Load Plan editor. The main differences are:

- There is no root step for the Exception Step hierarchy. Each exception step is a separate root step. An Exception step contains only common step properties. See [Common Properties](#) for more information on these properties.
- The Serial, Parallel, Run Scenario, and Case steps have the same properties as on the Steps tab but do not have an Exception Handling properties group. An exception step that results in an error cannot raise another exception step. See [Steps](#) for more information on the step properties

See "Handling Load Plan Exceptions and Restartability" for more information on how to define the exception flow and how to use exceptions in Load Plans.

6.27.4 Variables

Use to declare and edit the Load Plan variables.

Project and Global Variables used in a Load Plan are declared as Load Plan variables. Load Plan variables are automatically available in all steps and their value passed to the Load Plan steps.

The value of the variables are passed to the Load Plan on startup. At a step level, you can override the variable value (by setting it or forcing a refresh) for this step and its child steps.

You can use variables in Run Scenario steps - the value of the variable are passed as startup parameters to the scenario - or in Case/When/Else steps for conditional branching.

See "Working with Variables in Load Plans" in the "Using Load Plans" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator* for more information.

The Variable table displays the Load Plan variables and contains the following columns:

- **Variable:** Displays the fully qualified name of the variable
- **Data Type:** Displays the datatype of the variable
- **Logical Schema:** Displays the logical schema where the SELECT statement for refreshing the variable value is executed
- **Description:** Displays the description of the variable
- **Select Statement:** Displays the SELECT statement of a Refresh variable.

Related Topics

"Introduction to the Load Plan Editor" in the "Using Load Plans" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

"Executing a Load Plan" in *Oracle Fusion Middleware Administering Oracle Data Integrator*

6.28 Locked Objects Dialog

Use to lock and unlock objects.

The object locking mechanism is automatically activated in Oracle Data Integrator when closing Oracle Data Integrator Studio, or it must be activated manually.

When an object is locked, only the user owning the lock can perform modifications on the object, such as editing or deleting. Other operations, such as executing, can be performed by other users, but with a warning displayed.

Note: You can keep objects locked even if you are not connected to Oracle Data Integrator. This allows you to prevent other users from editing them.

In the Locked Objects Dialog, set the locking parameters as follows:

Properties	Description
User	Set the filter on the user who locked the object.
Object Type	Set the filter on the type of object that is locked.
Unlock	Select if you want to unlock the object.
Type	Type of object that is locked.
Name	Name of the locked object.
Locked by	User who has locked the object.
Locked Since	Date and time when the object was locked.

You can lock or unlock all displayed objects at the same time by clicking **Lock all** or **Unlock all**.

Related Topics

"Object Locking" in the "Organizing and Documenting Integration Projects" chapter in the *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.29 Lookup Scenario Dialog

Use to look up a scenario or an executable object.

This dialog box displays all the available scenarios and executable objects in a tree. From the tree, select the scenario or executable object you want to use in the Load Plan step and click **OK**.

This dialog box displays cached object names. If you modify an object name while the Lookup Scenario Dialog is open, the dialog box still displays the original name. To update the dialog box, close and reopen it. The latest object names are loaded when the Lookup Scenario dialog box is reopened.

Note: If you select an executable object, as for example a Package or a Mapping, ODI generates automatically a new scenario for it.

Related Topics

"Defining the Load Plan Step Sequence" in the "Using Load Plans" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.30 Lookup Variable Dialog

Use to find a variable.

This dialog box displays all the available (global and project) variables in a tree. From the tree, select the variable you want to use in the Load Plan step and click **OK**.

Find: Enter all or part of the variable name to narrow your search.

Related Topics

"Defining the Load Plan Step Sequence" in the "Using Load Plans" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.31 Mapping Editor

Use to create and configure your mappings in Oracle Data Integrator.

A mapping consists of a set of rules that defines the loading of one or more target data stores, from one or more source data stores.

More

The Mapping Editor is organized into three main tabs. Each main tab is made up of several tabs and components:

- [Mapping Editor Overview Tab](#)
- [Mapping Editor Logical Tab](#), also called the Logical Diagram
- [Mapping Editor Physical Tab](#), also called the Physical Diagram

6.31.1 Connector Points

Connector points define the connections between components inside a mapping. Each component can have input and output connector points. The component type might place limitations on how many connector points are allowed.

Input Connector Points

Properties	Description
Name	Name of the connector point. This field can be edited.
Description	Description of the connector point.
Bound Object	Name of the object to which the connector point is bound. This field remains blank if the component type does not support connector point binding.
Connected From	Names of the preceding components this component connects from.

Output Connector Points

Properties	Description
Name	Name of the connector point. Can be changed by the user.
Description	Description of the connector point.
Bound Object	Name of the object to which the connector point is bound. This field remains blank if the component type does not support connector point binding.

Properties	Description
Connected To	Names of the succeeding components this component connects to.

Related Topics

"Creating and Using Mappings" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.32 Reusable Mapping Editor

Use to create and configure reusable mappings in Oracle Data Integrator.

A reusable mapping is a mapping fragment that can be included in mappings or other reusable mappings. A reusable mapping can define zero to many input and output signatures to pass data from and to the included mapping. Similar to a regular mapping, a reusable mapping can also include datastores as sources and targets, as well as other components.

More

The Reusable Mapping Editor is organized into three main tabs. Each main tab is made up of several sub-tabs and components:

- [Mapping Editor Overview Tab](#)
- [Reusable Mapping Editor Logical Tab](#), also called the Logical Diagram

6.32.1 Reusable Mapping Editor Logical Tab

The logical mapping rules from input to output signatures are defined in the Logical tab of the Reusable Mapping Editor.

More

The Logical Diagram contains components that are linked through connectors with arrow heads pointing the direction of the flow from source to target. Each component has left and right connector ports to anchor incoming and outgoing connectors.

In order to create a Logical Diagram perform the following steps:

1. Drag and drop input and output signature components from the Component Palette into the appropriate areas. Add attributes to signatures as necessary. You might use the attribute matching capabilities (see: [Attribute Matching Dialog](#)) later to populate the signatures with attributes from other components.
2. If applicable, drag and drop datastores or reusable mappings from the designer navigator into the diagram. Source datastores and reusable mappings can also be organized within a [Dataset Component](#).
3. Drag and drop components such as joins, filters, or others from the component palette into the component area to define rules for the reusable mapping between input and output signatures. If datasets are used, join, lookup, or filter rules can be defined by dragging attributes between datastores and onto the dataset background.
4. Create connectors between components by dragging from the originating connector port to the destination connector port. Connectors can also be implicitly created by dragging attributes between components. When creating a connector between two ports, an [Attribute Matching Dialog](#) may be shown to automatically map attributes based on name or position.

- Set component properties, expressions, and conditions by selecting components and editing the properties in the Property Inspector.

Mapping

Properties	Description
Name	Name of the reusable mapping.
Description	Detailed description of the reusable mapping.
Target Load Order	List of targets that are loaded in a defined order. Uses the Target Load Order Dialog to select and order targets.

Related Topics

"Reusable Mappings" in the "Creating and Using Mappings" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.33 11g-Compatible Mapping Editor

11g compatible mappings maintain compatibility to the 11g SDK, so that existing programs using the ODI SDK continue working in 12c.

11g-compatible mappings can only be created in the following ways:

- Upgrade from an 11g work repository, and in the Upgrade Assistant, de-select **Don't Maintain ODI 11g SDK Compatibility**.
- Through commands of the 11g-compatible SDK.

11g-compatible mappings are presented under the mapping node in the Designer Navigator. 11g-compatible mappings cannot be created or modified in ODI Studio; a mapping opened in the mapping editor is read-only. These mappings are distinguished from 12c mappings by an 11g interface icon.

Related Topics

"Using Compatibility Mode" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.34 Mapping Editor Overview Tab

Use to set general properties and review executions, scenarios, markers, memos, versions, privileges, and flexfields of a mapping.

The following tabs are available:

- [Definition Tab](#)
- [Execution Tab](#)
- [Scenarios Tab](#)
- [Markers Tab](#)
- [Memo Tab](#)
- [Version Tab](#)
- [Privileges Tab](#)
- [Flexfields Tab](#)

6.34.1 Definition Tab

This tab defines the general properties of a mapping.

Properties	Description
Name	Name of the mapping as it appears in the Oracle Data Integrator user interface.
Description	Detailed description of the mapping.

6.34.2 Execution Tab

A mapping can be executed directly without generating a scenario. Execution is launched by clicking **Run...** in the menu bar.

The Execution tab is organized into the Direct Executions and Scenario Execution tabs and shows the results of previous executions.

Properties	Description
Agent	Name of the agent that executed the scenario. Internal indicates that the agent used is the one built in the Oracle Data Integrator Studio.
Context	Execution context of the scenario.
Status	State of the scenario (Done, Error, Running, Waiting, Warning).
Start	Start date and time of execution of the scenario.
End	End date and time of execution of the scenario.
Duration	Time taken for execution of the scenario.
Return Code	Return code of the scenario.
Message	Scenario execution error message.
Rows	Total number of rows processed by the scenario.
Inserts	Number of rows inserted during the scenario.
Updates	Number of rows updated during the scenario.
Deletes	Number of rows deleted during the scenario.
Errors	Number of rows in error in the scenario.

6.34.3 Scenarios Tab

This tab displays in a table view the scenarios generated for this component, as well as their schedules. Double-click a scenario or a schedule to display its properties.

Create and delete scenarios by clicking the **Generate Scenario** and **Delete** buttons.

The popup menus for scenarios and schedules are also available from this tab.

Properties	Description
Name	Name of the scenario.
Version	Version of the scenario.

6.34.4 Markers Tab

Markers can be defined in a Project (Project Markers) or in the Others navigation tree in Designer Navigator (Global Markers). The project markers can be used only on objects of the project, and global markers can be used in all models of the repository.

A marker can be set on an object and can contain a value that will appear in the tooltip of the marker icon in the tree view.

More

The Markers tab lists the markers set on this object.

Properties	Description
Group	Name of the marker's group. See Marker Group Editor .
Marker	Marker assigned to the object.
Value	Marker value specified for the object.

6.34.5 Memo Tab

Use to attach a descriptive memo to a Project or Model element to reflect your project's methodology or to help with development.

Almost all Project and Model elements can have descriptive memos attached to them.

A memo is an unlimited amount of text attached to virtually any object, visible on its Memo tab. When an object has a memo attached, the Memo icon appears next to it in the Projects navigation tree.

6.34.6 Version Tab

Use to view and manage versions of your object.

A version is a backup copy of an object; it is checked in at a given time and may be restored later. Versions are saved in the master repository.

The version management system allows flags on developed objects (such as projects, models, and so forth) to be set automatically to indicate their status (such as new or modified). The version management system also allows these objects to be backed up as stable checkpoints, and later restored from these checkpoints. These checkpoints are created for individual objects in the form of versions.

Creation and Update Information

Properties	Description
Created By	User who created the object.
Created On	Date and time when the object was created.
Updated By	User who updated the object.
Updated On	Date and time when the object was updated.

Lock Information

Properties	Description
Lock	Name of user holding a lock on this object. This is empty if the object is not locked.

Information

Properties	Description
Internal ID	ID used for internal and legacy purposes. The Internal ID is unique within a repository.
Global ID	Global ID of the object. The global ID is unique for the object type across repositories.

Versions

This table lists the version number, the creation date of the version, the name of the user who created the version, and a version description.

You can perform the following actions by clicking the corresponding button:

- Create a new version
- Delete a version
- Restore a version
- Export a version

More

Properties	Description
Version	Version of mapping.
Date	Date when the version was created.
User	User who created the version.
Description	Description of version.

6.34.7 Privileges Tab

Use to view the authorized methods for a given user on this specific object instance.

An object instance (instance) is attached to an object type (an object). For example, the project `MY_PROJ_1` is an instance of the project object type (or object). Similarly, another instance of a project object type is `YOUR_PROJ_2`.

Privileges are set up and managed in Security Navigator. The most common actions in Security Navigator are:

- Creating and editing users and profile
- Assigning user rights for methods on ODI objects (such as data server and datatypes)
- Fine-tuning user rights on the object instances

More

Properties	Description
Method	Name of the object method. A method is a type of action that can be performed on an object, for example edit or delete. Each object has a series of methods that are specific to it. The concept of methods in Oracle Data Integrator is similar to the concept in Object-Oriented Programming.
User	Oracle Data Integrator user corresponding to the login name used for connecting to a repository. A user has profile privileges, privileges on objects, and privileges on instances.

6.34.8 Flexfields Tab

Use to customize the properties of Oracle Data Integrator objects.

Flexfields are user-defined fields that are set for each object instance through the Flexfield tab of the object editor, and their values are defined for each object instance through the Flexfield tab of the object's instance window. Flexfield values can be used through Oracle Data Integrator substitution methods.

Note: Flexfields exist only for certain object types. Objects that do not have a flexfield tab do not support them.

Properties	Description
Name	Label of the flexfield as it appears in the Flexfield tab of the object editor.
Code	Code of the flexfield.
Technology	Technology for which the flexfield is activated. For instance, a flexfield on a Procedure appears only for Procedures using this technology.
Type	Type of the flexfield (String or Numeric).
Default	Default value of the flexfield.

Related Topics

"Using Flexfields" in the "Introduction to the Substitution API" chapter in *Oracle Fusion Middleware Developing Knowledge Modules with Oracle Data Integrator*

"Navigating the Mapping Editor" in the "Creating and Using Mappings" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.35 Mapping Editor Concepts

This section contains information about general concepts applicable to the Mapping Editor. It contains the following topics:

- [Component Styles](#)
- [Component Attributes](#)

6.35.1 Component Styles

Mapping components in the Logical Diagram flow have two different styles that affect how attributes are represented and can be mapped:

- [Projector Components](#)
- [Selector Components](#)

Projector Components

Projector components define their own attributes. Attributes from preceding components are mapped through expressions to the projector's attributes. A projector hides attributes from preceding components; all succeeding components can only use the attributes from the projector.

More

Built-in projector components:

- [Dataset Component](#)
- [Datastore Component](#)
- [Flatten Component](#)
- [Set Component](#)
- [Reusable Mapping Component](#)
- [Aggregate Component](#)
- [Distinct Component](#)
- [Pivot Component](#)
- [Unpivot Component](#)
- [Subquery Filter Component](#)
- [Table Function Component](#)

Selector Components

Selector components reuse attributes from preceding components. Join and Lookup selectors combine attributes from the preceding components. For example, a Filter component following a datastore component reuses all attributes from the datastore component. As a consequence, with the exception of the Expression component, selector components do not display their own attributes in the diagram and as part of the properties; they are displayed as a round shape. When mapping to a component succeeding the selector component, the attribute can be directly dragged from the source, across a chain of connected selector components, to the target or next projector component.

More

Built-in selector components:

- [Expression Component](#)
- [Filter Component](#)
- [Join Component](#)
- [Lookup Component](#)
- [Sort Component](#)
- [Split Component](#)

6.35.2 Component Attributes

An attribute is a data field or column within a component. Depending on the type of component an attribute might have different properties.

General

Properties	Description
Name	Name of the attribute.
Description	Detailed description of the attribute.
Data type	SQL datatype of the attribute.
Bound Object	Datastore component only. Name of the bound base datastore attribute.

Target

The target tab is only shown for components accepting expressions, such as aggregate, distinct, expression, output signature, set, reusable mapping, or target datastore.

Properties	Description
Expression	SQL expression to map to the attribute. This field provides an Expression Editor for graphical editing. For a Set component the Expression property is a table with Expression and Execute on Hint for each Connection Point.
Execute on hint	Hint on which execution unit the expression is evaluated. Possible values: No Hint (default), Source, Staging, Target.
Fixed Execution Location	Specifies the actual logical schema that an expression shall be executed. Overrides Execute on hint for the expression of a target column.
Technical Description	Datastore component only Provides a version of the join with all attribute names replaced by their descriptions. This field cannot be edited.
Active	Datastore component only Indicates that the expression must be considered when executing the mapping.
Key	Datastore component only Shows that the attribute participates in the update key for the mapping.
Insert	Datastore component only Shows that this mapping attribute participates in the insertion operations in the target table.
Update	Datastore component only Shows that this mapping attribute participates in the update operations in the target table.
Check Not Null (Flow control only)	Datastore component only Shows that during the flow control, all the "not null" on this attribute must be verified.

Properties	Description
UD 1 – UD 10	Datastore component only Shows that this attribute participates in the operations linked to User Defined 1 to 10. See the documentation of the KM in use for more information.

Format

Properties	Description
Size	Logical length of the attribute, as it is functionally seen by the user. For a number, this is the total number of digits (Precision).
Scale	Number of digits contained in the decimal part of a numerical data type.

Related Topics

"Defining New Attributes" in the "Creating and Using Mappings" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.36 Mapping Editor Validation Results Tab

Use to view the issues that resulted from validating a mapping object.

Properties	Description
Object	Component of the mapping that failed the validation.
...	Shows a box with the issue details.
Severity	Shows an Error or Warning, depending on the issue.
Issue	Description of the issue.

Related Topics

"Running Mappings" in the "Creating and Using Mappings" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.37 Mapping Editor Logical Tab

Use this to define the logical mapping rules from source to target datastores.

More

The Logical tab consists of a logical diagram which contains components that are linked through connectors with arrow heads pointing the direction of the flow from source to target. Each component has left and right connector ports to anchor incoming and outgoing connectors.

In order to create a Logical Diagram perform the following steps:

1. Drag and drop datastores or reusable mappings from the designer navigator into the source or target areas. Source datastores and reusable mappings can also be organized within a [Dataset Component](#).
2. Drag and drop components such as joins, filters, or others from the component palette into the source area to define rules for the mapping between source and

target. If datasets are used, join, lookup, or filter, rules can be defined by dragging attributes between datastores and onto the dataset background.

3. Create connectors between components by dragging from the originating connector port to the destination connector port. Connectors can also be implicitly created by dragging attributes between components. When creating a connector between two ports, an [Attribute Matching Dialog](#) can be shown to automatically map attributes based on name or position.
4. Set component properties, expressions, and conditions by selecting components and editing the properties in the Property Inspector.

Toolbar

Name	Description
Perform Layout	Orders the mapping components in a default layout.
View as Compact	Changes all selected components to a compact view as icons.
View as Expanded	Changes all selected components to an expanded view.
Validate the Mapping	Checks the mapping for correctness; any issues that are found are listed in the Validation Window. Mapping Editor Validation Results Tab

Properties

Properties	Description
Name	Detailed description of the mapping.
Description	Name of the mapping.
Target Load Order	List of targets that are loaded in a defined order. Uses the Target Load Order Dialog to select and order targets.
Staging Location Hint	Specifies which logical schema is used as a staging location. If left as the default <unbound>, the logical schema of the target is used.

6.37.1 Component Palette

The component palette allows you to pick components and drag and drop them into a logical diagram (see: [Mapping Editor Logical Tab](#)) of a mapping or reusable mapping.

The component palette has the following components:

- [Aggregate Component](#)
- [Dataset Component](#)
- [Distinct Component](#)
- [Expression Component](#)
- [Filter Component](#)
- [Flatten Component](#)
- [Jagged Component](#)
- [Join Component](#)
- [Lookup Component](#)
- [Pivot Component](#)

- [Set Component](#)
- [Sort Component](#)
- [Split Component](#)
- [Subquery Filter Component](#)
- [Table Function Component](#)
- [Unpivot Component](#)
- [Input and Output Signature Components](#) (Reusable Mapping only)

The following options are available on the component palette:

- **All Pages Option:** Shows components from all Component Palette pages
- **Logical Option:** Shows components from the Mapping Logical view
- **Reusable:** Shows components that can be used in a Reusable Mapping Logical view. Only visible for a [Reusable Mapping Editor](#).
- **My Components Option:** Shows a subset of components based on prior use and selection as a favorite
 - **Favorites:** Shows components that were selected as favorite. Favorites can be added to this list by right clicking on a component and choosing the "Add to Favorites" menu item.
 - **Recently Used:** Shows components that were recently used in mappings

Note: Because of the new custom component extensibility in ODI there could be other components in the palette.

6.37.1.1 11g Compatible Dataset Component

An 11g compatible dataset component is special case of the [Dataset Component](#).

A dataset can be converted into equivalent flow components by using the command **Convert to Flow** in the dataset component context menu.

11g-compatible datasets are only present in 11g-compatible mappings. 11g compatible mappings maintain compatibility to the 11g SDK, so that existing programs using the ODI SDK continue working in 12c.

- Upgrade from an 11g work repository, and in the Upgrade Assistant, deselect **Don't Maintain ODI 11g SDK Compatibility**.
- Through commands of the 11g-compatible SDK.

11g-compatible maps are presented under the mapping node in the Designer Navigator. 11g-compatible mappings cannot be created or modified in the Studio; a mapping opened in a mapping editor is read-only. These mappings are distinguished from 12c mappings by an 11g interface icon.

The 11g-compatible dataset uses the same properties as a standard dataset but it additionally has a list of [Component Attributes](#) with expressions and execute-on hints.

6.37.1.2 Aggregate Component

Use to group and aggregate attributes using aggregate functions, such as average, count, maximum, sum, and so on. ODI automatically selects attributes without aggregation functions to be used as group-by attributes. You can override this by using the **Is Group By Column** and **Manual Group By Clause** properties.

More

AttributesSee: [Component Attributes](#)**General**

Properties	Description
Name	Name of the Aggregate component.
Description	Detailed description of the Aggregate component.
HAVING	Having clause to be added to the generated SQL. This is optional.
Manual Group by Clause	Group by clause to be used in the generated SQL, instead of the columns automatically selected by the Is Group By property.
Execute on Hint	Preferred execution unit that performs the transformation. This is a hint that can be overridden on the physical view. This execution unit must have sufficient capacity (functions, SQL, and so forth) to effect these transformations. <ul style="list-style-type: none"> ▪ No Hint (default): The system determines the optimal location for the execution. ▪ Source: This is the data server that hosts the source datastore or datastores. For a source execution of a transformation that could be executed on several sources, a drop-down list allows you to choose the source on which the mapping is to be executed. For example, if, with two source tables on two Oracle servers, you enter the <code>SYSDATE</code> mapping, with execution on the source, the choice allows you to state if you want the date of one Oracle server or the other. ▪ Staging Area: This is the execution unit selected on the Definition tab on the mapping. By default, this is the Target unit. ▪ Target: This is the data server that hosts the target datastore. Mappings on the target cannot reference source attributes, nor can they contain aggregate functions.
Component Type	Type of the component, set to <code>AGGREGATE</code> .

Connection PointsSee: [Connector Points](#).**6.37.1.3 Dataset Component**

A dataset component is a selector component (see: [Selector Components](#)) that allows the user to group multiple data sources and join them through relationship joins.

More

A dataset contains the following components:

- [Datastores](#)
- [Joins](#)
- [Lookups](#)
- [Filters](#)
- [Reusable Mappings](#): Only reusable mappings with no input signature and one output signature are allowed.

Joins and lookups are created by dragging an attribute from one datastore to another inside the dataset. A dialog box is shown to select if the relationship will be a join or lookup. A filter is created by dragging a datastore or reusable mapping attribute onto

the dataset background. Joins, lookups, and filters cannot be dragged from the component palette into the dataset.

General

Properties	Description
Name	Name of the dataset.
Description	Description of the dataset.
Component Type	Type of the component, set to DATASET

Datastores

See the properties in [Datastore Component](#).

Joins

See the properties in [Join Component](#).

Filters

See the properties in [Filter Component](#).

Lookups

See the properties in [Lookup Component](#).

Reusable Mappings

See the properties in [Reusable Mapping Component](#).

Connection Points

See: [Connector Points](#).

6.37.1.4 Distinct Component

A distinct component is a projector component that projects a subset of attributes in the flow. The values of each row have to be unique; the behavior follows the rules of the SQL DISTINCT clause.

More

Attributes

See: [Component Attributes](#)

General

Properties	Description
Name	Name of the Distinct component.
Description	Detailed description of the Distinct component.
Execute on Hint	Hint on which execution unit the distinct clause will be evaluated. Possible Values: No Hint (default), Source, Staging, Target.;
Component Type	Type of the component, set to DISTINCT

Connection PointsSee: [Connector Points](#).**6.37.1.5 Expression Component**

An expression component is a selector component (see: [Selector Components](#)) that inherits attributes from a preceding component in the flow and adds additional reusable attributes. An expression can be used to define a number of reusable expressions within a single mapping. Attributes can be renamed and transformed from source attributes using SQL expressions. The behavior follows the rules of the SQL `SELECT` clause.

More

AttributesSee: [Component Attributes](#)**General**

Properties	Description
Name	Name of the expression component.
Description	Detailed description of the expression component.
Execute on Hint	Hint on which execution unit the distinct clause is evaluated. Possible Values: No Hint (default), Source, Staging, Target.
Component Type	Type of the component, set to <code>EXPRESSION</code> .

Connection PointsSee: [Connector Points](#).**6.37.1.6 Filter Component**

A filter is a selector component (see: [Selector Components](#)) that can select a subset of data based on a filter condition. The behavior follows the rules of the SQL `WHERE` clause.

Filters can be located in a dataset or directly in a mapping as a flow component. When used in a dataset, a filter is connected to one datastore or reusable mapping to filter all projections of this component out of the dataset.

More

Condition

Properties	Description
Filter Condition	SQL Condition to filter the data passed through the component.
Technical Description	Provides a version of the filter with all attribute names replaced by their descriptions. This field cannot be edited.
Execute on Hint	Hint on which execution unit the filter will be evaluated. Possible Values: No Hint (default), Source, Staging, Target.

General

Properties	Description
Name	Name of the filter component.
Description	Detailed description of the filter component.
Active Indicator	Indicates that the filter must be considered when executing the mapping.
Component Type	Type of the component, set to <code>FILTER</code> .

Connection Points

See: [Connector Points](#).

Note: Connection Points are not shown for components within a dataset.

6.37.1.7 Flatten Component

The flatten component is a Projector component (see: [Projector Components](#)) that can process input data with complex structure and produce a flattened representation of the same data using standard datatypes.

More

Attributes

See [Component Attributes](#) for information on attributes. In addition to the attributes described in the Component Attributes section, the following attribute is specific to the flatten component.

Properties	Description
Tag	Used if the component attribute has a different name from the one used to define the physical data.

General

Properties	Description
Name	Name of the component.
Description	Detailed description of the flatten component.
Component Type	Type of component, set to <code>Flatten</code> .
Complex Type Attribute	Input attribute that contains the complex data, which must be flattened.
Include Nulls	Select to generate rows with no data for the attributes that are defined as <code>NULL</code> .
Execute on Hint	Hint on which execution unit the expression is evaluated. Possible values are: <code>No Hint</code> (default), <code>Source</code> , <code>Staging</code> , <code>Target</code> .

Connection Points

See: [Connector Points](#).

6.37.1.8 Input and Output Signature Components

Input Signature and Output Signature are Projector components (see: [Projector Components](#)) that define the input and output of reusable mappings. A reusable mapping can have zero to many input and output signatures.

More

Attributes

Attributes are defined by the user to define the data structure of the reusable mapping. Input and output signatures have different sets of attribute properties. See: [Component Attributes](#) for details.

General

Properties	Description
Name	Name of the signature component.
Description	Detailed description of the signature component.
Component Type	Type of the component, set to INPUTSIGNATURE or OUTPUTSIGNATURE.

Connection Points

See: [Connector Points](#).

6.37.1.9 Jagged Component

A jagged component processes unstructured data. Source data from sources such as social media or e-commerce businesses are represented in a key-value free format. Using the jagged component, this data can be transformed into structured entities that can be loaded into database tables.

More

Attributes and Attribute Groups

The jagged component has one input group and multiple output groups based on the configuration of the component. Using the jagged component, you can group attributes based on dimensionality. These attribute groups are exposed as output groups. You can specify one or more attribute groups and assign named attributes to these groups. Attribute grouping allows you to specify data or dimensional sectioning. The key, value datasets can contain multiple logical datasets. These can be dimensional data, detail data, or hierarchical data. Cube data require grouping dimensional attributes and complex data require attribute grouping to specify complex type definitions.

See [Component Attributes](#) for information on attributes. In addition to the attributes described in the Component Attributes section, the following attributes are specific to the jagged component.

Properties	Description
Default	Use to define a default value for the missing keys in an incoming dataset.

Properties	Description
Record ID	Select to allow the use of running sequences to delineate rows in the datasets. An incoming dataset is usually a name and a value pair but in some cases, running row ID sequences are also generated to help in the pivoting of the dataset. The Record ID attribute stores the input data set sequence value.
Unique Index	Select to identify the attribute as the primary key for the attribute grouping to which it belongs. This can be used to validate or consolidate the incoming datasets and is used in combination with the Consolidate Data property defined for the group.
Required	Select to identify the attribute as required for generating an output record. Note: If the required attribute is not part of the incoming dataset, then no output record will be generated.

General

Properties	Description
Name	Name of the component.
Description	Detailed description of the component.
Component Type	Type of the component, set to Jagged.
End of Data Indicator	Use to indicate the attribute that will mark the beginning of a new record.

Connection Points

See [Connector Points](#) for information on connector points. In addition to the connector points described in the Connection Points section, the following attribute group properties are populated under output connector points.

Properties	Description
Serialization file	Name of the temporary file used by the jagged component to process incoming records. The jagged component partitions the incoming data to the relevant attribute groups and to the corresponding file. Jagged uses default names if the serialization file names are not specified. Note: The default attribute group others has a predefined temporary file name.
Consolidate data	Select to overwrite or update the existing row with the incoming row if there is key collision. This allows the data to be consolidated before it is sent to the downstream component for further processing. Note: For attribute groups that have the Consolidate data option selected and at least one primary key attribute, the jagged component retains the last updated row.

6.37.1.10 Join Component

A join is a selector component (see [Selector Components](#)) that creates a join between multiple flows. The attributes of all flows are combined as the attributes of the join component.

A join can be located in a dataset or directly in a mapping as a flow component. A join combines data from two or more components, datastores, datasets, or reusable mappings.

When used in a dataset, a join combines the data of the datastores using the selected join type. Joins in a dataset can be created by dragging an attribute from one datastore to another.

A join used as a flow component can join two or more datastores. A join condition can be formed by dragging attributes from two components successively onto a join component in the mapping editor; the join condition is an equi-join between the two attributes.

More

Condition

Properties	Description
Join Condition	A join condition is usually expressed in SQL language. For sequenced joins (ISO), this is the text of the <code>ON</code> clause. For example: <code>REG.REGION_ID=CIT.REGION_ID</code>
Join Type	The following join types can be selected: <ul style="list-style-type: none"> ■ Inner Join (No check boxes selected) ■ Left Outer Join (Left datastore check box selected) ■ Right Outer Join (Right datastore check box selected) ■ Full Outer Join (Both datastore check boxes selected) ■ Cross Join (Cross check box selected) ■ Natural Join (Natural check box selected)
Technical Description	Provides a version of the join with all attribute names replaced by their descriptions. This field cannot be edited.
Generate ANSI Syntax	Only available for join components inside datasets. If set, ODI generates SQL in ANSI syntax to implement the join logic. In ANSI syntax, changing the order of joins may change the result, especially when outer joins are involved.
Join Order	Only available for join components inside datasets. If true, join order is determined by the numerical field, User Defined. If false, ODI will choose an arbitrary join order when converting the dataset logic to flow based logic. If Generate ANSI Syntax is set, it is strongly recommended that you carefully review the join orders. When the Generate ANSI Syntax is checked, ODI will automatically set a unique user-defined value for Join Order, if it has not yet been set.
User Defined	Join order number to determine the order of multiple joins and lookups in a dataset. A join with a lower order number is resolved first. The number must be a positive integer and is only considered if the Join Order check box is checked.
Execute on Hint	Hint on which execution unit the join will be evaluated. Possible Values: No Hint (default), Source, Staging, Target. If Source is selected, all datastores in the join need to be in the same data server, and the server must have join capabilities.

General

Properties	Description
Name	Name of the join component.
Description	Detailed description of the join component.

Properties	Description
Active Indicator	This check box indicates that the join must be considered when executing the mapping.
Component Type	Type of component, set to JOIN.
Join Source Order	<p>Only available for join components inside datasets.</p> <p>This field cannot be edited.</p> <p>Displays a comma-separated list of two input datastore components for Join Source Order.</p> <p>Note: All joins in a dataset are binary joins.</p> <p>The first of this list is taken as the "left" side of the join, and the second of the list is taken as the "right" side of the join. This order of the datasource to a join is important in unambiguously defining outer joins.</p>
Derived from	<p>If the join is a derived join, shows name of the Join component this join is derived from.</p> <p>A derived join is created if a join condition of another join involves more than two datastores. In this case the join is deconstructed into multiple conditions with two datastores; all additional conditions create derived joins.</p> <p>Only available for join components inside datasets.</p>

Connection Points

See: [Connector Points](#).

Note: Connection Points are not shown for components within a dataset.

6.37.1.11 Lookup Component

A lookup is a selector component (see [Selector Components](#)) that returns data from a lookup flow being given a value from a driving flow. The attributes of both flows are combined, similarly to a join component. A lookup can be implemented in generated SQL either through a Left Outer Join or a nested Select statement.

Lookups can be located in a dataset or directly in a mapping as a flow component.

When used in a dataset, a Lookup is connected to two datastores or reusable mappings combining the data of the datastores using the selected lookup type. Lookups in a dataset can be created by dragging an attribute from the driving datastore to the lookup datastore. The lookup datastore can be identified in the dataset by an arrowhead pointing to it.

Lookups used as flow components can join two flows. A lookup condition can be created by dragging an attribute from the driving flow and then the lookup flow onto the lookup component; the lookup condition is an equi-join between the two attributes.

More

Condition

Properties	Description
Lookup Condition	A lookup expression is usually expressed in SQL language. For example: REG.REGION_ID=CIT.REGION_ID
Technical Description	Formal syntax of expression used for lookup. For example: [SRC_CUSTOMERS (SRC_CUSTOMERS)].[ADDRESS_KEY] = [SAMP_ADDRESSES_D (SAMP_ADDRESSES_D)].[ADDRESS_KEY] This field cannot be edited.
Execute on Hint	Hint on which execution unit the lookup is evaluated. Possible Values: No Hint (default), Source, Staging, Target. If Source is selected, all datastores in the lookup need to be in the same data server, and the server must have join capabilities.

Match Row Rules

Properties	Description
Multiple Match Rows	Select from the following to specify the action to perform when multiple rows are returned by the lookup operation: <ul style="list-style-type: none"> ■ All Rows (number of result rows may differ from the number of input rows): This option indicates that when the lookup operation returns multiple rows, all the rows should be returned as the lookup result. ■ Error: multiple rows cause mapping to fail: This option indicates that when the lookup operation returns multiple rows, the mapping execution fails. ■ Select any single row: This option indicates that when the lookup operation returns multiple rows, any one row from the returned rows must be selected as the lookup result. ■ Select first single row: This option indicates that when the lookup operation returns multiple rows, the first row from the returned rows must be selected as the lookup result. ■ Select last single row: This option indicates that when the lookup operation returns multiple rows, the last row from the returned rows must be selected as the lookup result. ■ Select nth single row: This option indicates that when the lookup operation returns multiple rows, the nth row from the result rows must be selected as the lookup result. <p>Note: For existing mappings in Oracle Data Integrator 12.1.2, the "Deprecated - All Rows (number of result rows may differ from the number of input rows)" and "Deprecated - Error: multiple rows cause mapping to fail" options indicate the same functionality as the "All Rows (number of result rows may differ from the number of input rows)" and "Error: multiple rows cause mapping to fail" options respectively.</p>

Properties	Description
No-Match Rows	<p>Select one of the following options to specify the action to perform when no rows are returned by the lookup operation:</p> <p>Return no row: This option does not return any row when no row in the lookup results satisfies the lookup condition.</p> <p>Return a row with the following default values: This option returns a row that contains default values when no row in the lookup results satisfies the lookup condition. Use the Lookup Attributes Default Value & Order By to specify the default values for each lookup attribute.</p>
Nth Row Number	<p>Specify the value of the nth row. The value of the nth row should be equal to or greater than 1.</p> <p>This field is displayed when the Select nth single row option is selected in the Multiple Match Rows drop-down list.</p>
Lookup Attributes Default Value & Order By	<p>Specify how the result set that contains multiple rows should be ordered. Ensure that the attributes are listed in the same order (from top to bottom) in which you want the result set to be ordered. The Default Value column is displayed if Return a row with the following default values is selected in No-Match Rows.</p>

General

Properties	Description
Name	Name of the lookup component.
Description	Detailed description of the lookup component.
Component Type	Type of component, set to LOOKUP.
Driver Source	Name of the driver source component.
Lookup Source	Name of the lookup source component.

Connection Points

See: [Connector Points](#).

Note: Connection Points are not shown for components within a dataset.

6.37.1.12 Pivot Component

A pivot component is a projector component (see [Projector Components](#)) that transforms data that is contained in multiple input rows into a single output row. The pivot component extracts data from a source once and produces one row from a set of source rows that are grouped by attributes in the source data. The pivot component can be placed anywhere in the data flow of a mapping.

More

Attributes

See [Component Attributes](#) for information on attributes. In addition to the attributes described in the Component Attributes section, the following attribute is specific to the pivot component.

Properties	Description
Matching Row	Value of the row locator, which indicates the row within the pivot group from which the attribute obtains its data. You can modify the Row Locator values to define the available values for the Matching Row attribute.

General

Properties	Description
Name	Name of the pivot component.
Description	Detailed description of the pivot component.
Execute on Hint	Hint on which execution unit the join is evaluated. Possible Values: No Hint (default), Source, Staging, Target.
Aggregate Function	Name of the aggregate function used in a pivot operation. For example, MIN (default), SUM, AVG.
Component Type	Type of component, set to PIVOT.

Row Locator

When using the pivot component, multiple input rows are transformed into a single row based on the row locator. The row locator is an attribute that must be selected from the source to correspond with the set of defined output attributes. The row locator Values specify the possible values of the row locator. It is necessary to specify a row locator to perform the pivot operation.

Connector Points

See: [Connector Points](#).

6.37.1.13 Reusable Mapping Component

A reusable mapping component is a projector component (see [Projector Components](#)) that transforms zero or more incoming flows to zero or more outgoing flows, based on a nested reusable mapping (see [Reusable Mapping Editor](#)). A reusable mapping allows the definition of flow logic and expressions that can be reused in multiple mappings.

Select **View As: Expanded** in the context menu of the Reusable Mapping component, to display the embedded reusable mapping.

More

Reusable Mapping Signatures

This table lists the attributes of input and output signatures (see: [Input and Output Signature Components](#)) defined in the reusable mapping. The attribute properties are read-only and can only be changed in the reusable mapping itself. Input attributes allow the setting of an expression to map from attributes of the incoming flow, as well as an Execute on Hint for executing the expression.

General

Properties	Description
Name	Name of the reusable mapping component.
Description	Detailed description of the reusable mapping component.

Properties	Description
Component Type	Type of the component, set to REUSABLEMAPPING.
Reusable Mapping	Name of the reusable mapping used in this component.
Subselect enabled	<p>If set to true, the SQL statement generated from the reusable mapping is placed into a nested select statement.</p> <p>The Subselect Enabled check box is provided for backward compatibility. It's only enabled if the Reusable Mapping has no Input Signatures and exactly one Output Signature, which is the equivalent of the old Temporary Interface.</p> <p>When this check box is unchecked, the code generator will decide if the reusable mapping expression is "flattened" into the generated SQL or if a subselect is still used.</p>

6.37.1.14 Set Component

A set component is a projector component (see: [Projector Components](#)) that combines multiple input flows into one using set operation such as UNION, INTERSECT, EXCEPT, MINUS, and others. The behavior reflects the SQL operators.

Additional input flows can be added to the set component by connecting new flows to it. The amount of input flows is shown in the list of Input Connection Points in the Operators tab. If an input flow is removed, the input connection point needs to be removed as well.

More

Attributes

Properties	Description
Target	Name of the attribute.
Expression for: INPUT<#>	Expression originating from the connector point called <INPUT_NAME>.

General

Properties	Description
Name	Name of the set component.
Description	Detailed description of the set component.
Execute on Hint	<p>Hint on which execution unit the join is evaluated. Possible Values: No Hint (default), Source, Staging, Target.</p> <p>If Source is selected, all datastores in the join need to be in the same data server, and the server must have join capabilities.</p>
Component Type	Type of component, set to SET.

Operators Tab

The operators tab contains the standard [Connector Points](#) properties and a set operator for the second and following input connection points. The operator can be one of the following SQL-defined set operations:

- EXCEPT
- EXCEPT ALL

- INTERSECT
- INTERSECT ALL
- INTERSECT DISTINCT
- MINUS
- MINUS ALL
- MINUS DISTINCT
- UNION
- UNION ALL
- UNION DISTINCT

6.37.1.15 Sort Component

A sort is a selector component (see [Selector Components](#)) that sorts the incoming flow based on a list of attributes. The functionality is equivalent to the SQL `ORDER BY` clause.

Sort condition fields can be added by dragging source attributes onto the sort component.

More

Condition

Properties	Description
Sorter Condition	Comma-separated list of attributes to be used for sorting.
Execute on Hint	Hint on which execution unit the join is evaluated. Possible Values: No Hint (default), Source, Staging, Target. If Source is selected, all datastores in the join must be in the same data server, and the server must have join capabilities.

General

Properties	Description
Name	Name of the sort component.
Description	Detailed description of the sort component.
Component Type	Type of component, set to <code>SORT</code> .

Connection Points

See: [Connector Points](#).

6.37.1.16 Split Component

A Split is a Selector component (see: [Selector Components](#)) that divides a flow into two or more flows based on if-then-else conditions. If a flow is divided unconditionally into multiple flows, no split component is necessary. It is sufficient to connect multiple flows to an outgoing connector port of any preceding component.

More

General

Properties	Description
Name	Name of the split component.
Description	Detailed description of the split component.
Execute on Hint	Hint on which execution unit the join is evaluated. Possible Values: No Hint (default), Source, Staging, Target. If Source is selected, all datastores in the join need to be in the same data server, and the server must have join capabilities.
Component Type	Type of the component, set to SPLIT.

Split Conditions

The Split Conditions tab contains the standard [Connector Points](#) properties as well as expression and remainder properties.

Properties	Description
Remainder	If checked, only select rows that do not satisfy any of the other conditions. Note: If the Remainder check box is checked and the split condition contains an attribute whose value is null, the corresponding rows are not moved to the target table.
Expression	Condition to filter rows for the given output connector point. Conditions do not need to be mutually exclusive; the same row can be passed to multiple flows. The expression cannot be empty.

6.37.1.17 Subquery Filter Component

A subquery filter component is a projector component (see [Projector Components](#)) that filters rows based on the results of a subquery.

More

Attributes

See [Component Attributes](#) for information on attributes. In addition to the attributes described in the Component Attributes section, the following attributes are specific to the subquery filter component.

Properties	Description
Expression for DRIVER_INPUT	SQL expression for the DRIVER_INPUT connector point.
Expression for SUBQUERY_FILTER_INPUT	SQL expression for the SUBQUERY_FILTER_INPUT connector point.

Conditions

The conditions that can be used to filter rows are EXISTS, NOT EXISTS, IN, NOT IN, =, >, <, >=, <=, !=, <>, and ^=.

Properties	Description
Subquery Filter Condition	A subquery filter condition is usually expressed in SQL language for the EXISTS or NOT EXISTS conditions.
Subquery Filter Input Role	Select from the EXISTS , NOT EXISTS , IN , NOT IN , = , > , < , >= , <= , != , <> , or ^= conditions.
Group Comparison Condition	A group comparison condition specifies a comparison with any or all members in a subquery. Group comparison condition can only be used with = , > , < , >= , <= , != , <> , ^= . Select from No condition , Any , Some , and All group conditions. No condition is the default value and signifies that no group comparison condition should not be used.

General

Properties	Description
Name	Name of the subquery filter component.
Description	Detailed description of the subquery filter component.
Component Type	Type of component, set to SUBQUERYFILTER.
Execute on Hint	Hint on which execution unit the expression is evaluated. Possible values are: No Hint (default), Source, Staging, Target.

Connector Points

A subquery filter component has two input connector points and one output connector point. The two input connector points are Driver Input connector point and Subquery Filter Input connector point. The Driver Input connector point is where the main datastore is set, which drives the whole query. The Subquery Filter Input connector point is where the datastore that is used in the subquery is set.

See: [Connector Points](#).

6.37.1.18 Table Function Component

A table function component is a projector component (see [Projector Components](#)) that represents a table function in a mapping. Table function components can be used to manipulate a set of input rows and return another set of output rows of the same or different cardinality. The set of output rows can be queried like a physical table. A table function component can be placed anywhere in a mapping, as a source, a target, or a data flow component.

More

Attributes

See: [Component Attributes](#)

General

Properties	Description
Name	Name of the table function component.
Description	Detailed description of the table function component.
Component Type	Type of component, set to TABLEFUNCTION.

Properties	Description
Function Name	Name of the function.
Execute on Hint	Hint on which execution unit the expression is evaluated. Possible values are: No Hint (default), Source, Staging, Target.

Connector Points

A table function component can have multiple input connector points and one output connector point. The input connector point attributes act as the input parameters for the table function, while the output connector point attributes are used to store the return values.

For each input connector, you can define the parameter type, REF_CURSOR or SCALAR, depending on the type of attributes the input connector point will hold.

See: [Connector Points](#).

6.37.1.19 Unpivot Component

An unpivot component is a projector component (see [Projector Components](#)) that transforms data that is contained across attributes into multiple rows.

The unpivot component does the reverse of what the pivot component does. Similar to the pivot component, an unpivot component can be placed anywhere in the flow of a mapping.

The unpivot component is specifically useful in situations when extracting data from non-relational data sources, such as a flat file, which contains data across attributes rather than rows.

More

General

Properties	Description
Name	Name of the unpivot component.
Component Type	Type of component, set to UNPIVOT.
Description	Detailed description of the unpivot component.
Execute on Hint	Hint on which execution unit the expression is evaluated. Possible values are: No Hint (default), Source, Staging, Target.
Row Locator	Name of the Row Locator attribute. Attributes in the unpivot component that do not have an expression can be row locator candidates.
INCLUDE NULLS	Select to generate rows with no data for the attributes that are defined as NULL.

Attributes

The row locator is an output attribute that corresponds to the repeated set of data from the source. Output attributes that are not assigned an expression are row locator candidates and are displayed in the Row Locator drop-down list. Once an attribute is selected as a row locator, the remaining output attributes are referred to as value locators. The row locator and value locators are displayed in the Unpivot Transforms table.

See: [Component Attributes](#)

Unpivot Transforms

In the Unpivot Transforms section, the transform rules for each output attribute can be added and the default values of the transform rules can be edited. The appropriate expressions to create the required logic can also be specified. Note that the value attribute expression can also be an arbitrary expression, for example, UNPIVOT_EMP_SALES.Q1_SALES + 100.

6.37.2 Datastore Component

A datastore component is a projector component (see [Projector Components](#)) that represents a source or target datastore based on a model. A datastore is a table-like data structure based on a physical storage mechanism such as a database, file, API, or other mechanism.

More

Datastores are added to a mapping by dragging and dropping them from the Model Navigator onto the logical mapping diagram. Datastores can be inside a dataset or in the source or target areas. Datastores can also be used as both target and source by connecting input and output connector ports to other components.

A datastore can have multiple output connectors to use the same data in multiple flows. Like other components, a datastore can also be used multiple times in a flow. Only one input connector is permitted; if multiple flows should be combined into one target datastore, a set component should be used.

A datastore is considered a source datastore if it has at least one output connector but no input connectors. When it has no output connectors, a datastore is considered a target datastore.

More

Attributes

Attributes are fixed and provided by the base datastore. Source and target datastores have different sets of attribute properties. See: [Component Attributes](#) for details.

General

Properties	Description
Name	Name of the datastore component.
Description	Detailed description of the Datastore component.
Partition/Sub-Partition	The partition or sub-partition of the datastore to use as a source of the mapping.
Datastore	Name of the base datastore.
Shortcut	Name of the shortcut if base model is a shortcut. Empty if no shortcut is used.
Logical Schema	Name of the logical schema specified for the datastore's model.
Component Context (Forced)	Context within which this mapping uses the datastore. Leave empty to use the execution context. For more details, see Context Editor .
Physical Schema	Physical Schema containing the datastore in the context.
Data Server	Data Server containing the datastore in the context

Properties	Description
Catalog	Catalog containing the datastore in the context.
Component Type	Type of component, set to <code>DATASTORE</code> .

Target

Properties	Description
Integration Type	<p>Type of integration strategy used by the Integration Knowledge Module:</p> <ul style="list-style-type: none"> ■ Control Append: Inserts rows to the target datastore. ■ Incremental Update: Inserts, changes, or deletes rows in the target datastore. ■ Slowly Changing Dimension: Adds data to the target as slowly changing dimension.
Update Key	<p>Specifies the primary or alternate key that is used when matching the source rows with the target datastore.</p> <p>This value is usually used when UPDATE DML is generated; for example in IKM SQL Incremental Update.</p>
Maximum Number of Errors Allowed	<p>Applies to errors detected while extracting data from source file datastores, and to errors detected by the flow control process. When this number or percentage of errors is exceeded, the mapping switches to an error state.</p> <p>If the Integration Errors as Percentage check box is unselected, the mapping switches to the error state when one of the following conditions is true:</p> <ul style="list-style-type: none"> ■ The number of errors detected while extracting data from one file exceeds the number of errors allowed. ■ The number of errors detected during the flow control exceeds the number of errors allowed. <p>If the Integration Errors as Percentage check box is selected, the mapping switches to the error state when one of the following conditions is true:</p> <ul style="list-style-type: none"> ■ The number of errors detected while extracting data from one file exceeds the percentage of errors allowed. The percentage is computed from the number of lines extracted from the file. ■ The number of errors detected during the flow control exceeds the percentage of errors allowed. This percentage is computed from the number of records integrated to the target datastore (Insert + Update). <p>Note: When this field is left empty, then an unlimited number of errors is allowed.</p> <p>Note: The total number of errors can exceed the number of rows in the mapping, as a line can infringe several constraints or contain several errors. In this case, the multiple errors add up.</p>
Integration Errors as Percentage	Causes the Maximum Number of Errors Allowed field to be interpreted as a percentage value.

Journalizing

Properties	Description
Journalized Data Filter	Filter to use for Journalized Data Only. The default filter checks for the subscriber name that will process the data; you should substitute the default <code>SUNOPSIS</code> with the actual subscriber name. Filter is ignored when Journalized Data Only is deselected.

Constraints

Properties	Description
Constraint Name	Name of constraint as defined in base datastore.
Constraint Type	Constraint type as defined in base datastore.
Value	Shows if the given constraint should be checked (<code>true/false</code>).

Connection Points

See: [Connector Points](#).

Note: Connection Points are not shown for components within a dataset.

6.37.3 Mapping Diagram

Use the mapping diagram to lay out a visual flow of data from source datastores, through intermediate components, and into target datastores. You can view the mapping diagram from the Logical or Physical tabs. The diagram shows either the logical or physical layout of your mapping. The logical tab of the mapping diagram may be referred to as the Logical Diagram, and the physical tab may be referred to as the Physical Diagram.

Right-click an object in the mapping diagram to access a context menu for that object.

Table 6–2 Mapping Diagram Toolbar

Name	Description
Zoom In	Use the zoom buttons and drop-down menu to zoom in and out of the mapping diagram
Zoom Out	
Zoom drop-down list	
Perform Layout	Automatically rearranges the components in the mapping diagram in a neat and organized way, with data flowing from the left to the right
View As Compact	Shows all components in the mapping diagram as small icons.
View As Expanded	Shows all projector components in the mapping diagram as boxes containing lists of their attributes.
Validate the Mapping	Performs a validation operation on the mapping. The Validation Results pane will list validation errors detected in the mapping.
Expand All Execution Units	Shows execution units as boxes with contained components displayed.

Table 6–2 (Cont.) Mapping Diagram Toolbar

Name	Description
Collapse All Execution Units	Shows execution units as compact icons.
Synchronize with Logical	Updates a physical diagram to incorporate changes made to the parent logical diagram.
Reset Physical Mapping Design	Resets the current physical mapping design to remove all changes made since it was generated by ODI from the logical diagram.

Related Topics

"Navigating the Mapping Editor" in the "Creating and Using Mappings" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

"Using Mapping Components" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.38 Mapping Editor Physical Tab

Use this to view the distribution of execution among different execution units that represent physical servers. You can use the diagram to move components between execution units or onto the diagram background, which would create a separate execution unit.

A physical diagram is stored as a Physical Mapping Design. A mapping can have multiple physical mapping designs; they are listed in tabs under the diagram. By having multiple physical mapping designs the user can create different execution strategies for the same mapping. In order to create a physical mapping design, click the right-most tab at the bottom of the physical diagram. To delete physical mapping designs, use the context menu on the Physical Mapping Design tabs.

More

6.38.1 Properties

Shows general properties for the physical design and execution of a mapping.

6.38.1.1 Physical Mapping Design Tab

Properties	Description
Description	Detailed description of the Physical Mapping Design.
Use Unique Temporary Object Names	If selected, generates unique staging table names to avoid collisions between concurrent scenario executions of the same mapping.
Is Frozen	If selected, prevents recalculation of the execution units based on the logical design. This ensures that the execution units remain the same even if the staging location is changed.
Name	Name of the Physical Mapping Design.
Optimization Context	Context used at design time to identify physical details of participating datastores. The execution context is selected separately at runtime.

Properties	Description
Preset Staging Location	<p>When left blank, target execution unit is used as staging location.</p> <p>As a general rule, if the target data server has join and transformation capabilities (classic SQL capabilities), it is advised to use the engine of this data server for the transformations.</p> <p>Temporary objects are created in a schema on this server called the Staging Area. If the target server does not have adequate SQL capabilities (for example if the target is an MOM or a file), you can choose another transformation engine, which can be another data server, or the internal engine working in memory.</p> <p>Caution: Only by choosing a transformation engine on the target can the data flow consistency be checked automatically.</p>
Remove Temporary Objects on Error	Removes all temporary objects including staging tables if a session execution fails.

Note: To add a second Physical Mapping Design, click on the "new" tab to the right of existing Physical Mapping Design tabs.

6.38.2 Source Group and Target Group Properties

Source Groups and Target Groups contain one or more execution units.

More

Properties	Description
Name	Name of the execution group.
Description	Detailed description of the execution group.

6.38.2.1 Execution Unit

An Execution Unit represents a physical server or process that can execute parts of a mapping.

More

Additional execution units can be created by dragging and dropping components from an existing execution group onto the physical diagram background.

Execution Unit

Properties	Description
Name	Name of the execution unit.
Description	Detailed description of the execution unit.
Location	Logical Schema associated with the execution unit.
Technology	Technology associated with the execution unit.

Expressions

Shows expressions that are either owned by components in the execution unit, or are intended to be executed on the unit.

Properties	Description
Name	Name of the attribute owning the expression.
Expression	Text of the expression available for pushing.
Owner	Name of the logical component owning the expression.
Execute on	Execution unit where the expression will be executed. <here> indicates the expression will be executed in the currently selected execution unit. Select a different execution unit to move the expression.

6.38.3 Access Point Properties

In the physical diagram, an Access Point is a physical component that controls the loading of a component that is remote to the target. In a target execution group, whenever the flow of data goes from one execution unit to another there is an access point (shown with a round icon). Loading Knowledge Modules (LKMs) control how data is transferred from one execution unit to another.

General

Properties	Description
Name	Name of the access point.
Description	Detailed description of the access point.
Node Component Type	Type of the component, set to AP Node.
Logical Component	Name of the logical component linked to this physical component.
Logical Connection Point	Name of the logical connection point linked to this physical component. This applies when the access point node has a downstream component with multiple input connector points.
Fixed Execution Unit	Set to true if the physical node is moved explicitly by the user to the parent execution unit. ODI will try to preserve the execution unit and the physical node in that execution unit when synchronizing between the logical and physical mapping.
Fixed by Hint	Set to true if the physical node is moved to the parent execution unit by the Execute on hint property, specified in the logical mapping. ODI may not preserve the execution unit where the current physical node resides when synchronizing between logical and physical mapping, but considers the position of the physical node based on execute on hint.

Attributes

The attributes that need to be made available from the upstream execution unit to this execution unit. If the execution unit containing the access point node is Oracle, ODI creates a staging table with a prefix C\$, and a set of columns to load the necessary data from the upstream execution unit.

Properties	Description
Name	Name of the attribute.
Description	Description of the attribute.

Properties	Description
Source Attribute	Name of attribute bound from source component. The logical map attribute that is used to derive this physical attribute (thus it is called source attribute of the physical attribute). In other words, the data of the source logical map attribute in the upstream execution unit is staged in this C\$ table column.
DDL Datatype	Datatype string that will be used to create the staging table.
Datatype	Logical datatype of the source attribute.
Size	Size of the logical attribute, for numerical data types.
Scale	Scale of the logical source attribute, for numerical types.

Loading Knowledge Module

Properties	Description
Loading Knowledge Module	Loading Knowledge Module (LKM) used to extract the data from the selected space toward the staging area or target space. It is possible to select either a global LKM or a LKM imported into the project. Note: only appropriate KMs for this technology are displayed here.
Description	Detailed description of the LKM, written by the KM developer.
Options	Lists the LKM options with their names and values in a table. Options have a default value specified during the creation of the KM.

6.38.4 Physical Component Properties

Physical components define how a mapping is executed at runtime; they are the physical representation of logical components. Depending on the logical component a physical component might have a different set of properties.

More

General

Properties	Description
Name	Name of the physical component.
Description	Detailed description of the physical component.
Node Component Type	Type of the component.
Logical Component	Name of the logical component linked to this physical component.
Logical Connector Point	Name of the logical connector point linked to this physical component. This applies when the Access Point node has a downstream component with multiple input connection points.
Fixed Execution Unit	Set to true if the physical node is moved explicitly by the user to the parent execution unit. ODI will try to preserve the execution unit and the physical node in that execution unit when synchronizing between the logical and physical mapping.
Fixed by Hint	Set to true if the physical node is moved to the parent execution unit by the Execute on hint property, specified in the logical mapping. ODI may not preserve the execution unit where the current physical node resides when synchronizing between logical and physical mapping, but considers the position of the physical node based on execute on hint.

Properties	Description
Parent Reusable Mapping Components	Lists reusable mappings in which this component is contained.
Execution Context	Datastore Only Context that is used to resolve the logical schema of the datastore physical node to the physical schema. The context of a target datastore can be explicitly set through its Component Context (Forced) property in the logical diagram. If this property is set to some context, the context is shown under Execution Context in the physical diagram.
Datastore Logical Schema	Datastore Only Logical schema of the datastore.
Journalized Data Only	Datastore Only Select to enable Journalized data collection for this datastore.
Temporary Indexes	Join Only Select to define the temporary index information.

Attributes

Attributes are fixed and provided by the logical component. The list of attribute properties varies by component. See [Component Attributes](#) for details.

In addition to the logical attributes, physical components have the following property:

Properties	Description
Execute On (Push Expression)	Directive for the execution location. Select a different execution unit to move expression.

Extract Options

Defines options for data extraction from source components. Shown for source components only. The properties are divided into three tabs: Options, Description, and Advanced.

- **Options:** lists extract options for this module, in a table. Each option has a Name, Description, Value, and Use Default check box. Select the check box to reset the value to default. See knowledge module documentation for details of options available for each module.
- **Description:** A description of the selected knowledge module, if available.
- **Advanced:** The Extract Knowledge Module field shows the EKM selected for data extraction. You can select a different EKM from a drop-down list; only EKMs applicable to the underlying datastore type can be selected.

Integration Knowledge Module

Defines options for data integration into target components. Shown for target components only. The properties include the Integration Knowledge Module field, and two tabs: Options and Description.

- **Integration Knowledge Module:** shows the IKM selected for data integration. You can select a different IKM from a drop-down list; only IKMs applicable to the underlying datastore type can be selected.

- **Options:** lists integration options for this module, in a table. Each option has a Name, Description, Value, and Use Default check box. Select the check box to reset the value to default. See knowledge module documentation for details of options available for each module.
- **Description:** A description of the selected knowledge module, if available.

Check Knowledge Module

Defines options for flow control. Shown for datastore components only. The properties include the Check Knowledge Module field, and two tabs: Options and Description.

- **Check Knowledge Module:** shows the CKM selected for flow control. You can select a different CKM from a drop-down list; only CKMs applicable to the underlying datastore type can be selected.
- **Options:** lists checking options for this module, in a table. Each option has a Name, Description, Value, and Use Default check box. Select the check box to reset the value to default. See knowledge module documentation for details of options available for each module.
- **Description:** A description of the selected knowledge module, if available.

Loading Knowledge Module

Defines options for data loading into target components. Shown for target components only. The properties include the Loading Knowledge Module field, and two tabs: Options and Description.

- **Loading Knowledge Module:** shows the LKM selected for data loading. You can select a different LKM from a drop-down list; only LKMs applicable to the underlying datastore type can be selected.
- **Options:** lists integration options for this component type, in a table. Each option has a Name, Description, Value, and Use Default check box. Select the check box to reset the value to default. See component documentation for details of options available for each component.
- **Description:** A description of the selected knowledge module, if available.

Related Topics

"Physical Design" in the "Creating and Using Mappings" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.39 Marker Group Editor

Use to create and define marker groups.

A **marker group** is a set of flags that can be applied to elements of a project or a model to organize the developments.

The Marker Group Editor has the following tabs:

- [Definition](#)
- [Memo](#)
- [Version](#)
- [Privileges](#)

6.39.1 Definition

Properties	Definition
Group Name	Name of the marker group. This name appears in the user interface.
Group Code	Code of the marker group. This code is used to access a marker from the Oracle Data Integrator tools and API.
Display Position	Position of the marker relatively to the marked object icon in the tree view. If set to Never , the marker appears only in the Marker tab of the Object Editor, and is not visible in the tree view.
Order	Relative position of markers from this group among all markers. The markers with the lowest order value appears first in the lists and the tree view.
Attributes	<p>Multi-state marker groups allow several markers from the same group to be applied to the same object. If this check box is unselected, only one marker from this group may be applied at a time to an object. A priority marker group would be monostate.</p> <p>Auto-increment marker groups do not allow multiples markers on the same object. In addition, when an auto-increment marker is clicked in the tree view, it switches to the next marker from the group. A progress flag would be auto-increment.</p>

Markers

Properties	Definition
Icon	Icon for the marker. Icons are allowed for any marker type. If a marker stores date or a number, the icon should be set to <none>. Markers with no icon do not appear in the tree view.
Name	Name of the marker. This name appears in the user interface.
Code	Marker code. This code is used to access a marker from the Oracle Data Integrator tools and API.
Type	<p>Type of the marker. This field reflects the type of data the marker stores:</p> <ul style="list-style-type: none"> ■ String -This type is also used for icons. ■ Number ■ Date <p>Number and dates do not appear in the tree view and can be changed through the Marker tab of the Element Editor.</p>
Active	Select this check box to display the marker for this group. Otherwise, it is always hidden.
Tooltip	Select this check box to display the tooltip on the marker's icon.

Related Topics

"Using Markers and Memos" in the "Organizing and Documenting Integration Projects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.40 Model Editor

Use to create and configure your data models in Oracle Data Integrator.

A model is a set of datastores corresponding to data structures contained in a physical schema. Models can be organized into model folders.

The Model Editor toolbar has the following items:

- **Reverse-Engineer:** Click to launch the reverse-engineering process.
- **Check Model:** Click to perform a static check on the model.
- **Generate and Deploy:** Click to deploy this datastore as a data service.

The Model Editor has the following tabs:

- [Definition](#)
- [Reverse Engineer](#)
- [Selective Reverse Engineering](#)
- [Control](#)
- [Journalizing](#)
- [Journalized Tables](#)
- [Services](#)
- [Memo](#)
- [Version](#)
- [Privileges](#)
- [Flexfields](#)

6.40.1 Definition

Properties	Description
Name	Name of the model used in the user interface.
Code	Unique code for this model.
Technology	Technology of the data model. As a data model is linked to a single technology, it is a physical data model whose datatypes are native to this technology.
Logical schema	Logical schema of the topology to which this model is attached.
Action Group	Action group used to generate DDL scripts for this model. If no action group is selected, the <Generic Action> group is used. More
Default Folder	Default project folder into which DDL scripts for this model is generated. More Click Search , to search for the project folder or click Clear Folder Selection to remove the folder specified in this field.
Display the Metadata changes in the Model tree	Changes in the database since the last reverse-engineering operation. If this option is selected, all elements of the model that have been deleted or changed in the database since the last reverse-engineering operation are flagged in the tree. Notes: <ul style="list-style-type: none"> ■ Selecting this option may cause reduced performance when browsing the model in the tree. ■ This option works only for models that use Standard reverse-engineering.
Description	Detailed description of the model.

Note: If you duplicate a model, the Technology field changed to read-only in both models.

6.40.2 Reverse Engineer

Use to reverse-engineer.

Reverse-engineering consists in recovering the metadata (the description of the data structure) of an application from the dictionary of the technology that stores this information, and storing it in the Oracle Data Integrator repository.

Properties	Description
Type of Reverse	<ul style="list-style-type: none"> ■ Standard reverse-engineering uses the capacities of the driver (JDBC, ODBC, and so forth) to retrieve the metadata and store it in the repository. ■ Customized reverse-engineering uses a procedure, the Reverse Knowledge Module (RKM), to extract the metadata for a specific type of application, and to store it in the Repository.
Context	<p>Context in which the reverse is executed.</p> <p>The combination of the context and the logical schema allows Oracle Data Integrator to connect to the required data server to access the metadata.</p>
Logical Agent	Agent used in customized reverse-engineering mode.
Types of objects to reverse-engineer	<p>Types of the objects that should be taken into account by the reverse-engineering process.</p> <p>This list may include tables, views, queues, system tables, table aliases, and synonyms.</p>
Mask	<p>Mask which selects the objects to reverse.</p> <p>This mask uses the SQL LIKE syntax. The percent (%) symbol means zero or more characters, and the underscore (_) symbol means one character.</p>
Characters to remove for the Table Alias	<p>Characters to delete in order to derive the alias.</p> <p>Each datastore has an alias to be used in transformation, filter, and check expressions. This is a short name made up, by default, of the first three characters of the datastore name. For applications that contain many tables beginning with the same prefix, you can ignore the prefix by completing the section "Characters to delete". For example, the value 'DWG_' allows you to derive the aliases PRO, CUS, ITE for the tables named DWG_PROD, DWG_CUS, DW_ITEM.</p> <p>Note: If the datastores already exist, the characters specified here are not removed from the table alias. Updating a datastore is not applied to the table alias.</p>
Knowledge Module	<p>This field is only displayed for Customized reverse-engineering.</p> <p>The Reverse-engineering Knowledge Module (RKM) used to extract the metadata for a specific type of application, and to store it in the Repository. It is possible to select either a global RKM or a RKM imported into the project. Note that only for this technology appropriate KMs are displayed here.</p>
Table Alias maximum length	<p>Maximum length of a generated alias for this model.</p> <p>Characters exceeding this length are truncated.</p>

6.40.3 Selective Reverse Engineering

Use to select the datastores to reverse from a list.

This list takes into account the mask and the type of object to reverse that have been defined on the Reverse Engineer tab.

This functionality is only available for Standard reverse-engineering.

Properties	Description
Selective Reverse	Specifies that the next reverse-engineering operation will be selective. That is, it will use the parameters on this tab.
New Datastores	Specifies that you want to reverse-engineer datastores that are not present in the current model.
Existing Datastores	Specifies that you want to re-reverse-engineer datastores already present in the selected model. The metadata for these objects is updated in the repository. Any changes are noted in the execution log.
Objects to Reverse	Allows you to individually specify which datastores to reverse-engineer. The list displayed takes into account the mask and parameters. You can reduce this list by deselecting the objects you do not wish to reverse-engineer. This list is not available in Custom mode.

6.40.4 Control

Use to perform a static control.

The static control (also known as the data quality audit) determines if all the data in a model meets the integrity constraints defined on the datastores. These constraints include references, primary keys, alternate keys, conditions, and mandatory columns. Specifying the appropriate Check Knowledge Module (CKM) is mandatory in order to perform the control.

Properties	Description
Check Knowledge Module	Strategy used for data quality control. Only a CKM valid for the technology of the model can be selected. Only global KMs and KMs imported into at least one project are displayed.
Option	Parameters specified by the CKM. You can change the default values of the parameters. You can see a description for each parameter by selecting the parameter in the table.
Description	Detailed description of the CKM and its parameters.
Execute	Starts a session to perform the data quality check. You must select an agent and a context.

6.40.5 Journalizing

Use to track data changes (Insert/Delete/Update) in the datastores of a model.

This tab allows you to define and configure the journalizing method used for this model.

Properties	Description
Journalizing Mode	Mode used for journalizing this model. More
Journalizing KM	Strategy used for data journalizing. Only a Journalizing Knowledge Module (JKM) that is valid for the technology of the model can be selected. Only global KMs and KMs imported into at least one project are displayed.

Properties	Description
Options	Parameters specified by the JKM. The default values of the parameters can be changed. You can see the description of each parameter at the bottom of the table when the parameter is selected.
Description	Detailed description of the JKM and its parameters.
Subscriber List	List of the subscribers currently tracking changes in this model.

GoldenGate Process Selection

Properties	Description
Capture Process	Name of the Oracle GoldenGate capture process. Lists logical capture process instances available for selection.
Delivery Process	Name of the Oracle GoldenGate delivery process.
Initial Load Capture Process	Name of the Oracle GoldenGate initial load capture process. Lists logical capture process instances available for selection. <ul style="list-style-type: none"> ▪ Select Undefined if not using the initial load. ▪ Initial load process instance can be used only in offline mode.
Initial Load Delivery Process	Name of the Oracle GoldenGate initial load delivery process. Lists logical delivery process instances available for selection. <ul style="list-style-type: none"> ▪ Select Undefined if not using the initial load. ▪ Initial load process instance can be used only in offline mode.

Create Capture Process Properties

Click **Create** on the Capture Process to modify its properties.

Properties	Description
JAgent	Available JAgent instances created under a physical architecture.
Context	Available contexts. Based on the context selection, mapping between logical schemas to the physical object happens.
Process Name	Name of the Oracle GoldenGate capture process.
Logical Process Name	Name of the logical schema object for the capture process.

Create Delivery Process Properties

Click **Create** on the Delivery Process to modify its properties.

Properties	Description
JAgent	Available JAgent instances created under a physical architecture.
Context	Available contexts. Based on the context selection, mapping between logical schemas to the physical object happens.
Process Name	Name of the Oracle GoldenGate delivery process.
Logical Process Name	Logical name of the delivery process.
Target DB Logical Schema	Name of the logical schema object for the delivery process.

6.40.6 Journalized Tables

Use to list the datastores from the model included in the Changed Data Capture (CDC).

This table allows you organize the datastores in the CDC set. More

Properties	Description
Order	Position of the journalized datastore within the CDC set. This order usually depends on the foreign key relationships. A referenced table should have a lower order number than all its referencing tables.
Table Name	Name of the journalized datastore.

6.40.7 Services

Use to configure the generation and the deployment options of data services for this model. More

Properties	Description
Application server	Logical schema corresponding to the application server (web services container) into which the generated web services are deployed.
Namespace	Namespace for your web services. It is used to generate the Web Services Description Language (WSDL).
Package name	Name of the Java Package generated to contain all the web services. Generally, this is of the form <code>com.<company name>.<project name></code> .
Name of the data source	Name of the data source that you defined in the application server, and that corresponds to the server containing your model's data. This name should be prefixed with <code>java:/comp/env/</code> if your J2EE application uses a standard JNDI namespace.
Name of the data service	Name of the data service that is providing the (CDC related) features at model level.
Service KM	Service Knowledge Module used to generate the web service. Only global KMs and KMs imported into at least one project are displayed.
Deployed datastores	List of datastores that are deployed as data services. For each datastore, you can give a data service name and the name of the published entity.

Related Topics

"Creating and Using Data Models and Datastores" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

"Configuring the Model" in the "Setting Up Data Services" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

6.41 Model Folder Editor

Use to create and edit model folders.

A model folder is an object that groups models together. For example, you can group all models based on a particular technology, located at a particular site, or used in a particular project.

The Model Folder Editor has the following tabs:

- [Definition](#)

- [Markers](#)
- [Memo](#)
- [Version](#)
- [Privileges](#)

6.41.1 Definition

Properties	Description
Name	Name of the model folder used in the user interface.
Description	Detailed description of the model folder.

Related Topics

"Creating a New Model Folder" in the "Organizing and Documenting Integration Projects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.42 Modify Case Step

Use to modify the variable to evaluate in the Case step.

Properties	Description
Variable	Variable to use for the conditional branching. You can either select one of the Load Plan variables from the list or click Lookup Variable to add a new variable to the load plan and use it for this Case step.
Step Name	Step type and name of the variable. The Step Name field is automatically populated and can be edited.

Click **Finish** to change the variable in the Case step.

Related Topics

"Working with Variables in Load Plans" in the "Using Load Plans" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.43 Modify Run Scenario Step 1

Use to modify the scenario used in the Run Scenario step.

Click **Lookup Scenario** to select the scenario.

Tip: At design time, you may want to create a Run Scenario step using a scenario that does not exist yet. In this case, instead of selecting an existing scenario, enter a Scenario Name and a Version number and click **Finish**. Later on, you can select the scenario using the [Modify Run Scenario Step 1](#) wizard.

Note that when you use the version number -1, the latest version of the scenario is used.

If your scenario uses variables, click **Next**. Otherwise, click **Finish** to replace the scenario.

Related Topics

"Change the Scenario of a Run Scenario Step" in the "Using Load Plans" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.44 Modify Run Scenario Step 2

Use to select the scenario variables that are added to the Load Plan.

The **Step Name** is automatically populated with the name and version number of the scenario. You can change the Step Name.

Click **Finish** to change the variable to evaluate.

Related Topics

"Change the Scenario of a Run Scenario Step" in the "Using Load Plans" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.45 New Mapping Dialog

Use to create a new mapping in the selected project and folder.

The New Reusable Mapping Dialog creates a new reusable mapping in the selected project and folder.

Properties	Description
Name	Name of mapping or reusable mapping to be created.
Description	Detailed description of mapping or reusable mapping.
Create Empty Dataset	New mapping only If true, creates an empty dataset component in the source area.
Create Default Input Signature	New reusable mapping only If selected, creates an empty input signature in the reusable mapping.
Create Default Output Signature	New reusable mapping only If selected, creates an empty output signature in the reusable mapping.

Related Topics

"Creating a Mapping" in the "Creating and Using Mappings" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.46 New Scenario Dialog

Use to generate new scenarios.

A scenario is designed to put a source component (mapping, package, procedure, variable) into production. A scenario results from the generation of code (SQL, shell, and so forth) for this component.

Notes:

- Once generated, the scenario is stored inside the work repository. The scenario can be exported then imported to another repository (remote or not) and used in different contexts.

- The scenario code (the language generated) is frozen, and all subsequent modifications of the components that contributed to creating it will not change it in any way.
- The **Name** and **Version** fields of the Scenario are preset with the following values:
 - **Name:** The same name as the latest scenario generated for the component
 - **Version:** The version number is automatically incremented, if the latest version number ends with an integer. If the latest version does not end with an integer, the latest version number is displayed in the Version field. Please change the version number according to your needs. Note that you cannot use the same version number and version name for a scenario.

If no scenario has been created yet for the component, a first version of the scenario is automatically created.

New scenarios are named after the component according to the Scenario Naming Convention user parameter.

Specify the following parameters for the scenario generation:

Name: Name of scenario.

Version: Version of the scenario.

Physical Mapping Design: Physical Mapping Design defined in the mapping. Used for mappings only.

Select **Generate scenario as if all underlying objects were materialized** to generate the scenario as if the shortcuts were real objects.

Note: The logical schema of the model containing the shortcut(s) is used for the scenario generation.

The scenario properties are displayed and can be configured in the [Scenario Editor](#).

Related Topics

"Generating a Scenario" in the "Using Scenarios" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.47 Package Editor

Use to create and configure your packages in Oracle Data Integrator.

A Package is made up of a sequence of steps organized into an execution diagram.
More

The Package Editor has the following tabs:

- Overview
 - [Definition](#)
 - [Execution](#)
 - [Scenarios](#)
 - [Markers](#)
 - [Memo](#)
 - [Version](#)
 - [Privileges](#)

- [Flexfields](#)
- [Diagram](#)

6.47.1 Definition

Properties	Description
Name	Name of the package, as it appears in the Oracle Data Integrator user interface.
Description	Detailed description of the package.

6.47.2 Execution

Use to view the execution results of the Package.

A Package can be executed directly without generating a scenario. By executing a Package, the sequence of steps that make it up can be tested. Click **Execute** in the menu bar to start the execution.

The Execution tab is organized into the Direct Executions and the Scenario Executions and the **Scenario Execution** tabs and shows the results of previous executions.

The **Direct Executions** tab contains the following elements:

Properties	Description
Agent	Name of the agent that executed the package. Internal indicates the use of Oracle Data Integrator Studio built-in agent.
Context	Execution context of the package.
Status	State of the Package execution (Done, Error, Running, Waiting, Warning).
Start	Start date and time of execution of the package.
End	End date and time of execution of the package.
Duration	Time taken for execution of the package.
Return Code	Return code of the package.
Message	Package execution error message.

The **Scenario Execution** tab contains the following elements:

Properties	Description
Agent	Name of the agent that executed the scenario. Internal indicates the use of Oracle Data Integrator Studio built-in agent.
Context	Execution context of the scenario.
Status	State of the scenario execution (Done, Error, Running, Waiting, Warning).
Start	Start date and time of execution of the scenario.
End	End date and time of execution of the scenario.
Duration	Time taken for execution of the scenario.
Return Code	Return code of the scenario.

Properties	Description
Message	Scenario execution error message.
Rows	Total number of rows processed by the scenario.
Inserts	Number of rows inserted during the scenario.
Updates	Number of rows updated during the scenario.
Deletes	Number of rows deleted during the scenario.
Errors	Number of rows in error in the scenario.

6.47.3 Scenarios

Use to display in a table the scenarios generated for this component, and their schedules.

Double-click a scenario or a schedule to display its properties.

Click **Generate Scenario** and **Delete** to create and delete scenarios.

The context menus for scenarios and schedules are also available from this tab.

6.47.4 Diagram

Use to define the sequence of Steps within a package in a graphical way.

The Diagram tab has three main components:

- The Package toolbar
- The Package diagram that contains a graphical view of the steps and the sequence of steps
- The Properties panel that displays information about the selected step

In a Package, a step is always, upon success or failure, followed either by another step (linked by an ok (green) or a ko (red) arrow), or by the end of the Package (not represented). More

Package Toolbar

The tools in the Package toolbar can do the following:

- Select a link or a step.
- View/Hide links between Steps after successful Step.
- View/Hide links between Steps after failed Step.
- View/Hide the OK links between Steps.
- View/Hide KO links between Steps.
- Define the next step upon step success.
- Define the next step upon step failure.
- Duplicate the selected step.
- Delete the selected steps or links.
- Reorganize the diagram.
- Edit the linked object to a selected step.
- Debug Step

- Execute the selected step.
- Execute the whole package.
- Printing options.
- Page configuration.

The **Errors** button changes color when there is an error in the package organization (for instance, an unlinked step). Click **Errors** to view a detail of the errors. If there are no errors in the package, the Errors button is disabled.

The Package Diagram

You can add steps to the Package by dragging and dropping an object (a mapping, variable, datastore, and so forth) from the Designer Navigator or a tool from the toolbox into the Package diagram.

In the Package diagram, each step is displayed as an icon specific to the step type. The sequence of steps appear as ok (green) arrows (for success), and ko (red) arrows (for failure). For Evaluate Variable steps, both arrows are green with a marker indicating if the next step that is executed in the condition is true or false.

It is also possible to rearrange the steps in a Package diagram by dragging and dropping steps and links, or clicking **Reorganize** to automatically rearrange the position of the steps.

You can open a step's Properties panel by clicking its icon in the Package diagram. You can also open the linked object (such as a mapping for a flow step, a variable for an evaluate variable step) by double-clicking the **Step** icon.

You can execute, duplicate, edit, or delete a step by right-clicking it and selecting the appropriate option from the context menu.

The Properties Panel

The Properties panel describes the information about the selected step.

Use the Properties panel to configure the steps of your Oracle Data Integrator Packages. The properties on this tab depend on the step type. More

The Properties panel has the following tabs:

- [General](#)
- [Additional Variables](#) (only for OdiStartScen and OdiStartLoadPlan tools)
- [Command](#)
- [Advanced](#)
- [Options](#)
- [Memo](#)
- [Version](#)
- [Privileges](#)

6.47.4.1 General

Use to display the properties of the selected step.

The following properties appear for most step types. The properties on this tab depend on the selected step type.

Properties	Description
Step Name	Name of the step, as it appears in the Oracle Data Integrator user interface.
Type	Type of step. This field does not appear for Oracle Data Integrator tools.
Linked object	Name of the Mapping, Procedure, Model, Sub-model, Datastore, or Variable inserted into the Package to create this step. This field does not appear for Oracle Data Integrator tools.
Path	Path to the linked object. For example, the Project and Folder containing the mapping for a flow step. This field does not appear for Oracle Data Integrator tools.
Physical Mapping Design	Executes the mapping using the selected Physical Mapping Design.

The following tables detail the properties specific to each type of step. The steps that do not have specific properties are omitted from these tables.

For **Check type steps**, one additional element is displayed:

Properties	Description
Delete Errors from the Checked Table	Deletes the erroneous data from the checked datastore, model, or sub-model.
Recurse Sub-Models	Performs the check operation on sub models data stores too.

For **Oracle Data Integrator tool steps**, the tool's parameters are displayed for editing purposes.

For a **Journalizing step**, the following properties appear in addition to the common properties. The fields that are described vary depending on the object being processed (datastore or model) and on the type of journalizing mode (Consistent Set or Simple).

Properties	Description
Journalizing	
Start	Starts journalizing for the datastore, model, or sub-model.
Stop	Stops journalizing for the datastore, model, or sub-model.
Add Subscribers	Creates the subscriptions for the list of subscribers. You can add or remove subscribers from the list using the Add or Delete buttons.
Remove Subscribers	Deletes the subscriptions indicated in the list of subscribers.
Consumption	
Extend Window	Extends the consistency window for the Changed Data Capture set or the datastore.
Purge Journal	Purges the journals by deleting useless entries.
Lock Subscribers	Locks entries in the consistency windows for the given subscribers.
Unlock Subscribers	Unlocks the entries in the consistency windows for the given subscribers. Validates consumption of the changes.
Subscribers	Click Add Subscribers to add a subscriber to the list. Click Remove Subscribers to delete the selected subscriber(s) from the list.

For a **Set variable step**, the following properties appear in addition to the common properties:

Properties	Description
Assign	When this option is selected, the variable takes the value specified in the Value field. If there are no errors in the package, the Errors button is disabled Variable tab.
Increment	This option appears only for numerical-type variables. It increments the variable value with the value defined in the Increment field.
Value	Value assigned to the variable (in the case of an assignment), or the increment added to the variable (in the case of increment of the numerical variable).

For an **Evaluate variable step**, the following properties appear:

Properties	Description
Operator	Relational operator between the value of the variable and the value defined in the Value field on the Variable tab.
Value	Value compared to the value of the variable.

Note: You can compare two variables by entering in the Value field the name of the variable to which a comparison is to be made. The variable name must be prefixed with GLOBAL or the code of the project containing this variable.

Note: When using the IN operator, the variable value is searched for in a semicolon-separated list, such as AB;AC;AA;AE or 66;33;88;11;22;92.

For **Execute OS command type steps**, one additional property is displayed:

Properties	Description
Text of your command	Command is started on the operating system.

Note: OS Commands are started by an agent in a given environment (OS, machine, and so forth), and therefore, the environment-specific restrictions and features should be considered (file access path, shell command syntax, and so forth) in the command text.

6.47.4.2 Additional Variables

This tab appears when using the OdiStartScen and OdiStartLoadPlan tools. You can specify in this tab the list of variable values for this scenario.

6.47.4.3 Command

This tab appears when using an Oracle Data Integrator tool. It contains the command corresponding to the tool call. You can use the **General** tab in the Package properties pane to define this command. More

6.47.4.4 Advanced

This tab allows you to determine the behavior of the package according to whether the step succeeds or fails. More

Default Behavior

The default behavior of steps in a package is as follows:

- If the step is successful, it moves on to the following step within the Package.
- If the step ends abnormally, execution of the Package is stopped.

Behavior following success or failure

You can define the behavior following success or failure by specifying the following parameters:

Properties	Description
Processing after Success	Describes the behavior to implement if the step has been successful.
End	Execution of the package is stopped, and all transactions are subject to a Rollback.
Execute Step	This field allows a package step to be selected. The special step <<Next Step>> allows the step following the current step in the package to be executed.
Processing after failure	Describes the behavior to implement if the step has failed.
Number of Attempts	Defines the number of times the step will be re-run if it has failed.
Interval between Attempts second(s)	The time interval in seconds between two re-launches of the step.
End	Execution of the Package is stopped, and all transactions are subject to a Rollback. This behavior is implemented after the retries.
Execute Step	This field allows a Package step to be selected. The special step <<Next Step>> allows the step following the current step in the Package to be executed. This behavior is implemented after the retries.
Log Steps in the Journal	Describes how the step information is logged in the journal when its execution is finished.
Log Steps in the Journal	Before its execution and while being executed, a step appears in the execution log. This drop box defines whether the step should be kept in the journal after its execution is finished: <ul style="list-style-type: none"> ■ Never: The step is deleted from the journal. ■ Always: The step is always kept in the journal. ■ Errors: The step is kept in the journal only if it failed. Otherwise, it is deleted.

6.47.4.5 Options

This tab allows you to set the values of the options of a procedure or Knowledge Module (for journalizing, checking or reverse-engineering steps).

Properties	Description
Option	Name of the procedure option the value is assigned to.
Value	Value assigned to the option for the execution.

Related Topics

"Creating a New Package" in the "Creating and Using Packages" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.48 Procedure Editor

Use to create and configure your procedures in Oracle Data Integrator.

A procedure is a set of commands that can be executed by an agent. These commands concern all technologies accessible by Oracle Data Integrator (OS, JDBC, JMS commands, and so forth).

The Procedure Editor has the following tabs:

- [Definition](#)
- [Tasks](#)
- [Options](#)
- [Execution](#)
- [Scenarios](#)
- [Markers](#)
- [Memo](#)
- [Version](#)
- [Privileges](#)
- [Flexfields](#)

6.48.1 Definition

Properties	Description
Name	Name of the procedure, as it appears in the user interface.
Multi-Connections	Shows if the procedure accesses paired data servers. If this check box is selected, data loading commands can be exchanged between two data servers, of different technologies if necessary. For example, on procedure commands, you can specify a command "SELECT ..." in one database and a command "INSERT" into another database to transfer the whole result.
Source Technology	Source technology used by default on procedure commands. This information is only available if the procedure is a "Multi-connections"-type and if the technology has existing dataservers defined or is of type AAA, BBB, CCC, or DDD.
Target Technology	Default technology to which the commands of this procedure are addressed. This information is only available if the technology has existing dataservers defined or is of type AAA, BBB, CCC, or DDD.
Use Unique Temporary Object Names	If checked, multiple instances of same procedure can be executed in parallel without interfering with each other.

Properties	Description
Remove Temporary Objects On Error	Indicates if the temporary objects created with the unique names should be deleted if the procedure ends up in Error. This allows to cleanup the temporary objects automatically on Error. If unchecked, and unique temporary objects names are used, then temporary objects created with unique names are not cleaned automatically and should be cleaned manually.
Description	Detailed description of the procedure.

6.48.2 Tasks

A procedure is made up of several commands. The Tasks tab shows a list of these commands, which you can create, delete, and arrange by using the buttons in the toolbar. To edit a task, select the task, and edit in the Properties Editor. To update any information directly in the task list, click a particular field and update it.

General

Properties	Description
Name	Name of the command, as it appears in the user interface.
Cleanup	If procedure flag Remove Temporary Objects OnError is set to true, the tasks flagged as Cleanup tasks get executed even on error.
Log Counter	Shows which counter (Insert, Update, Delete or Errors) records the number of rows processed by this command.
Log Level	Logging level of the command. At execution time, commands can be kept in the log based on their log level.
Ignore Errors	Shows that the procedure is not interrupted in case of an invalid return code. If this box is selected, the procedure command goes into "warning" instead of "error", and the procedure is not be stopped.
Log Final Command	Select to log final code for source and target commands. This improves the readability of logs which have a lot of runtime parameters. Use this option only for tasks that do not contain passwords or other sensitive data.nad

Target Command

Properties	Description
Technology	Technology on which the command is executed. Note: To use Oracle Data Integrator Tools (commands) in KM procedure commands or procedures commands, you must set the technology to ODI Tools. To use OS commands or to execute external programs, you must set the technology to Operating System

Properties	Description
Transaction Isolation	<p>Transaction isolation level for Select orders. The isolation levels shown are those of the SQL ISO. They are not supported by all data servers.</p> <p>The levels of isolation are:</p> <p>Default: The transaction keeps the connection's isolation level.</p> <p>Read Uncommitted: The transaction can read data not committed by another transaction.</p> <p>Read Committed: The transaction can only read data committed by other transactions (in general, this is the default mode of many data servers).</p> <p>Repeatable Read: The transaction is certain to read the same information if it executes the same SQL query several times, even if the rows have been modified and committed by another transaction in the meantime.</p> <p>Serializable: The transaction is certain to read the same information if it executes the same SQL query several times, even if the rows have been modified, deleted, or created and committed by another transaction in the meantime.</p>
Context	<p>Execution context of the query. If "Execution context" is left, the command will be executed in the execution context chosen on launching or on the step.</p>
Schema	<p>Logical schema for execution. The logical schema, linked to the context, allows the physical execution schema to be defined.</p>
Transaction	<p>You can execute commands on several concurrent transactions (numbered from 0 to 9) or work off-transaction by choosing the option Autocommit.</p>
Commit	<p>If your command is being executed in a transaction (numbered 0 to 9), you can decide to continue or to commit the current transaction according to the following modes:</p> <p>No Commit: The transaction is not committed. In this case, it can be committed in a later command. If a session ends normally, all transactions are committed by default.</p> <p>Commit: The transaction is committed.</p> <p>Commit 1000 rows: Oracle Data Integrator commits every 1000 records processed. This choice is only possible on a loading procedure command, that is, one containing a source command returning a result set and a destination command that carries out inserts.</p>
Command	<p>Text of the command expressed in a native language or in a standard language (such as SQL, PL/SQL, Transact-SQL, shell, and so forth). You can use the Expression Editor, accessible through the button to the right of the text.</p> <p>Warning: If the command is made of random characters, then the procedure is encrypted. For more details about encryption and decryption, see "Encrypting and Decrypting Procedures" in <i>Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator</i></p>

Source Command

Properties	Description
Technology	<p>Technology on which the command is executed.</p> <p>Note: To use Oracle Data Integrator Tools (commands) in KM procedure commands or procedures commands, you must set the technology to ODI Tools. To use OS commands or to execute external programs, you must set the technology to Operating System</p>
Transaction Isolation	<p>Transaction isolation level for Select orders. The isolation levels shown are those of the SQL ISO. They are not supported by all data servers.</p> <p>The levels of isolation are:</p> <p>Default: The transaction keeps the connection's isolation level.</p> <p>Read Uncommitted: The transaction can read data not committed by another transaction.</p> <p>Read Committed: The transaction can only read data committed by other transactions (in general, this is the default mode of many data servers).</p> <p>Repeatable Read: The transaction is certain to read the same information if it executes the same SQL query several times, even if the rows have been modified and committed by another transaction in the meantime.</p> <p>Serializable: The transaction is certain to read the same information if it executes the same SQL query several times, even if the rows have been modified, deleted, or created and committed by another transaction in the meantime.</p>
Context	<p>Execution context of the query. If "Execution context" is left, the command will be executed in the execution context chosen on launching or on the step.</p>
Schema	<p>Logical schema for execution. The logical schema, linked to the context, allows the physical execution schema to be defined.</p>
Transaction	<p>You can execute commands on several concurrent transactions (numbered from 0 to 9) or work off-transaction by choosing the option Autocommit.</p>
Commit	<p>If your command is being executed in a transaction (numbered 0 to 9), you can decide to continue or to commit the current transaction according to the following modes:</p> <p>No Commit: The transaction is not committed. In this case, it can be committed in a later command. If a session ends normally, all transactions are committed by default.</p> <p>Commit: The transaction is committed.</p> <p>Commit 1000 rows: Oracle Data Integrator commits every 1000 records processed. This choice is only possible on a loading procedure command, that is, one containing a source command returning a result set and a destination command that carries out inserts.</p>
Command	<p>Text of the command expressed in a native language or in a standard language (such as SQL, PL/SQL, Transact-SQL, shell, and so forth). You can use the Expression Editor, accessible through the button to the right of the text.</p> <p>Warning: If the command is made of random characters, then the procedure is encrypted. For more details about encryption and decryption, see "Encrypting and Decrypting Procedures" in <i>Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator</i></p>

Options

Properties	Description
Always Execute	If checked, the Task executes under any conditions. If not checked, the Task executes under the conditions set in the Options tab.

Tasks Toolbar

Name	Description
Expand All	Expands all the task groups.
Collapse All	Collapses all the task groups in task hierarchy.
Add	Adds a new task.
Delete	Deletes the selected task.
Up	Moves the selected task up in the list.
Down	Moves the selected task down in the list.
Move task to top	Moves the selected task to the top.
Move task to bottom	Moves the selected task to the bottom.
Select Columns	Adds a task property to the task table.

6.48.3 Options

A procedure may be parameterized when executed, by means of options. The Options tab allows all options to be displayed, as well as their default values. The options values specified in this tab are only used during the execution of the procedure.

Properties	Description
Name	Name of the option. It is a common practice to use upper-case names with underscores ("_") such as DETECTION_STRATEGY.
Type	Datatype of the option. Possible values are Boolean, Choice, Text, or Value. Boolean options can be true or false. A choice option allows you to choose the option value from a pre-defined list of option values. These values can be edited using the Edit Choice Dialog . Value options can hold an alphanumeric value up to 250 characters, and Text options can hold an alphanumeric value of unlimited length and are parsed for variable and sequence references.
Default Value	Value of the option that is set by default.
Condition Expression	Click the condition expression cell against a particular option to enter or edit the groovy script, which will determine if a procedure option should be disabled, enabled, shown, or hidden.
Description	Text that will be shown in the Oracle Data Integrator UI next to the option. Text should be short to be readable in a single line inside a table.

Options Toolbar

Name	Description
Add	Adds a new option. Note: The Add Group option is not supported and is therefore disabled.
Delete	Deletes the selected option.
Move Up	Moves the selected option up in the list.
Move Down	Moves the selected option down in the list.
Move to top	Moves the selected option to the top of the list.
Move to bottom	Moves the selected option to the bottom of the list.
Select Column	Selects or deselects the columns to be displayed in the table.

6.48.4 Execution

A procedure is executed on a context and an agent. Click **Run** in the menu bar to start the execution process.

The Execution tab is organized into the Direct Executions and the Scenario Execution tabs and shows the results of previous executions.

The **Direct Executions** tab contains the following elements:

Properties	Description
Agent	Name of the agent that executed the procedure. Internal indicates that the agent used is the one built in the Oracle Data Integrator Studio.
Context	Execution context of the mapping.
Status	State of the procedure execution (Done, Error, Running, Waiting, Warning).
Start	Start date and time of execution of the procedure.
End	End date and time of execution of the procedure.
Duration	The time taken for execution of the procedure.
Return Code	Return code of the procedure.
Message	Procedure execution error message.

The **Scenario Execution** tab contains the following elements:

Properties	Description
Agent	Name of the agent that executed the scenario. Internal indicates that the agent used is the one built in the Oracle Data Integrator Studio.
Context	Execution context of the scenario.
Status	State of the procedure execution (Done, Error, Running, Waiting, Warning).
Start	Start date and time of the execution of the scenario.
End	End date and time of the execution of the scenario.
Duration	The time taken for the execution of the scenario.

Properties	Description
Return Code	Return code of the scenario.
Message	Procedure execution error message.
Rows	Total number of rows processed by the scenario.
Inserts	Number of rows inserted during the scenario.
Updates	Number of rows updated during the scenario.
Deletes	Number of rows deleted during the scenario.
Errors	Number of rows in error in the scenario.

Related Topics

"Creating Procedures" in the "Creating and Using Procedures, Variables, Sequences, and User Functions" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*.

Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator

6.49 Project Editor

Use to create and define a project.

A project is a group of objects developed using Oracle Data Integrator.

The Project Editor has the following tabs:

- [Definition](#)
- [Markers](#)
- [Memo](#)
- [Version](#)
- [Privileges](#)
- [Flexfields](#)

6.49.1 Definition

Properties	Description
Name	Name of the project, as it appears in the Oracle Data Integrator user interface.
Code	Project code used as prefix for project-type variables. Note: If you modify the project code, the name of project-type variables is also changed.

6.49.2 Markers

Markers can be defined in a Project (Project Markers) or in the Others navigator tree in Designer Navigator (Global Markers). The project markers can be used only on objects of the project, and global markers can be used in all models of the repository.

A marker can be set on an object and can contain a value that will appear in the tooltip of the marker icon in the tree view. More

The Markers tab lists the markers set on this object.

Properties	Description
Group	Name of the marker's group.
Marker	Marker assigned to the object.
Value	Value specified for the object.

Click **Insert a Marker** to add a marker to your object.

Select a marker and click **Delete a Marker** to remove a marker.

6.49.3 Memo

Use to attach a descriptive memo to a Project or Model element to reflect your project's methodology or to help with development.

Almost all Project and Model elements can have descriptive memos attached to them.

A memo is an unlimited amount of text attached to virtually any object, visible on its Memo tab. When an object has a memo attached, the Memo icon appears next to it in the Projects tree. More

6.49.4 Version

Use to view and manage versions of your object.

A version is a backup copy of an object and is checked in at a given time and may be restored later. Versions are saved in the Master Repository.

The version management system allows flags on developed objects (such as projects, models, and so forth) to be set automatically to indicate their status (such as new or modified). The version management system also allows these objects to be backed up as stable checkpoints, and later restored from these checkpoints. These checkpoints are created for individual objects in the form of versions.

Creation and Update Information

Properties	Description
Created by	User who created the object.
Created on	Date and time when the object was created.
Updated by	User who updated the object.
Updated on	Date and time when the object was updated.

Information

Properties	Description
Internal ID	Internal ID that enables object identification. Internal ID is unique for the object type within a repository.
Global ID	Global ID of the object. The global ID is unique for the object type across repositories.

Versions

Properties	Description
Current Version: N/A (created)	<p>This table lists the version number, the creation date of the version, the name of the user who created the version, and a version description.</p> <p>You can perform the following actions by clicking the corresponding button:</p> <ul style="list-style-type: none"> ■ Create a new version. ■ Delete a version. ■ Restore a version. ■ Export a version. <p>See "Working with Versions" for more information.</p>

6.49.5 Privileges

Use to view the authorized methods for a given user on this specific object instance.

An object instance (instance) is attached to an object type (an object). For example, the project MY_PROJ_1 is an instance of the project object type (or object). Similarly, another instance of a project object type is YOUR_PROJ_2.

Privileges are set up and managed in Security Navigator. The most common actions in Security Navigator are:

- Creating and editing users and profiles.
- Assigning user rights for methods on ODI objects (such as data server and datatypes).
- Fine-tuning user rights on the object instances.

See "Managing the Security in Oracle Data Integrator" for more information.

Property	Descriptions
Method	<p>Name of the object method.</p> <p>A method is a type of action that can be performed on an object, for example <i>edit</i> or <i>delete</i>. Each object has a series of methods that are specific to it. The concept of methods in Oracle Data Integrator is similar to the one in Object-Oriented programming.</p>
User/Role	<p>Oracle Data Integrator User corresponding to the login name used for connecting to a repository.</p> <p>A user has profile privileges, privileges on objects, and privileges on instances.</p>

6.49.6 Flexfields

Use to customize the properties of Oracle Data Integrator objects.

Flexfields are user-defined fields that are set for each object instance through the Flexfield tab of the object editor, and their values are defined for each object instance through the Flexfield tab of the object's instance window. Flexfield values can be used through Oracle Data Integrator substitution methods.

Note: Flexfields exist only for certain object types. Objects that do not have a Flexfield tab do not support them.

Property	Description
Name	Label of the flexfield as it appears in the Flexfield tab of the Object Editor.
Code	Code of the flexfield.
Technology	Technology for which the flexfield will be activated. For instance, a flexfield on a procedure appears only for Procedures using this technology.
Type	Type of the flexfield (String, Numeric, or Long Text).
Default	Default value of the flexfield.

Related Topics

"Creating a New Project" in the "Creating an Integration Project" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.50 Run <Object> Dialog

Use to specify the run parameters of any object.

Properties	Description
Context	Specifies the context in which the session starts.
Logical Agent	Specifies the agent or Oozie engine that executes the object. The object can also be directly executed by ODI Studio, by selecting Local (No Agent) .
Log Level	Specifies the level of logging information that should be retained. All session tasks with a defined log level lower than or equal to this value are kept in the Session log when the session completes. However, if the object execution ends abnormally, all tasks will be kept, regardless of this setting. Note: Log level 6 has the same behavior as log level 5, but with the addition of variable and sequence tracking. More
Simulation	Select to run a simulation and create a run report. Note: The Simulation option is disabled when an Oozie engine is selected in the Logical Agent field.
Deploy Only	Select to process the scenario, generate the Oozie workflow, and deploy it to the Hadoop Distributed File System (HDFS). Note: The Deploy Only option is available only when running a scenario and an Oozie engine is selected in the Logical Agent drop-down.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

Oracle Fusion Middleware Administering Oracle Data Integrator

Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator

6.51 Reference Editor

Use to create and configure the references of your datastores.

A reference is a functional link between two datastores. It corresponds to a foreign key in a relational model. For example: The INVOICE datastore references the CUSTOMER datastore through the customer number.

The Reference Editor has the following tabs:

- [Definition](#)
- [Columns](#) (for user or database references)
- [Expression](#) (for complex user references)
- [Behavior](#) (for database references)
- [Control](#)
- [Markers](#)
- [Memo](#)
- [Version](#)
- [Privileges](#)
- [Flexfields](#)

6.51.1 Definition

Properties	Description
Name	Reference name.
Type	<ul style="list-style-type: none"> ■ User Reference: A reference defined by the user, based on column equality. ■ Complex User Reference: A reference between two datastores using a complex expression. ■ Database Reference: A reference based on column equality that exists in the database engine.
Model	Model of the referenced table, or the primary key table, or the table known as parent.
Table	Referenced table, or the primary key table, or the table known as parent.
Active on the database	Reference on the database that contains the datastore.

6.51.2 Columns

Use to view and define user or database references.

For user or database references, a list of the correspondence between the foreign key table columns (child table) and the primary key table (parent or referenced table) must be defined. In accordance with the ISO standard for relational models, only the rows in the foreign key table that include all the non-null foreign key columns are checked.

This tab is only displayed for user or database references.

6.51.3 Expression

Use to view and define complex user references.

The criterion for connecting two tables may be complex. In this case, a free expression may be entered in this tab.

Use the Expression Editor to define the free expression. Click **Launch the Expression Editor** to open it.

Verify your expression by clicking **Testing query on the DMBS**.

This tab is only displayed for complex user references.

6.51.4 Behavior

Use to view indicative metadata information.

This information shows what action is launched by the data server if a row is deleted from the primary key table, or if the primary key value for a row is modified in the primary key table.

This tab is only displayed for database references.

6.51.5 Control

Use to display and configure the quality control properties.

The quality control properties are taken into account during a flow control or during a static control.

Control

This section shows the type of quality control for which this reference will be checked.

Properties	Description
Flow	During a flow control for this datastore, the reference is verified if this check box is selected. This information is a default value that can be modified when the mapping is designed. It is recommended to activate this type of control to ensure data quality, even if the target technology also controls the information.
Static	During a static control, that is, a quality control of data already present in this datastore, the reference is verified if the box is selected.

Synchronous Control

This section shows the quality control statistics.

Properties	Description
Number of rows in the table (Without Filter)	Number of rows in the table with no filtering.
Number of rows checked in the table	Number of rows with filter on this datastore. These rows are checked for the reference.
Number of Correct References (on not null columns)	Number of rows with a match in the referenced table.
Number of References with errors	Number of rows with no match in the referenced table.

Click **Check** to retrieve the number of records that respect or do not respect this constraint. To obtain a trace of the rows in error, a static control must be launched from either the Datastore or the Model Editor.

Related Topics

"Adding and Deleting Constraints and Filters" in the "Creating and Using Data Models and Datastores" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.52 Replace Object Dialog

Use to import an object in replace mode.

In the Replace Object Dialog, specify the `.xml` file that contains the replacement object.

Note the following when using the Import Replace mode:

- If your object is currently used by another ODI component, such as a KM used by a mapping, this relationship is not impacted by the import. The mappings automatically use this new KM in the project.
- When replacing a Knowledge Module with another, Oracle Data Integrator sets the options in the new module using option name matching with the old module's options. New options are set to the default value. It is advised to check the values of these options in the mappings.
- Replacing a KM with another one may lead to issues if the KMs are radically different. It is advised to check the mapping's design and execution with the new KM.

Related Topics

"Import Modes" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.53 Sequence Editor

Use to define sequences.

A sequence is a variable that increments itself each time it is used. Between two uses, the value can be stored in the repository or managed within an external RDBMS table.

Oracle Data Integrator supports three types of sequences:

- **Standard sequences:** The last value is stored in the repository.
- **Specific sequences:** The last value is stored in an RDBMS table cell. Oracle Data Integrator undertakes to read the value, to lock the row (for concurrent updates), and to update the row after the last increment.
- **Native sequences:** This type of sequence maps to a sequence defined in and managed by a database engine.

The Sequence Editor has the following tabs:

- [Definition](#)
- [Values](#)
- [Markers](#)
- [Memo](#)
- [Version](#)
- [Privileges](#)

6.53.1 Definition

Properties	Description
Name	Name of the sequence, in the form it is used. This name should not contain characters that could be interpreted as word separators (blanks, and so forth) by the technologies the sequence is used on. Upper and lower case are not distinguished during interpretation. For example: ORDER_NO
Increment	Value of the increment. This value can be positive or negative. This increment only applies to standard and specific sequences.
Scope	Validity range of the sequence, either Global or Project. A sequence can be valid for all projects (a global sequence), or only for the current project. Oracle Data Integrator examines Project-level sequences first, before evaluating Global-level sequences. Thus, if a sequence ORDER_NO exists at both project-level and global-level, only the project-level sequence value can be evaluated.
Standard Sequence	Select this option for storing the sequence value in the Oracle Data Integrator repository.
Position	Last value allocated for a standard sequence.
Specific Sequence	Select this option for storing the sequence value in a table in a given data schema.
Schema	Logical schema containing the sequences table.
Table	Table containing the sequence value.
Attribute	Name of the attribute containing the sequence value.
Filter to retrieve a single row	When the sequence table contains more than one row, a filter should be completed in order to locate this row in the table. This filter picks up the SQL syntax of the data server. For example: CODE_TAB = '3'
Native Sequence	Select this option if your sequence is implemented in the database engine. Position and increment are fully handled by the database engine.
Schema	Logical schema containing the sequence object.
Native sequence name	Name of the database sequence object.

6.53.2 Values

Properties	Description
Context Filter	Select a context for displaying the values for a sequence on this context.
Context	Name of the context.
Value	Value of the sequence in the given context.

Related Topics

"Creating Sequences" in the "Creating and Using Procedures, Variables, Sequences, and User Functions" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.54 Scheduling Editor

Use to schedule the executions of your scenarios using the Oracle Data Integrator built-in scheduler.

A schedule concerns only one scenario, but a scenario can be scheduled in several ways, and can have several schedules. Each enables a start date and a repetition cycle to be specified.

For example:

- **Schedule 1:** Every Thursday at 9 PM, once only.
- **Schedule 2:** Every day from 8 AM to 12 noon, repeated every 5 seconds.
- **Schedule 3:** Every day from 2 PM to 6 PM, repeated every 5 seconds, with a maximum cycle duration of 5 hours.

The Scheduling Editor has the following tabs:




- [Definition](#)
- [Execution Cycle](#)
- [Variables](#)
- [Privileges](#)
- [Version](#)

6.54.1 Definition

Properties	Description
Context	Execution context of the scheduled scenario.
Agent	Agent responsible for executing the scenario.
Log Level	Logging level. All the tasks whose logging level is lower than or equal to this value are saved in the log at the end of execution.

Status

Defines the activation of the schedule.

Icon	Properties	Description
	Active	Schedule is active when the agent is restarted or when the scheduling of the physical agent is updated.
	Inactive	Schedule is not active and will not run.
	Active for the period	Activity range of the schedule. A schedule active for a period of time only runs within this given period.

Execution

Defines the frequency of execution of each execution cycle

Properties	Description
Execution	Frequency of execution (annual, monthly, simple, and so forth). This option is completed by a set of options that depend on this.

6.54.2 Execution Cycle

The execution cycle shows the repetition mode of the scenario.

Properties	Description
None (Execute once)	Scenario is executed only one time.
Many times	<p>Scenario is repeated several times.</p> <ul style="list-style-type: none"> ■ Maximum number of repetitions: The maximum number of times the scenario is repeated during the cycle. ■ Maximum Cycle Duration: The maximum duration of time, after which the scenario is no longer restarted and the cycle stops. ■ Interval between repetitions: The downtime between each scenario execution. <p>Notes:</p> <ul style="list-style-type: none"> ■ The minimum value allowed for a schedule is 1 second. ■ Repetition interval is the time between the start of two jobs. If the first execution takes longer than the repetition interval, then overlap is possible between the first job execution and its repetition.
Constraints	<p>Allows limitations to be placed on one cycle iteration, in the event of a problem during execution.</p> <ul style="list-style-type: none"> ■ Number of attempts on Failure: Maximum number of consecutive execution attempts for one iteration. ■ Stop Execution after: Maximum execution time for one iteration. If this time is reached, the scenario is automatically stopped.

6.54.3 Variables

When you create a schedule for a scenario or Load Plan, you can define the values taken by the parameter variables for the scheduled executions.

Properties	Description
Name	Name of the variable. It is prefixed by GLOBAL for a global variable, or by the code of the project containing the variable.
Latest Value	The variable takes its last value (or its default value if it has never been set) if this check box is selected. If this check box is not selected, the variable takes the value indicated in the Value field.
Value	Value taken by the variable for the execution.

Related Topics

"Scheduling Scenarios and Load Plans" in the "Running Integration Processes" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

6.55 Run... Dialog

Use to specify the run parameters.

Properties	Description
Context	Specifies the context in which the session starts. Only the contexts for which you have the appropriate rights are listed here.
Physical Mapping Design	Specifies the deployment requirements defined in the mapping.
Logical Agent	Specifies the agent that executes the object. The object can also be directly executed by ODI Studio, by selecting Local (No Agent) .
Log Level	Specifies the level of logging information that should be retained. All session tasks with a defined log level lower than or equal to this value are kept in the Session log when the session completes. However, if the object execution ends abnormally, all tasks will be kept, regardless of this setting. Note: Log level 6 has the same behavior as log level 5, but with the addition of variable and sequence tracking. More
Delete Errors from Checked Tables (only for static check on Models, Sub-Models, or Datastores)	Select to remove the rows that are detected as erroneous from the checked tables.
Recurse Sub-models (only for static check on Models or Sub-Models)	Select to check recurse Sub-Models.
Simulation	Select to run a simulation and create a run report.

Click **OK** to start the run.

Related Topics

"Simulating an Execution" in the "Running Integration Processes" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

6.56 Scenario Editor

Use to configure and execute scenarios.

A scenario is designed to put a source component (mapping, package, procedure, variable) into production. A scenario results from the generation of code (SQL, shell, and so forth) for this component.

The Scenario Editor has the following tabs:

- [Definition](#)
- [Variables](#)
- [Privileges](#)
- [Version](#)
- [Flexfields](#)

6.56.1 Definition

Properties	Description
Package/Mapping/Variable/Procedure	Name of the component used to create the scenario.
Name	Name of the scenario.
Version	Version of the scenario.
Deployment	Deployment plan used for generating the scenario.
Description	Detailed description of the scenario.

Concurrent Execution Controller

Properties	Description
Limit Concurrent Executions	If checked, enables the Concurrent Execution Controller for the Scenario.
Violation Behavior	Specifies the violation behavior action. Default is <code>Raise Error and End Execution</code> .
Wait Polling Interval	Specifies the polling frequency (in seconds) for the wait behavior to check for its turn to run. This field accepts only a positive integer, and it is enabled only when Violation Behavior is selected as Wait . Note: A newly generated scenario or a newly created load plan has blank as the default value, meaning that the runtime agent determines the actual value used for the polling (currently the default is 30 seconds in the runtime agent).

Click **Execute** to launch the scenario. The editor for selecting the execution options opens.

6.56.2 Variables

This tab allows you to select the scenario variables. They are displayed as parameters when starting the scenario or when creating a schedule. Selecting parameter variables enables you to hide the user variables that cannot be considered as parameters.

If the **Use All** value is selected, then all variables are considered as parameters. If the **Use Selected** value is selected, you can select the variables to use as parameters.

Related Topics

"Introduction to Scenarios" in the "Using Scenarios" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.57 Scenario Generation Editor

Use to replace, regenerate, or create new scenarios in the selected folder or project.

Scenario Source Objects

Select the objects for which you want to generate a scenario: packages, mappings, procedures, and variables.

Note: Variables object is enabled at project level only.

Marker Filter

Set the filters for the scenario generation based on markers.

Select a group filter from the Group list and a marker from the Marker list.

Generation Mode

Select one of the following generation modes:

- **Replace** to replace the old existing scenarios with the new generated scenarios. New scenarios are created for objects which do not have old existing scenarios.
- **Re-generate** to overwrite the latest scenario version and preserve the schedules.
- **Creation** to generate a new version of the scenarios for all objects.

Select **Generate scenario as if all underlying objects were materialized** to generate the scenario as if the shortcuts were real objects.

Click **OK** to launch the scenario generation.

Related Topics

"Generating a Scenario" in the "Using Scenarios" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.58 Scenario Execution Report Editor

Use to view the report of your scenario executions.

The Scenario Execution Report Editor has the following tab:

- [Definition](#)

6.58.1 Definition

Properties	Description
Physical Agent	Name of the agent that executed the scenario. Internal indicates that the agent used is the one built in the Oracle Data Integrator Studio.
Status	State of the scenario (Done, Error, Running, Waiting, Warning).
Context	Execution context of the scenario.
Return code	Return code of the scenario.
Start	Start date and time of execution of the scenario.
End	End date and time of execution of the scenario.
Duration	Time taken for execution of the scenario.
Error Message	Scenario execution error message with ODI specific exception numbers, if any.
No. of Inserts	Number of rows inserted during the execution of the scenario.
No. of Deletes	Number of rows deleted during the execution of the scenario.
No. of Updates	Number of rows updated during the execution of the scenario.
No. of Rows	Total number of rows processed by the scenario. Note: This value is not the sum of the above values.
No. of Errors	Number of rows in error in the scenario.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

Oracle Fusion Middleware Administering Oracle Data Integrator

6.59 Scenario Variable Dialog

Use to select the variables as startup parameters.

The selected variables are considered as parameters for the scenario. From the Startup Parameters list, select:

- **Use All**, if you want all variables to be startup parameters.
- **Use Selected**, if you want to use the selected variables as parameters and define the variables to use by selecting the variables in the Startup Parameter column.
- **None**, if you do not want to use the variables as startup parameters.

You can use the **Move up** and **Move Down** arrows to change the order of the variables.

Click **OK** to validate the selection and close the dialog box.

Related Topics

"Generating a Scenario" in the "Using Scenarios" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.60 Shortcut Conflicts Dialog

Use to view and fix shortcut conflicts that have been detected during the reverse-engineering process.

The errors occur when the reversed datastore differs from the base datastore referenced by a shortcut in the model.

Properties	Definition
Conflicting Datastore	Name of the datastore generating the error.
Materialize	Select to materialize and reverse-engineer the model as a new datastore in this model. Note: Unselected datastores are not reverse-engineered and their shortcut remains as is. You can select Select All or Unselect All to materialize all listed datastores or undo previous selections.

Click **OK** to apply the changes and close the dialog box.

Related Topics

"Using Shortcuts" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.61 Shortcut Editor

Use to view and configure shortcuts in Oracle Data Integrator.

The Shortcut Editor has the following tabs:

- [Definition](#)
- [Execution](#) (only for shortcuts of packages, mappings, and procedures)
- [Scenarios](#) (only for shortcuts of packages, mappings, and procedures)
- [Version](#)

Click **View Referenced Object** to display the editor of the referenced object.

Click **View Base Object** to display the editor of the base object.

6.61.1 Definition

Properties	Definition
Name	Name of the shortcut. You can modify the default name.
Referenced Object	Path and name of the referenced object.
Base Object	Path and name of the base object.

6.61.2 Execution

This tab is only displayed for shortcuts of Packages, mappings, and procedures.

These objects can be executed directly without having to generate a scenario. The execution of these objects is started by clicking **Execute** in the menu toolbar.

The Execution tab is organized into the Direct Executions and the Scenario Execution tabs and shows the results of previous executions.

The Direct Executions tab contains the following elements:

Properties	Description
Agent	Name of the agent that executed the object. Internal indicates that the agent used is the one built in the Oracle Data Integrator Studio.
Context	Execution context of the object.
Status	State of the object execution (Done, Error, Running, Waiting, Warning)
Start	Start date and time of execution of the object.
End	End date and time of execution of the object.
Duration	Time taken for execution of the object.
Return Code	Return code of the object.
Message	Object execution error message.
Rows	Total number of rows processed by the execution of the object.
Inserts	Number of rows inserted during the execution of the object.
Updates	Number of rows updated during the execution of the object.
Deletes	Number of rows deleted during the execution of the object.
Errors	Number of rows in error in the execution of the object.

The **Scenario Execution** tab contains the following elements:

Properties	Description
Agent	Name of the agent that executed the scenario. Internal indicates that the agent used is the one built in the Oracle Data Integrator Studio.
Context	Execution context of the scenario.
Status	State of the scenario (Done, Error, Running, Waiting, Warning).
Start	Start date and time of execution of the scenario.
End	End date and time of execution of the scenario.
Duration	Time taken for execution of the scenario.
Return Code	Return code of the scenario.
Message	Scenario execution error message.
Rows	Total number of rows processed by the execution of the scenario.
Inserts	Number of rows inserted during the execution of the scenario.
Updates	Number of rows updated during the execution of the scenario.
Deletes	Number of rows deleted during the execution of the scenario.
Errors	Number of rows in error in the execution of the scenario.

Related Topics

"Introduction to Shortcuts" in the "Using Shortcuts" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.62 Sign In Dialog

Use to provide credentials for WebLogic Server.

Properties	Definition
User Name	Name of the WebLogic Server user.
Password	Password of the user.

Related Topics

Oracle Fusion Middleware Administering Oracle Data Integrator

6.63 Label Editor

Use to create and configure labels in Oracle Data Integrator.

A label is a comprehensive and consistent set of interdependent versions of objects. Like other objects, it can be selected at a given time as a version, and may be restored at a later date. Labels are saved in the master repository. A label assembles a group of versions called the label's elements.

A **label** is automatically assembled using cross-references. By scanning cross-references, a label automatically includes all dependant objects required for a particular object. For example, when adding a project to a label, versions for all the models used in this project's mappings are automatically checked in and added to the label. You can also manually add or remove elements into and from the label.

The following objects may be added into labels:

- Projects
- Load Plans
- Models
- Scenarios
- Global Variables, User Functions and Sequences

The Label Editor has the following tabs:

- [Definition](#)
- [Version](#)
- [Privileges](#)

6.63.1 Definition

Properties	Definition
Name	Name of the label. This name appears in the user interface.
Description	Description of the label.
Principal Elements	Elements (projects, load plans, models, scenarios, and so forth) that are manually added to the label. These elements are versions of objects. Click Restore or Delete to manage these versions. Click Synchronize to force Oracle Data Integrator to re-analyze the dependencies, and suggest addition or deletion to the required elements.
Required Elements	Elements that are directly or indirectly referenced by the principal elements. These elements are automatically added by analyzing the dependencies on the principal elements. These elements are versions of objects. Click Restore or Delete to manage these versions.

Related Topics

"Working with Labels" in the "Using Version Control" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.64 Path Selection Dialog

Use to pick an existing path for an outgoing connection from any component with multiple inputs or outputs, for example, the Split component and Subquery Filter component.

In order to change the number of connection points or set a SQL condition, open the properties of the split component.

Properties	Description
Connection Path	Path used for the connection.
SQL Condition	Split component only. SQL condition of connector point.
Connected to	Existing connected components of the connector point.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

6.65 Step Report Editor

Use to view the report for a specific step of your scenario executions.

The Step Report Editor has the following tabs:

- [Definition](#)

6.65.1 Definition

Properties	Description
Name	Name of the executed step.
Order number	Order number of the step for this session.
Status	State of the step (Done, Error, Running, Waiting, or Warning).
Return code	Return code for the step.

Record Statistics

Properties	Description
No. of Inserts	Number of rows inserted during the step.
No. of Updates	Number of rows updated during the step.
No. of Deletes	Number of rows deleted during the step.
No. of Errors	Number of rows in error in the step.

Execution Statistics

Properties	Description
Start	Start date and time of execution of the step.
End	End date and time of execution of the step.
Duration	Time taken for execution of the step.
Number of Executions	Execution number, in the event of successive executions of the step (loops).
Message	Step execution error message with ODI specific error numbers, if any.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

6.66 Sub-Model Editor

Use to create sub-models.

A sub-model is a group of functionally homogeneous datastores within a model. The datastores of a model can be inserted into a sub-model using drag-and-drop, or by automatic distribution.

The Sub-Model Editor has the following tabs:

- [Definition](#)
- [Distribution](#)

- [Version](#)
- [Privileges](#)

Click **Sub-Model Check** to perform an immediate static check of the datastores in the sub-model. This check uses the Check Knowledge Module (CKM) shown at the model level.

6.66.1 Definition

Properties	Description
Name	Name of the sub-model used in the user interface of the Designer Navigator.
Code	Unique code of the sub-model.

6.66.2 Distribution

Use to define a distribution rule of the datastores in your sub-models.

Datastores, depending on the datastores distribution rule for each sub-model, are compared to the automatic assignment mask. If they match this pattern, then they are moved into the sub-model.

There are two methods to classify:

- Click **Distribution** to apply the current rule to the datastores.
- At the end of the reverse-engineering process, all rules are applied in the order specified in the **Order of mask application after a Reverse** field.

Properties	Description
Datastores Distribution rule	<p>Determines which datastores will be considered and compared to the automatic assignment mask:</p> <ul style="list-style-type: none"> ■ No automatic distribution: No datastore is considered. ■ Automatic Distribution of all Datastores not already in a Sub-Model: Datastores located in the root model in the sub-model tree are considered. ■ Automatic Distribution of all Datastores: All datastores in the model (and sub-models) are considered.
Automatic Assignment Mask	Determines the pattern to which the datastores names must adhere, to be classified in this sub-model.
Order of mask application after a Reverse.	<p>Determines the order in which the mask is applied at the end of a reverse-engineering process.</p> <p>Consequently, a rule with a high priority order on all datastores will have precedence. A rule with a high order on non classified datastores will apply only on datastores ignored by the other rules' patterns.</p> <p>At the end of the reverse-engineering process, new datastores are considered as non classified. Those already classified as a sub-model stay attached to their sub-model.</p>

Related Topics

"Creating and Organizing Sub-Models" in the "Organizing and Documenting Integration Projects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.67 Synchronizing Label Dialog

Use to specify the elements to add or remove from the label.

These elements are grouped into the following:

- Principal Elements, which are added manually.
- Required Elements, which are directly or indirectly referenced by the principal elements.
- Unused Elements, which are no longer referenced by the principal elements.

Select **Accept** to version and include the required elements or delete the unused ones.

Click **OK** to synchronize your label.

Related Topics

"Synchronizing Labels" in the "Using Version Control" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.68 Target Load Order Dialog

Use to define the load order of targets.

It can be defined in the properties of the logical mapping diagram (see [Mapping Editor Logical Tab](#)). If no targets are selected to be ordered, ODI determines a load order.

Properties	Description
Available Targets	List of targets that are not ordered.
Ordered Targets	List of targets that are ordered. Use the up and down arrows to move a target up or down in the list.

Related Topics

"Configuring Parallel Target Table Load" in the "Creating and Using Mappings" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

"Specifying Target Order" in the "Creating and Using Mappings" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.69 User Function Editor

Use to define your customized functions.

User functions enable you to define customized functions that can be used in mappings or procedures. These functions are implemented in one or more technologies.

A function can be created as a global function or in a project. In the first case, it is common to all projects, and in the second, it is attached to the project in which it is defined.

The User Function Editor has the following tabs:

- [Definition](#)
- [Implementations](#)
- [Markers](#)

- [Memo](#)
- [Version](#)
- [Privileges](#)

6.69.1 Definition

This tab lets you declare the generic properties for the user function

Properties	Description
Name	Name of the user function. Example: NullValue.
Group	Group of the user function. If you type a group name that does not exist, a new group is created with this group name when the function is saved.
Syntax	<p>Syntax of the user function that appears in the Expression Editor. The arguments of the function must be specified in this syntax. The function's syntax is expressed in one of the following ways:</p> <pre>function_name(\$(arg_name)arg_type, \$(arg_name)arg_type, ...) function_name(\$(arg_name), \$(arg_name), ...) function_name() function_name</pre> <p>arg_type is used to give a type to the arguments and is not mandatory. It may be equal to s (string), n (numeric) or d (date).</p> <p>For instance, to declare the function NullValue, use:</p> <pre>NullValue(\$(variable), \$(default))</pre>
Description	Detailed description of the function.

6.69.2 Implementations

This tab enables you define the implementations of the function for each technology, or for groups of technologies.

Properties	Description
Implementations	Technologies or groups of technologies for which the function is implemented.
Add - Edit - Delete	Enables you to delete an implementation, to create a new implementation, or to change an existing one.
Technologies with no implementations (for information)	Technologies for which the function is not implemented.

If you change or create an implementation the Implementation Editor opens.

Properties	Description
Implementation syntax	<p>Syntax of the function, in the language of the selected technologies. The arguments of the functions are used with the syntax \$(arg_name).</p> <p>For instance, the two implementations of the function NullValue(\$(variable), \$(default)) are:</p> <p>Oracle: nvl(\$(variable), \$(default))</p> <p>Microsoft SQL Server: case when \$(variable) is null then \$(default) else \$(variable) end</p>

Properties	Description
Linked Technologies	List of the technologies associated to this implementation. There can be only one implementation per technology.
Automatically include new technologies	If this check box is selected, the new technologies are automatically added to this implementation. Only one implementation may have this check box selected.

Related Topics

"Working with User Functions" in the "Creating and Using Procedures, Variables, Sequences, and User Functions" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.70 Variable Editor

Use to create and configure variables.

Variables can be used in any expression (SQL or others), as well as within the metadata of the repository. A variable is resolved when the command containing it is executed by the agent or the user interface.

The Variable Editor has the following tabs:

- [Definition](#)
- [Refreshing](#)
- [History](#)
- [Markers](#)
- [Memo](#)
- [Version](#)
- [Privileges](#)

6.70.1 Definition

Properties	Description
Name	Name of the variable, in the form it will be used. This name should not contain characters that could be interpreted as word separators (blanks, and so forth) by the technologies the variable will be used on. Variable names are case-sensitive. That is, "YEAR" and "year" are considered to be two different variables.
Datatype	Type of variable: Alphanumeric (250 characters), Date, Numeric (Maximum 10 digits), or Text (unlimited length).
Keep History	Length of time the value of a variable is kept for: <ul style="list-style-type: none"> ■ No History: The value of the variable is kept in memory for a whole session. ■ Latest value: Oracle Data Integrator stores in its repository the latest value held by the variable. ■ All values: Oracle Data Integrator keeps the history of all the values held by this variable.

Properties	Description
Secure Value	Select Secure Value if you do not want the variable to be recorded. This is useful when the variable contains passwords or other sensitive data. If Secure Value is selected: <ul style="list-style-type: none"> ■ The variable is never tracked and is displayed as unresolved in the source or target code. The variable is not tracked in the repository and is not historized. ■ The Keep History parameter is automatically set to No History and cannot be edited.
Default Value	Value assigned to the variable by default.
Description	Detailed description of the variable.

6.70.2 Refreshing

Refreshing a variable allows assigning a new value based on the result of a SQL-type query. All expressions using this variable are subsequently executed with the new value until a new refresh is done.

Properties	Description
Schema	Logical schema, for identifying the connection on which the SQL query will be executed.
Select Query	Select-type SQL query or any command (execution of a stored procedure) allowing a result array (Resultset) to be returned for a row and a column. For example: <code>Select max(order_no) from crm.order</code> Caution: Table names should be specified in full (catalog, schema, and so forth) because connection parameters do not allow positioning on the right physical schema. Click Refresh to calculate the last value of the variable while executing the query entered above. An editor is displayed for entering the context and the execution agent. You check the execution in the log. Note: If this variable is not persistent, you are not be able to display the value calculated. Use the Expression Editor to define the WHERE expression. Click Launch the Expression Editor to open it. Verify your expression by clicking Testing query on the DMBS .

6.70.3 History

This tab displays the history of the values of the variable with the context for variables with the "Historize" action, or the last value for a variable with the "last value" action.

Note: Each value is attached to the context in which the session was executed:

- The **Context Filter** field lets you filter the entries for a given context.
- Click **Refresh** to update the history.
- Click **Delete** to delete a line of the history.













Related Topics

"Working with Variables" in the "Creating and Using Procedures, Variables, Sequences, and User Functions" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

6.71 HTTP Analyzer

The HTTP Analyzer allows you to examine the content of HTTP request/response package pairs. You can edit the content of a request package, resend it and observe the response packet returned. More

The following buttons are available on the HTTP Analyzer toolbar.

Element	Description
	Click to open the HTTP Analyzer dialog where you can specify a new listener port, or change the default proxy. An alternative way to access these settings is to navigate to the HTTP Analyzer page of the Preferences dialog (available from the Tools menu).
	Click to open the HTTP Analyzer Request/Response window, where you enter payload details, and edit and resend messages.
	Click to start the HTTP Analyzer running. The monitor runs in the background, and only stops when you click  or exit JDeveloper. If you have more than one listener defined, clicking this button starts them all. To start just one listener, click the down arrow and select the listener to start
	Click to stop the HTTP Analyzer running. If you have more than one listener running, clicking this button stops them all. To stop just one listener click the down arrow and select the listener to stop.
	Click to resend a request when you have changed the content of a request. The changed request is sent and you can see any changes in the response that is returned.
	Click to open the Select WS-I Log File to Upload dialog, where you can navigate to an existing WS-I log file.
	Click to save the contents of the HTTP Analyzer to a file.
	Click to invoke the WS-I Analyze wizard which allows you to examine a web service at packet level.
	Click to select all the entries in the HTTP Analyzer.
	Click to deselect all the entries in the HTTP Analyzer.
	Click to clear the selected entries in the HTTP Analyzer.

Related Topics


"Using the ODIInvokeWebService tool" in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

"Monitoring HTTP Using the HTTP Analyzer" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*

"Working with JDeveloper" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*









6.72 HTTP Analyzer Test Window

Use to examine and alter the content of request and response HTTP packets passing through instances of the HTTP Analyzer. More

A separate Request/Response Window is opened each time you click  in the HTTP Analyzer window.

6.72.1 HTTP Analyzer Test Window

The buttons along the top of the window allow you to work with the HTTP Analyzer Test Window.

Element	Description
	Click to open a file browser, in which you specify the name and location of a file to save the data collected so far about the request/response packets and their contents.
	Click to switch the Request and Response panes between side by side and one on top of the other.
	Click to display the previous request/response pair.
	Click to display the next request/response pair.
	Click to resend a request when you have changed the content of a request. The changed request is sent and you can see any changes in the response that is returned.
	Click to remove details of the last request.
	Click to invoke the WS-I Analyze wizard which allows you to examine a web service at packet level. How?
	Click to find this request in the list of transactions in the HTTP Analyzer Log Window. How?



6.72.2 HTTP Analyzer Request/Response Area

As well as resizing the panes by grabbing the bars with the mouse and dragging them, you can quickly maximize and minimize the panes using the up and down arrows for the URL pane, and the left and right arrows for the Request and Response panes.

The elements available depend on the tab at the bottom of the HTTP Analyzer Test window:

- SOAP Structure. Displays the SOAP structure of the data packet of a SOAP web service.
- HTTP Content. The HTTP Analyzer tries to reformat the message and use code highlighting to make the message more readable, for example with JSON it will use JSON code highlighting. It is commonly used for RESTful web services.
- Hex Content. Displays the data as hex values.
- Raw Message. Displays the pure data as text. This text does not wrap, and you may find it easier to copy the message and paste it into a text editor to examine it.
- WADL Structure. The Web Application Description Language (WADL) is an XML-based file format that describes a RESTful web services application. More

Element	Description
URL	The URL of the service.

Element	Description
Method	<p>(Not displayed for SOAP Structure tab or WADL Structure tab) Choose from the list of available methods for the Send Request operation:</p> <ul style="list-style-type: none"> ■ OPTIONS, for example, to show a WADL definition on a JAX-RS service ■ GET to retrieve the content ■ HEAD ■ POST to send a parameter ■ PUT to create new resource ■ DELETE to delete a temporary resource ■ TRACE ■ CONNECT ■ MERGE ■ PATCH <p>If you try to use a method which is not valid XML for the service you are examining a warning message is displayed.</p>
WSDL URL	Displays the URL of the WSDL for the web service being tested.
Open WSDL	Click to open the Select Web Service Description dialog where you can enter, or browse to, the URL of a WSDL document describing the service you want to invoke.
Operations	(Only SOAP Structure tab and WADL Structure tab) Lists the operations available on the service.
Credentials	<p>(Only SOAP Structure tab and HTTP Content tab) When you are using the HTTP Analyzer to test web services that use security policies, choose the credentials to use from the list. HTTPS Credential configuration is the default set of settings available in JDeveloper and it cannot be edited or deleted.</p> <p>You can add new credentials and edit credentials you have created in the Credentials page of the Preferences dialog, available from the Tools menu. Alternatively, click New.</p>
New	(Only SOAP Structure tab) Click to open the Credentials page of the Preferences dialog, where you can define new credentials.
Request Panel	<p>Shows details of the HTTP request sent from the client to the server.</p> <p>At the top is the HTTP header information passed between the client and server on each request. Click + to show the HTTP headers of the request. Click - to hide them.</p> <p>Click Send Request to resend a request when you have changed the content of a request. The changed request is sent and you can see any changes in the response that is returned.</p> <p>Click Copy Request to copy the contents of the request to the clipboard.</p> <p>Click the down arrow next to  to add headers. To remove a header, select it and click </p>
Response Panel	<p>Shows the data body of the HTTP response between the server and the client.</p> <p>At the top is the HTTP header information passed between the server and client on each response. Click + next to Response HTTP Headers to show the HTTP headers of the response. Click - to hide them.</p>







Related Topics

"Monitoring HTTP Using the HTTP Analyzer" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*

"Working with JDeveloper" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*

6.73 HTTP Analyzer Instances Window

Use to see details of the HTTP Analyzer instances currently open in JDeveloper. There will be one tab for each instance. More

Element	Description
	Click to open the HTTP Analyzer dialog where you can specify a new listener port, or change the default proxy.
	Click to open a new instance of the HTTP Analyzer Test window, where you enter payload details, and edit and resend messages.
	Click to start the HTTP Analyzer running. The monitor runs in the background, and only stops when you click the Stop button or exit JDeveloper. If you have more than one listener defined, clicking this button starts them all. To start just one listener, click the down arrow and select the listener to start.
	Click to stop the HTTP Analyzer running. If you have more than one listener running, clicking this button stops them all. To stop just one listener click the down arrow and select the listener to stop.
Name	The name of the instance, made up of the hostname and proxy, is shown. Double-click to display details of the rules running in this instance.
Status	Displays the current status of the instance, for example, Running or Stopped.
Command	Depending on the current status, you can either click  to stop the current instance running, or  to start the instance running.
Activity	Displays the current activity.

Related Topics

"Monitoring HTTP Using the HTTP Analyzer" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*

"Working with JDeveloper" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*

6.74 HTTP Analyzer Dialog

Use to set preferences for the HTTP Analyzer. More

You can use the default settings, or you can set rules and filters for each listener. You can also define outgoing HTTP proxy details for each configured proxy item.

The HTTP Analyzer can be used to examine, or replay the messages of anything that makes use of the HTTP protocol.

Element	Description
Proxy list	A list of the proxies already configured. When you first open this dialog the only proxy listed is the default localhost:8099.
Add	Click to add a new proxy. When you do not change the default host and port, the new proxy is created with a host of localhost, and with the port number incremented by 1, for example localhost:8100.
Remove	Click to delete the selected proxy.
Default Proxy for JDeveloper and New Processes	Select to set the selected proxy as the default. Selected by default.
Host	Displays the host name of the selected proxy. The default is localhost.
Listen On Port	Displays the port of the selected proxy. The default is 8099.
Override Outgoing Proxy (Replace IDE Settings)	Use to override the proxy settings in the Web Browser/Proxy Page of the Preferences dialog. Select this to enable the Configure Proxy button.
Configure Proxy	Click to open the Proxy Settings dialog, where you can enter new proxy server settings, including authentication details. The values you enter here are used by the HTTP Analyzer instead of the proxy settings in the Web Browser/Proxy Page of the Preferences dialog, which is available from the Tools menu.
Rules	Lists the available rules used to run HTTP messages through the HTTP Analyzer.
Configure Rules	Click to open the Rule Settings dialog, where you manage the rules for the HTTP Analyzer.
HTTPS Setup	Click to open the HTTPS Keystore Setup dialog, where you specify the keystores that the HTTP Analyzer uses to handle HTTPS traffic.

Related Topics

"Monitoring HTTP Using the HTTP Analyzer" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*

"Working with JDeveloper" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*

6.75 Proxy Settings Dialog

Use to enable a proxy server to be used by the HTTP Analyzer. These settings are specific to the HTTP Analyzer, and override the settings for the IDE which are set on the Web Browser/Proxy Page of the Preferences dialog. More

If enabled, the HTTP Analyzer will use these settings to access the Internet through your organization's proxy server. If you do not enable these proxy server settings, then the HTTP Analyzer will use the IDE proxy settings.

Element	Description
Use HTTP Proxy Server	Select to enable this dialog.
Host Name	Enter the URL or DNS name of the proxy server you want to use.
Port Number	Enter the port number your machine uses to access the proxy server.

Element	Description
Exceptions	Add exceptions to proxy servers for accessing internal sites. Use " " to separate entries and "*" as a wild card. The default appears when you click Use HTTP Proxy Server . The default is localhost localhost.localdomain 127.0.0.1 : :1.
Proxy Server Requires Authentication	Select to enable the username and password fields for accessing proxies that require authentication.
User Name	Enter the username for the proxy server.
Password	Enter the password for the proxy server.
Test Proxy	Click to test that these proxy settings work in the way you expect.



Related Topics

"Monitoring HTTP Using the HTTP Analyzer" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*

"Working with JDeveloper" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*

6.76 Rule Settings Dialog

Use to set the rules for running the HTTP Analyzer. The fields available depend on the type of rule. [More](#)

Element	Description
Rules list	<p>Lists the available rules. The default is that there are two pass through rules defined. The first is set to http://localhost:631, and the second is blank. The blank Pass through rule must be the last rule.</p> <p>Click Add to add a new pass through rule.</p> <p>Click the down arrow next to Add to choose the type of rule.</p> <p>Click Remove to delete the selected rule.</p> <p>Rules are processed in order from the top, and it is important to order the rules correctly using the  and  buttons</p>
Reference URL	Enter the URL of the reference service you want to use. The HTTP Analyzer connects to this service only once and then uses the rules you define here to run in simulator mode so that you can test what happens.

Related Topics

"Monitoring HTTP Using the HTTP Analyzer" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*

"Working with JDeveloper" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*

6.77 HTTP Keystore Dialog

Use to specify the keystores that the HTTP Analyzer uses to handle HTTPS traffic. [More](#)

When the Integrated WebLogic Server starts for the first time it generates a DemoIdentity that is unique to your machine, and the key in it is used to set up the HTTPS channel. Most of the defaults are already provided in this dialog.

Element	Description
Client Keystore	Enter or browse to the location of the client keystore.
Client Keystore Password	Enter the password for the client keystore.
Client Private Key Password	Enter the client private key password.
Client Trusted Certificate Keystore	Enter, or browse to, the location of the client certificate keystore. The default is <code>jdeveloper-install/Oracle/Middleware/wlserver_10.3/server/lib/DemoTrust.jks</code> .
Client Trusted Keystore Password	Enter the password for the client keystore. The default is <code>DemoTrustKeyStorePassPhrase</code> .
Append default CA certificates	Select to add the CA certificates. Selected by default.
Server Keystore	The keystore on the server. The default is <code>jdeveloper-install/jdeveloper/systemn.n.n.n.n.n.n.n/DefaultDomain/DemoIdentity.jks</code> .
Server Keystore Password	Enter the password for the server keystore. The default is <code>DemoIdentityKeyStorePassPhrase</code> .
Server Private Key Password	Enter the private key password. The default is <code>DemoIdentityPassPhrase</code> .
Default	Click to return to the default settings.

Related Topics

"Monitoring HTTP Using the HTTP Analyzer" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*

"Working with JDeveloper" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*

6.78 Rename Credentials Dialog

Use to rename a credential that you have created. More

Note: You cannot rename the default credential of HTTPS Credential.

Element	Description
New Name	Enter a new name for the selected credential.

Related Topics

"Monitoring HTTP Using the HTTP Analyzer" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*

"Working with JDeveloper" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*

6.79 Select Web Service Description Dialog

Use to enter or browse to the URL of the WSDL you want to examine in the HTTP Analyzer. More

You can also use this window to quickly copy to the clipboard the WSDL URL of a web service deployed to Integrated WebLogic Server or to Oracle WebLogic Server.

Element	Description
WSDL URL	Enter or browse to the location of the WSDL URL.

Related Topics

"Monitoring HTTP Using the HTTP Analyzer" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*

"Working with JDeveloper" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*

6.80 Certificate Error Dialog

This error message is displayed when the certification chain is invalid. More

Related Topics

"Monitoring HTTP Using the HTTP Analyzer" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*

"Working with JDeveloper" in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*

6.81 Dimension and Cube Accordion

Dimension and Cube accordion enables a more flexible and customizable dimensional data loading.

A Dimension Model serves as a folder to contain Dimensions and Cubes objects. The Dimension and Cube objects constitute a separate accordion in the Designer Navigator.

Double-clicking the Dimension object node opens the Dimension Editor.

Double-clicking the Cube object node opens the Cube Editor.

Related Topics

Oracle Fusion Middleware Developing Applications with Oracle JDeveloper

6.82 Dimension Editor

Use Dimension Editor to create a dimension from the start, or to edit a previously created dimension.

Related Topics

Oracle Fusion Middleware Developing Applications with Oracle JDeveloper

6.82.1 Definition Tab

Set values for the following parameters that define the Dimension:

Parameter	Description
Name	Enter the name of Dimension.
Description	Briefly describe the Dimension.
Pattern Name	By default, the Dimension pattern is automatically set. If you want a specific pattern to be applied to the Dimension, select a pattern name.
Binding Datastore	Select the datastore that the Dimension will be bound to.
Binding Surrogate Key Sequence	If you want to use a surrogate key for the Dimension, select an ODI sequence.

Related Topics

Oracle Fusion Middleware Developing Applications with Oracle JDeveloper

6.82.2 Levels Tab

Level represents a collection of dimension values that share similar characteristics. The Levels tab displays a table at the top, listing levels of the Dimension.

6.82.2.1 Levels of Dimension

Levels can be created in any order. Levels can be re-ordered either top-to-bottom or vice versa for better viewing/reporting. It is recommended to create levels in a top-to-bottom manner, that is level without any parent level in Dimension first.

Levels can be created or removed, using the "+" or "x" buttons.

Set values for the following parameters in the table:

Parameter	Description
Name	Enter the name of the level.
Description	Enter description of the level.
Datastore	The datastore used to store the data of this level. This datastore will be the Dimension Binding Table, and can not be edited.
Staging Datastore	The datastore used to stage the incoming data for this level, before loading into the actual binding table. Select a value from a drop-down box. All the datastores in all the models in ODI repository listed in the drop-down box are valid values. This excludes those datastores that are already selected as the binding or error table of the same level.

6.82.2.2 Level Attributes Table

The Level Attributes table displays the attributes of the level selected in the Levels table. Level Attributes can be:

- Created or removed using the "+" or "x" buttons.
- Reordered within the current level using the up or down buttons.

Set values for the following parameters in the table:

Note: Within the same level, only one level attribute can act as the Surrogate Key, Type 2 Setting Start Date, Type 2 Setting End Date.

Dimension Validation has multiple rules to make sure that Level Attributes properties are set correctly.

Parameter	Description
Name	Enter the name of the level attribute.
Surrogate Key	Select the check box to activate the Surrogate Key.
Datatype	Select a value from the drop-down box. Any datatypes in the Generic SQL technology defined in ODI repository, listed in the drop-down box are valid.
Type 2 Setting	Select a value from the drop-down box: <ul style="list-style-type: none"> ■ None ■ Start Date ■ End Date ■ Current Record ■ Trigger History
Type 3 Previous Attribute	Select a value from the drop-down box that lists the level attributes of the current level. Setting this value implies that the current level attribute is SCD3-enabled. When the incoming data contains change to this attribute, its value will be first moved to the SCD3 previous attribute, before being overridden by the incoming data.
Type 3 Start Date Attribute	For Type 3 Start Date, create level attribute of a Date/Timestamp data type, and assign this level attribute to Type 3 Start Date.
Attribute	Select a value from the drop-down box that lists the datastore columns, which store the data for this level attribute. The columns of the dimension table listed in the drop-down box are valid values.
Staging Attribute	Select a value from the drop-down box that lists the datastore columns, which store the data for this level attribute. The columns of the staging datastore selected as the binding of the current level are valid values.

6.82.2.3 Natural Key Members Table

The Natural Key Members table displays the key members for the Natural key of the level selected in the Levels table. The Natural key is also called "Business Key" or "Application Key." This key is usually used to store the key value(s) that can uniquely identify a particular record in the source systems.

The Natural Key Members can be created or removed using the "+" or "x" buttons.

Set values for the following parameter in the table:

Parameter	Description
Level Attribute	Constitutes the Natural Key of the selected level. Select a value from the drop-down box that lists the level attributes of the current level.

6.82.2.4 Parent Level References Table

The Parent Level References table defines the list of references from the selected level in the Level table to some other parent level. This table is driven from the Level table. The Parent Level Reference can be created or removed using the "+" or "x" buttons.

Functionally, a Parent Level Reference defines how to retrieve the parent level record from the current record. One or more attributes in the current record will be used as a foreign key to match the corresponding application key attributes in the parent level.

Set values for the following parameters in the table:

Parameter	Description
Name	Enter the name of the Parent Level Reference.
Description	Enter the details of the Parent Level Reference.
Parent Level	Select the Parent Level that this Reference is pointing to, from the drop-down box. The valid values in the drop-down list are the levels in the Dimension that is higher than the currently selected level.
Type 2 SCD Setting	Select one of the following from the drop-down box: <ul style="list-style-type: none"> ■ None ■ Trigger History If Trigger History is selected, a new row will be created during Dimension loading, if any one of the Parent Level Reference Key Member columns is not the same as the incoming data.

6.82.2.5 Parent Level Reference Key Members Table

This table is driven from the parent level reference that is selected in the Parent Level References table. Each row in this table represents how a key member of the natural key of the parent level matches with the columns in the current level record. The rows in this table are automatically populated with the key members of the parent level.

Set values for the following parameters in the table:

Parameter	Description
Parent Key Attribute	The level attribute that is part of the natural key of the parent level. The value of this column is automatically populated and cannot be changed.
Foreign Key Attribute	The datastore column of the current level that is used to match the corresponding natural key member of the parent level. Select a value from a drop-down box. The valid values of the drop-down box are all the columns of the datastore of the currently selected level.

Parameter	Description
Foreign Key Staging Attribute	The datastore column of the current level that is used to match the corresponding natural key member of the parent level. Select a value from a drop-down box. The valid values of the drop-down box are all the columns of the staging datastore of the currently selected level.

Note: For a star dimension, the key members of a parent level reference do not need to be specified, because a dimension record is stored in the same database row, which includes all information of the parent level.

6.82.3 Hierarchies Tab

Hierarchy defines how dimension data is summarized and rolled up in BI reporting tools.

6.82.3.1 Hierarchies Table

The Hierarchies table shows the list of hierarchies that are defined for this dimension.

Hierarchy can be created or removed using "+" or "x" buttons.

Set values for the following parameters in the table:

Parameter	Description
Name	Enter the name of the Hierarchy.
Description	Enter a short note about the Hierarchy.
Default	If a Dimension has multiple hierarchy, query tools show the default hierarchy. Only one hierarchy can be defined as default.

6.82.3.2 Hierarchy Members Table

The Hierarchy Members table shows the list of level members in the Hierarchy selected in the Hierarchies table.

Hierarchy Members can be:

- Added or removed using the "+" or "x" buttons.
- Sorted using the up and down buttons, for better viewing/reporting.

Set values for the following parameters in the table:

Parameter	Description
Level	<p>Select a level in the Hierarchy from the drop-down box.</p> <p>Valid values in the drop-down box are the list of levels in the Dimension that are not a member of the currently selected hierarchy, which are of lower level than any preceding hierarchy members.</p> <p>For example, if we have a dimension with 3 levels called BRAND, CATEGORY and PRODUCT, if the first hierarchy member is BRAND, the valid levels of the second hierarchy member is CATEGORY and PRODUCT.</p>
Parent Level Reference	<p>Use to navigate from the selected level to the level of the immediately preceding hierarchy members.</p> <p>Select a value from the drop-down box.</p> <p>The parent level references need to be created first in the Levels tab. Valid values are the parent level references between the current level and the level of the immediately preceding hierarchy member.</p>

6.82.3.3 Skip Levels Table

The Skip Levels table shows the list of skip level members in the hierarchy member selected in the Hierarchy Members table. This allows level data to have multiple paths to roll up to different parent levels, where some paths may skip one or more parent levels in the same hierarchy.

For example, assume that we have a hierarchy STORE -> CITY -> REGION -> STATE -> COUNTRY in a STORE Dimension of a retail company. In this company, some stores are more important and they report directly to the regional office. As a result, a store can have two parent levels, CITY and REGION, where CITY is optional. To describe this hierarchy, we have a hierarchy member for STORE level and its parent level is CITY. This hierarchy member also has a skip level and its parent level reference is pointing to REGION level.

As there may be multiple paths that parent levels can be skipped, the + or – buttons can be used to describe these different roll-up paths. The Skip Levels table contains only one column:

Parent Level Reference: The parent level reference that is used to skip to some preceding level in the hierarchy, listed in a drop-down box. Its valid values include:

- Parent level references defined for the level of the currently selected hierarchy member.
- Those parent level references that do not point to the immediately preceding level.

6.83 Cube Editor

Use Cube Editor to create a Cube from the start, or to edit a previously created Cube.

Related Topics

Oracle Fusion Middleware Developing Applications with Oracle JDeveloper

6.83.1 Definition Tab

Enter values for the following parameters in the table:

Parameter	Description
Name	Enter the name of the Cube.
Description	Enter the details of the Cube.
Pattern Name	By default, the Cube pattern is automatically set. If you want a specific pattern to be applied to the Cube, select a pattern name.

6.83.1.1 Binding Table

Set values for the following parameters in the table:

Parameter	Description
Datastore Table	Click the Search icon, and select a datastore to store the Cube data. Clicking the Search icon opens a list of all the datastores in all the models in the ODI repository.

6.83.2 Details Tab

The Details tab contains the following:

- Dimensions Table
- Key Binding Table
- Measures Table

6.83.2.1 Dimensions Table

The Dimensions table displays the list of Dimensions being used by the Cube. You can add or delete a Dimension, using the + or x icons at the top right corner.

Set value for the following parameter in the table:

Parameter	Description
Level	The level of the Dimension that is referenced by the Cube. Click the ... icon to open a list of all the levels in all the Dimensions in ODI repository. Once the level is selected, the qualified name of the level appears in the column in the following format: <dimension_name>.<level_name>
Role	If you plan to use the same Dimension multiple times in a Cube, then enter an alternate name to uniquely identify that Dimension.

6.83.2.2 Key Binding Table

The Key binding table is related to the Dimension table.

Set values for the following parameters in the table:

Parameter	Description
Dimension Key	<p>If any Dimension is selected in the Dimension table, then the Key binding table will display the keys in the Dimension Key column.</p> <p>If the dimension uses a surrogate key, then the Surrogate Key will be displayed.</p> <p>If the dimension does not use a surrogate key, then the Natural Keys will be displayed.</p>
Attribute	<p>Select an attribute from the drop-down list that displays all the attributes in the Cube Binding Table.</p> <p>If any attribute is already bound to a Dimension key or a measure, then that attributes will not be listed</p>

6.83.2.3 Measures Table

Use the Measures Table to:

- Add or delete new measures.
- Bind the measure to a Cube Binding Table attribute and Cube Binding Error Table attribute.

Set values for the following parameters in the table:

Parameter	Description
Name	Enter the name of the Measure.
Description	By default, this will not be visible in the UI, but you can enable it by clicking the "Select Columns" button.
Datatype	<p>Select the datatype of the Level attribute from the drop-down box.</p> <p>The valid values of the drop-down box will be one of the datatypes in the "Generic SQL" technology, defined in the ODI repository.</p>
Size	<p>Specify the length or precision of the Measure.</p> <p>By default, this will not be visible in the UI, but you can enable it by clicking the "Select Columns" button.</p>
Scale	<p>Set the scale of this Measure (for numeric datatype).</p> <p>By default, this will not be visible in the UI, but you can enable it by clicking the "Select Columns" button</p>
Attribute	<p>Select the datastore column to store the data for Measure.</p> <p>The valid values of the drop-down box are the columns of the Cube Table.</p> <p>If any attribute is already bound to a Dimension Key or a Measure, then that attributes will not be listed.</p>

6.84 Dimension Component Properties

The Dimension smart component is used in a mapping to load data into Dimensions and Slowly Changing Dimensions

The Dimension operator contains one group for each level in the Dimension. The groups use the same name as the Dimension Levels. The Level attributes of each Level are listed under the group that represents the Level.

To use a Dimension component in a Mapping, drag the Dimension from the Dimensions and Cubes Model and drop it in the Mapping Editor Canvas.

Related Topics

Oracle Fusion Middleware Developing Applications with Oracle JDeveloper

6.84.1 General Properties

The following are the General Properties of the Dimension Component:

Property	Description
Dimension	A read only property that states what base Dimension it is referred to.
Bound Datastore	A read only property for which the value is derived from the base Dimension, and it is the name of the datastore that the base Dimension is bound to.
Bound sequence	The sequence that the base Dimension will be using.
Storage Type	The type of Dimension being used, namely Star Dimension.
Pattern	By default, this will be Dimension Pattern. If you create your own Patterns, then you can select from it.
Target Load Order	Lists the order in which mapping targets are loaded.

6.84.2 History Properties

The following are the History Properties of the Dimension Component:

Property	Description
Default Effective Time of Initial Record	Applicable to Type 2 SCDs only. Represents the default value assigned as the effective time for the initial load of a particular Dimension record.
Default Effective Time of Open Record	Applicable to Type 2 SCDs only. Represents the default value set for the effective time of the open records, after the initial record. Note: This value should not be modified.
Default Expiration Time of Open Record	Applicable to Type 2 SCDs only. Represents a date value, which is used as the expiration time of a newly created open record for all the Levels in the Dimension.
Slowly Changing Type	A read-only property for which the value will be set, depending on the type of SCD settings.
Type 2 Gap	Applicable to Type 2 SCDs only. Represents the time interval between the expiration time of an old record and the start time of the current record, when a record is versioned. When the value of a triggering attribute is updated, the current record is closed and a new record is created with the updated values. Because the closing of the old record and opening of the current record occur simultaneously, it is recommended to have a time interval between the end time of the old record and the start time of the open record, instead of using the same value for both.

Property	Description
Type 2 Gap Units	Applicable for Type 2 SCDs only. Represents the unit of time used to measure the gap interval represented in the Type2 Gap property. The options are: Seconds, Minutes, Hours, Days, and Weeks. The default value is Seconds.

6.85 Cube Component Properties

The Cube component is a smart component, based on a Cube object. This component has a set of attributes according to the base Cube.

When you drag a Cube object onto the mapping editor, a smart Cube component will be created based on that Cube.

The Cube component only works as a Target and cannot be sourced from.

Related Topics

Oracle Fusion Middleware Developing Applications with Oracle JDeveloper

6.85.1 General Properties

The Cube component has the following General Properties:

Property	Description
Cube	A read-only property that states the base Cube of the Cube component.
Datastore	A read-only property, for which the value is derived from the base Cube. It is the bound datastore of the base Cube of this component.
Pattern	By default, this is CubePattern.
Target Load Order	Loading order for target components. This property is extended from smart component.

6.85.2 Target Properties

The following are the Target Properties of the Cube component:

Property	Description
Integration Type	Help determining the default IKM and limit the set of IKM choices. Select any of the following: <ul style="list-style-type: none"> ■ None ■ Incremental Update ■ Control Append The default value is Incremental Update.

Property	Description
Enable Source Aggregation	<p>If this property is enabled, an aggregate component will be added in the expanded map of the Cube component, before the fact table is loaded.</p> <p>The source row set will be grouped by the Dimension Reference attributes.</p> <p>Measure aggregation functions are determined by the SOURCE_AGGREGATION_FUNCTION attribute properties.</p>

6.85.3 Connector Points

The Cube component can only be used as target, so it has only one input connector point and has no output connector point.

6.85.4 Cube Component Attribute Properties

Cube component has a set of input map attributes derived from the base Cube.

Attribute	Description
Measure attributes	<p>Each measure in base Cube of the Cube component will have a corresponding map attribute in Cube component.</p> <p>The map attribute is bound to the Cube measure.</p> <p>The measure name is used as the map attribute name.</p>
Dimension natural key attributes	<p>Each natural key attribute of the referenced level will have a corresponding map attribute in Cube component.</p> <p>The map attribute is bound to the Natural Key Level Attribute of the referenced level.</p> <p>If the dimension reference of the cube has role qualifier set, then the map attribute name will be in the format <dimension name>_<role name>_<attribute name></p> <p>If the Dimension Reference of the Cube has no role qualifier, then the map attribute name will be in the form of <dimension name>_<attribute name></p>
Active Date	A special map attribute that is created if the base Cube of the Cube component references at least one SCD2 Dimension.

6.85.4.1 Target Properties

If the map attribute in Cube component represents as a Natural Key Identifier of the referenced Dimension, it has the following properties:

Property	Description
Null Data Value	Default is SQL NULL
Attribute Role	A read-only property, which indicates that the attribute is either "Cube Measure" or "Natural Key Identifier."
Bound Object Full Path	<p>A read-only property, which indicates that the attribute is bound to a Natural Key Level Attribute.</p> <p>This lists the full path of the bound attribute.</p>

If the map attribute in Cube component represents as a Cube measure, it has the following properties:

Target Properties:

Property	Description
Source Aggregation Function	The source loading aggregation function for the Measure. This property works jointly with the property "Enable Source Aggregation" on the Cube component. A combo box lists all the aggregate functions of the generic technology.

General Properties:

Property	Description
Attribute Role	A read-only property, which indicates that the attribute represents as a Cube measure.
Bound Object	A read-only property, which indicates that the attribute is bound to a Natural Key Level Attribute.

The Active Date attribute in cube component has following properties:

General Properties:

Property	Description
Data Type	The default data type for Active Date attribute is "TIMESTAMP"
Attribute Role	A read-only property, it is set to Active Date. The Active Date attribute has a default expression, which is set to "FUNC_SYSDATE". "FUNC_SYSDATE" is a global user function. This function is also used in the Dimension Component.

Topology Navigator Editors

This chapter lists the Oracle Data Integrator help topics for the Topology Navigator.

This chapter includes the following topics:

- [Action Editor](#)
- [Action Group Editor](#)
- [Change Work Repository Password Dialog](#)
- [Context Editor](#)
- [Create ODI Work Repository Wizard](#)
- [Data Server Editor](#)
- [Datatype Editor](#)
- [Enterprise Scheduler Dialog](#)
- [Export Execution Environment Dialog](#)
- [Export Logical Topology Dialog](#)
- [Export Master Repository Dialog](#)
- [Export Topology Dialog](#)
- [Import Execution Environment Dialog](#)
- [Import Logical Topology Dialog](#)
- [Import Master Repository Dialog](#)
- [Import Topology Dialog](#)
- [Index Type Editor](#)
- [Language Editor](#)
- [Language Element Editor](#)
- [Logical Agent Editor](#)
- [Logical Oozie Engine](#)
- [Logical Schema Editor](#)
- [Master Repository Editor](#)
- [New Action Line Dialog](#)
- [Oozie Runtime Engine](#)
- [Physical Agent Editor](#)

- [Physical Schema Editor](#)
- [Schedule Dialog](#)
- [Select Repository Dialog](#)
- [Select Repositories Dialog](#)
- [Server Template Generation Wizard](#)
- [Sub-language Editor](#)
- [Technical Environment Dialog](#)
- [Technology Editor](#)
- [Test Connection Dialog](#)
- [Test Connection Dialog for Enterprise Scheduler](#)
- [Topology Navigator](#)
- [Work Repository Editor](#)

7.1 Action Editor

Use to create and configure actions.

Actions are templates for Data Definition Language (DDL) commands. They are used by the Common Format Designer to generate the scripts that implement a data model into a data server or synchronize the differences between a data model described in Oracle Data Integrator and its implementation in the data server.

An action corresponds to a DDL operation (create table, drop reference, and so forth).

The Action Editor has the following tabs:

- [Definition](#)
- [Details](#)
- [Version](#)
- [Privileges](#)
- [Lines](#): One tab per Action Line

7.1.1 Definition

Properties	Description
Name	Name of the action, as it appears in the user interface.
Type	Type of operation performed by the action.
Description	Detailed description of the action.

7.1.2 Details

Each action contains several Action Lines, corresponding to the commands required to perform the DDL operation (for example, dropping a table requires dropping all its constraints first). The Detail tab lists these action lines, which you can create, delete, and organize using the buttons beside the grid.

To edit an existing action, select the tab that corresponds to the action line. The action lines details are described in the [Lines](#) section.

7.1.3 Lines

You can add or delete an action line by clicking the corresponding icon in the Lines toolbar of the Action Editor. You can arrange the order of the action lines by using the arrows to move the range forward or backward.

Properties	Description
Name	Name of the action line. This name will be given to the procedure command generated.
Ignore Errors	This option must be selected if you do not want the procedure to stop when this specific command returns an error. For example, dropping a nonexisting table should not block the whole process.
Commit	Indicates if a commit should be issued at the end of the command.
Log Level	Logging level of the command. At execution time, commands can be kept in the log based on their log level.
Isolation Level	Transaction isolation level for the command. The levels of isolation are: <ul style="list-style-type: none"> ■ Read Uncommitted: The transaction can read data not committed by another transaction. Dirty reads, non-repeatable reads, and phantom reads can occur. ■ Read Committed: The transaction can only read data committed by other transactions (in general, this is the default mode of many data servers). Dirty reads are prevented; non-repeatable reads and phantom reads can occur. ■ Repeatable Read: The transaction is certain to read the same information if it executes the same SQL query several times, even if the rows have been modified and committed by another transaction in the meantime. Dirty reads and non-repeatable reads are prevented; phantom reads can occur. ■ Serializable: The transaction is certain to read the same information if it executes the same SQL query several times, even if the rows have been modified, deleted, or created and committed by another transaction in the meantime. Dirty reads, non-repeatable reads, and phantom reads are prevented. ■ None: Transactions are not supported. ■ Default: This action line does not change the isolation level and the isolation level of the previous action line is used. If no isolation level has been defined yet, the default isolation of the JDBC driver will be used.
Transaction	You can execute commands on several concurrent transactions (numbered from 0 to 9) or work off-transaction by selecting the option Autocommit .
Action Text	Code of the command to execute. Click Expression Editor to open the Expression Editor and write the action text.

Related Topics

"Using Substitution Methods in Actions" in the "Introduction to the Substitution API" chapter in *Oracle Fusion Middleware Developing Knowledge Modules with Oracle Data Integrator*

7.2 Action Group Editor

Use to create and configure action groups.

Actions (templates for DDL commands) are organized into Action Groups. An action group corresponds to a given syntax or purpose (Oracle, SQL-92, and so forth).

The Action Group Editor has the following tabs:

- [Definition](#)
- [Version](#)
- [Privileges](#)

7.2.1 Definition

Properties	Description
Action Group Name	Name of the action group, as it appears in the user interface.
Group Code	The group code allows the action group to be referenced among the different repositories. Caution: Avoid changing this code because this could make some objects inconsistent.
Default Group	Select the Default Group check box if you want this action group to be selected for new models created for this technology. If no default action group is selected for a given technology, then new models use the generic actions.
Description	Detailed description of the action group.

Related Topics

"Generating DDL scripts" in the "Creating Data Models with Common Format Designer" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

7.3 Change Work Repository Password Dialog

Use to change the password of the selected work repository.

Properties	Description
Current Password	Enter the current work repository password.
New Password	Enter the new work repository password. The new password must adhere to the guidelines defined on Password Policies Security Settings Dialog .
Confirm New Password	Enter the new work repository password.

Click **OK** to apply the changes and close the dialog box.

Related Topics

"Changing a Work Repository Password" in the "Administering Repositories" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

7.4 Context Editor

Use to create and configure a context.

A context is a set of resources allowing the operation or simulation of one or more data processing applications. Contexts allow the same jobs (Reverse, Data Quality Control, Package, and so forth) to be executed on different databases or schemas.

In Oracle Data Integrator, a context allows logical objects (logical agents, logical Oozie engines, logical schemas) to be linked with physical objects (physical agents, physical Oozie engines, physical schemas).

Examples:

- The contexts *New York*, *Boston*, and *San Francisco* represent three data processing sites that use the same software with similar data structures that concern sales and marketing management and logistics. New York also operates an accounting package and a data warehouse.
- The *Development* and *Test* contexts allow procedures to be simulated, on replicated databases if necessary.

The Context Editor has the following tabs:

- [Definition](#)
- [Agents](#)
- [Schemas](#)
- [Version](#)
- [Privileges](#)
- [Flexfields](#)

7.4.1 Definition

Properties	Description
Name	Name of the context, as it appears in the user interface.
Code	Code of the context, allowing a context to be referenced and identified among the different repositories. Note: This code must be unique and the most stable possible. Modifying this code may cause readjustment of the context references from the user interfaces of the navigators.
Password	Password requested when the user requests to work on this context. This password allows you to restrict the access rights and avoid unwanted context changes. Note: For sensitive contexts such as production, it is strongly recommended that you enter a password when creating the context. Caution: If the password is not specified, no validation is requested for a change of context.
Default Context Used By Agents	Indicates that this context is displayed by default in the different lists, and selected when opening Designer or Operator.

7.4.2 Agents

This tab allows all the logical agents and logical Oozie engines accessible in this context to be displayed and updated.

The left column in the list contains all existing logical agents and logical Oozie engines. To be able to use a logical agent or logical Oozie engine in the given context, select the corresponding physical agent or physical Oozie engine in the right column.

Note: A logical agent can only be associated with a physical agent and a logical Oozie engine can only be associated with a physical Oozie engine.

7.4.3 Schemas

This tab allows all the logical schemas accessible in this context to be displayed and updated.

The left column in the list contains all existing logical schemas. To be able to use a logical schema in a given context, you must select in the right column the physical schema corresponding to this logical schema in the context.

Related Topics

"Creating a Context" in the "Setting Up a Topology" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator

7.5 Create ODI Work Repository Wizard

Use to create a new work repository.

Several work repositories can be designated with several master repositories, if necessary. However, a work repository can be linked with only one master repository for version management purposes.

Note: If an existing work repository is detected on this connection, you will be asked if you want to attach the work repository to the master repository or if you prefer to recreate the work repository.

How?

The Create Work Repository Wizard has the following steps:

- [Step 1 - Specify ODI Work Repository Connection Properties](#)
- [Step 2 - Specify ODI Work Repository Properties](#)
- [Step 3 - Specify the Password of the ODI Work Repository to attach](#)
- [Step 4 - Specify the name of the ODI Work Repository to attach](#)

7.5.1 Step 1 - Specify ODI Work Repository Connection Properties

Use to specify the Oracle Data Integrator work repository connection details.

Property	Description
Technology	Select the technology of the server to host your work repository. The default value is <i>Oracle</i> .
JDBC Driver	Driver used to access the technology, that hosts the repository.
JDBC Url	Path of the data server to host the work repository. Note: The JDBC Driver and URL parameters are synchronized and the default values are technology dependent.

Property	Description
User	User ID or login of the owner of the tables you are going to create and host of the work repository. Note: The user name should be in uppercase to ensure compliance with the Upgrade Assistant and avoid issues when upgrading the repository.
Password	Password of the user. This password is requested for attaching this work repository to a different master.

Click **Test Connection** to verify that the connection is working.

Click **Next** to move to the next step of the Create Work Repository Wizard.

Oracle Data Integrator verifies whether a work repository already exists on the specified connection.

If a work repository already exists on the specified schema a confirmation message opens.

7.5.2 Step 2 - Specify ODI Work Repository Properties

Use to specify the ODI work repository properties.

If no work repository is found, a new work repository will be created.

Property	Description
Name	Give a unique name to your work repository (for example: <i>DEVWORKREP1</i>).
Password	Enter the password for the work repository.
Type	Select the type for the work repository: <ul style="list-style-type: none"> ▪ Development: This type of repository allows management of design-time objects such as data models and projects (including mappings, procedures, and so forth.). A development repository also includes the run-time objects (scenarios and sessions). This type of repository is suitable for development environments. ▪ Execution: This type of repository only includes run-time objects (scenarios, schedules, and sessions). It allows starting and monitoring of data integration jobs in Operator Navigator. Such a repository cannot contain any design-time artifacts and the Designer Navigator cannot be used with it.

Click **Test Connection** to verify that the connection is working.

Click **Finish** to create a new work repository and close the wizard.

7.5.3 Step 3 - Specify the Password of the ODI Work Repository to attach

Use to specify the password of the ODI work repository to attach.

When an existing work repository has been detected on this connection, the next steps consist in attaching the work repository to the master repository.

In the **Password** field, enter the password of the Oracle Data Integrator work repository to attach.

Click **Next** to move to the next step of the Create Work Repository Wizard.

7.5.4 Step 4 - Specify the name of the ODI Work Repository to attach

Use to specify the name of the ODI work repository to attach.

Property	Description
Name	Enter a unique name for your work repository (for example: <i>DEVWORKREP1</i>).

Click **Finish** to create a new work repository and close the wizard.

Related Topics

"Creating a Work Repository" in the "Administering Repositories" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

7.6 Data Server Editor

Use to create and configure a data server.

A data server is a data processing resource that stores and serves data in the form of tables. It can be a database, a MOM, a connector or a file server.

A data server is always linked with only one technology. The data server is physically identified: it is located on a physical machine accessible by a TCP/IP network.

Caution: It is recommended that you define each data server only once. If the same data server is declared several times (possibly with different connection parameters), Oracle Data Integrator could generate unnecessary data loading phases that waste execution time.

After you have specified the data server details, you can test the data server connection by clicking **Test Connection** in the top left-hand corner of the Data Server Editor. To initialize the data server, click **Initialize**.

The Data Server Editor has the following tabs:

- [Definition](#)
- Depending on the access method selected for the data server:
 - [JDBC](#)
 - [JNDI](#)
- [On Connect/Disconnect](#) only for data servers with technology type Database (JDBC)
- [Properties](#)
- [DataSources](#)
- [JAgent](#)
- [Version](#)
- [Privileges](#)
- [Flexfields](#)

7.6.1 Definition

The Definition tab contains the following generic fields.

Note: For information specific to the File, Hadoop, HBase, Hive, Pig, SOAP Web Service, and Spark data servers, see the corresponding sections. For data servers based on a technology that is a web service container, such as JAX-WS, see [Web Service Container Parameters](#).

Properties	Description
Name	Name of the data server, as it appears in the user interface. Note: For naming data servers, it is recommended to use the following naming standard: <TECHNOLOGY_NAME>_<SERVER_NAME>.
Technology	Technology linked to the data server.
... (Data Server)	Physical name of the data server. Define this name if your data servers can be inter-connected in a native way. For example, for Oracle, define the name of the database link used for accessing this data server from another Oracle data server.

Connection

The following table details the authentication method on the data server.

Properties	Description
User	User name used for connecting to the data server. Depending on the technology, this could be a login, a user, or an account. For some connections using the JNDI protocol, the user name and its associated password are optional (if they have been given in the LDAP directory).
Password	Password linked with the user name. Note: This password is stored encrypted in the repository.
JNDI Connection	Select this box if you can or should connect to your data server using JNDI. Some technologies accept, forbid, or impose connection through a JNDI directory.
Array Fetch Size	When reading large volumes of data from a data server, Oracle Data Integrator fetches successive batches of records. This value is the number of rows (records read) requested by Data Integrator on each communication with the data server.
Batch Update Size	When writing large volumes of data into a data server, Oracle Data Integrator pushes successive batches of records. This value is the number of rows (records written) in a single Oracle Data Integrator INSERT command. Caution: The fetch array and batch update parameters are accessible only with JDBC. However, not all JDBC drivers accept the same values. At times, you are advised to leave them empty. Note (Fetch array and batch update): The greater the number specified in each of these values, the fewer are the number of exchanges between the data server and Oracle Data Integrator. However, the load on the Oracle Data Integrator machine is greater, as a greater volume of data is recovered on each exchange. Batch update management, like that of fetch array, falls within optimization. We recommend starting from a default value (30), then increase the value by 10 each time, until there is no further improvement in performance.

Properties	Description
Degree of Parallelism for Target	Indicates the number of threads allowed for a loading task. Default value is 1. Maximum number of threads allowed is 99.

File

The following table describes the options to be used to specify a Hadoop data server in the File data server.

Properties	Description
Local File	Select to load and write data to the local file system. By default, this option is selected.
HDFS File	Select to load and write data to the HDFS file system. Note: If this option is selected, the JDBC tab is disabled.
Hadoop DataServer	Hadoop data server associated with the File data server. Note: Hadoop data servers are displayed in the Hadoop DataServer drop-down list only if the HDFS File option is selected.

Hadoop

The following table describes the fields specific to the Hadoop data server.

Properties	Description
HDFS Node Name URI	URI of the HDFS node name. For example, <code>hdfs://localhost:8020</code>
Resource Manager/Job Tracker URI	URI of the resource manager or the job tracker. For example, <code>localhost:8032</code>
ODI HDFS Root	Path of the ODI HDFS root directory. For example, <code>/user/<login_username>/odi_home</code>
Additional Classpath	Additional classpaths, if any. For example: <code>/usr/lib/hadoop/*</code> <code>/usr/lib/hadoop/lib/*</code> <code>/usr/lib/hadoop-hdfs/*</code> <code>/usr/lib/hadoop-mapreduce/*</code> <code>/usr/lib/hadoop-yarn/*</code> <code>/usr/lib/oozie/lib/*</code> <code>/etc/hadoop/conf/</code>

Note: For more information regarding the Hadoop data server, see the "Creating and Initializing the Hadoop Data Server" section in *Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator*.

HBase

The following table describes the fields specific to the HBase data server.

Properties	Description
HBase Quorum	Quorum of the HBase installation. For example: localhost:2181
Hadoop DataServer	Hadoop data server associated with the HBase data server.
Additional Classpath	Additional classpaths.

Note: For more information regarding the HBase data server, see the "Setting Up HBase Data Sources" section in *Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator*.

Hive

The following table describes the fields specific to the Hive data server.

Properties	Description
Metastore URI	Hive Metastore URIs. For example: thrift://BDA:10000
Hadoop DataServer	Hadoop data server associated with the Hive data server.
Additional Classpath	Additional classpaths.

Note: For more information regarding the Hive data server, see the "Setting Up Hive Data Server" section in *Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator*.

Pig

The following table describes the fields specific to the Pig data server.

Properties	Description
Process Type	Choose one of the following: <ul style="list-style-type: none"> ▪ Local Mode: Select to run the job in the local mode. By default, the Local Mode option is selected. ▪ MapReduce Mode: Select to run the job in the MapReduce mode. Note: If this option is selected, the Pig data server must be associated with a Hadoop data server.
Hadoop DataServer	Hadoop data server associated with the Pig data server. Note: Hadoop data servers are displayed in the Hadoop DataServer drop-down list only if the MapReduce Mode option is selected as the Process Type.
Additional Classpath	Additional classpaths.

Note: For information regarding the Pig data server, its properties, and additional classpaths, see the "Setting Up Pig Data Server" section in *Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator*.

SOAP Web Service

The SOAP Web Service data server represents the mapping to a Web Services Description Language (WSDL). The following table describes the field specific to the SOAP Web Service data server.

Properties	Description
WSDL URL	Path to a valid WSDL document.

Spark

The following table describes the field specific to the Spark Python data server.

Properties	Description
Master Cluster (Data Server)	Master Cluster URL, which ODI connects to as a client. The local, spark standalone, and yarn-client modes are supported. For example: local [2] spark://hostname:7077 yarn-client

Note: For more information regarding the Spark data server, see the "Setting Up Spark Data Server" section in *Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator*.

Web Service Container Parameters

The following table details the field specific to web service containers.

Properties	Description
Base URL for published services	Base URL into which the web services will be deployed.

Deployment Options

The following table details the deployment method for the web services in this container.

Properties	Description
Save web services into directory	Allows web services to be deployed using file copy in the target directory. The directory should be accessible from the machine performing the web services generation.

Properties	Description
Upload web services by FTP	<p>Allows web services to be deployed using FTP upload. You must specify the target FTP URL as well as a valid FTP user.</p> <ul style="list-style-type: none"> ▪ FTP server URL FTP URL into which Oracle Data Integrator will deploy the web services. ▪ User name/password User name and password to connect the FTP server. This user must have write privileges to the FTP URL.

7.6.2 JDBC

A direct JDBC connection allows you to access a data server without using a JNDI directory.

This tab is only displayed if the JNDI Connection box on the Definition tab is not selected.

Properties	Description
JDBC Driver	<p>Name of the JDBC driver used for connecting to the data server.</p> <p>For more information on installing JDBC drivers, see the <i>Oracle Fusion Middleware Installing and Configuring Oracle Data Integrator</i>.</p>
JDBC URL	<p>URL allowing you to connect to the data server. The URL allows the data server to be located on the TCP/IP network. Each driver has a specific driver syntax.</p>

You can get a list of pre-defined JDBC drivers and URLs by clicking **Display Driver Chooser...**

Note: When creating a data server under the Complex File technology, click **Edit nXSD** to launch the Native Format Builder wizard and create or edit an nXSD file.

Properties

Use to customize ODI Special Drivers. ODI Special Drivers are JDBC drivers that are used to interface with technologies that cannot be handled as relational sources. These technologies are listed below:

- XML
- Complex File
- JMS XML
- JMS
- File
- LDAP

Note: The Properties section on the JDBC tab is only displayed for data servers using the technologies mentioned above.

Properties	Description
Category	Parent node of the property.
Key	Key identifying this property. This key is case-sensitive.
Value	Value for the property.
Use Default	Check this to change the value in the Value column to the default value.
Description	Description of the property.

Note: For information regarding the Properties section on the JDBC tab of the Hive data server, see the "Hive Data Server Connection Details" section in *Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator*.

7.6.3 JNDI

A Java Naming and Directory Interface (JNDI) connection is used for fetching a data server connection, identified by its JNDI Resource name, in a naming or directory service. This directory can be LDAP, Novell Netware NDS, CORBA Naming Service, or the File System. This naming or directory service is accessed through the JNDI protocol. The JNDI connection can be used to fetch connections to databases (JDBC), and is compulsory for accessing Message Oriented Middleware (MOM) through Java Messaging Service (JMS).

This tab only appears if the JNDI Connection check box on the Definition tab has been selected.

Properties	Description
JNDI authentication	<ul style="list-style-type: none"> ■ None: Anonymous access to the naming or directory service. ■ simple: Authenticated access, non-encrypted. ■ CRAM-MD5: Authenticated access, encrypted MD5. ■ <other value>: Authenticated access, encrypted according to <other value>.
JNDI User	User connecting to the naming or directory service.
Password	Password of the user connecting to the naming or directory service.
JNDI Protocol	<p>Protocol used for the connection.</p> <p>Note: Only the most common protocols are listed here. This is not an exhaustive list.</p> <ul style="list-style-type: none"> ■ LDAP: Access to an LDAP directory. ■ SMQP: Access to a SwiftMQ MOM directory. ■ <other value>: Authenticated access following the sub-protocol <other value>.
JNDI Driver	<p>Driver allowing the JNDI connection.</p> <p>Example of the Sun LDAP directory: <code>com.sun.jndi.ldap.LdapCtxFactory</code></p>

Properties	Description
JNDI URL	URL allowing the JNDI connection. For example: ldap://suse70:389/o=linuxfocus.org
JNDI Resource	Name of the resource in the directory server representing the data server connection. For example: cn=sampledb

7.6.4 On Connect/Disconnect

Use to specify the SQL commands to execute when the data server connection is created or closed. How?

On Connect

The parameters in this section specify the SQL command and execution settings that is executed when the data server connection is created.

On Disconnect

The parameters in this section specify the SQL command and execution settings that is executed when the data server connection is closed.

Both sections include the properties described in the following table.

Properties	Description
Command	SQL command code that is executed on the connection. Click Launch the Expression Editor to create or modify the code. The Expression Editor displays only the substitution methods and keywords that are available for the technology of the data server. Note: Global variables are only displayed if the connection to the work repository is available.
Commit	If selected, the connection is committed after executing the SQL command code. Note that if AutoCommit or Client Transaction is selected in the Execute On list, this value is ignored.
Ignore Errors	If selected, the exceptions encountered during the SQL command code execution is ignored. If not selected, the calling operation ends in error status.
Log Level	Logging level from 1 to 5 of the connect or disconnect commands. At execution time, commands can be kept in the session log based on their log level. Default is 3.
Execute On	You can execute commands on several concurrent transactions (numbered from 0 to 9) or work off-transaction by choosing the option Autocommit . Select Client Transaction to execute the command for UI operations. You can select Select All or Unselect All to select or unselect all transactions.

7.6.5 Properties

These properties are passed when creating the connection, in order to provide optional configuration parameters. Each property is a (key, value) pair.

- For JDBC: These properties depend on the driver used. For a list of available properties, see the driver documentation.

- For JNDI: These properties depend on the resource used.

Properties	Description
Key	Key identifying this property. This key is case-sensitive.
Value	Value for the property.

7.6.6 DataSources

On the DataSources tab you can define JDBC datasources that will be used by Oracle Data Integrator run-time agent deployed on a Java EE server to connect to this data server.

These datasources can be defined separately in the Java EE server, or automatically deployed from Oracle Data Integrator.

Properties	Description
Agent	Select the Oracle Data Integrator physical agent for which you want to configure this datasource.
JNDI Name	Datasource name. This name must match the name in your application server.
JNDI Standard	Select JNDI Standard if you want to use the environment naming context (ENC). If JNDI Standard is selected, Oracle Data Integrator automatically prefixes the datasource name with the string <code>java:comp/env/</code> to identify it in the Java EE server's JNDI directory. Note: The JNDI Standard is not supported by Oracle Weblogic Server and for global datasources.

7.6.7 JAgent

Oracle Enterprise Data Quality (EDQ)

The following table details the JAgent properties that must be specified when creating an EDQ data server.

Properties	Description
Name	Name of the physical server.
Server Host	Hostname or IP address of the server where the JAgent process is running.
JMX Port	Port number of the JAgent process. Default is 8090.
User	EDQ Administrator user.
Password	Password of the user credentials.
Domain	Name of the EDQ domain. Note: This is the base mbean to which you connect. Default is <code>edq</code> .

Oracle GoldeGate

JAgent issues the commands to control Oracle GoldenGate processes and modify its settings. External applications use APIs provided by JAgent to adopt the different functionalities offered by Oracle GoldenGate.

Properties	Description
Name	Name of the physical server.
Host	Hostname or IP address of the server where the JAgent process is running.
RMI Port	Port number of the JAgent process.
Manager Port	Port where the Oracle GoldenGate manager instance is running.
JMX User	User name to connect to JAgent.
Password	Password of the user credentials.
Installation Path	Oracle GoldenGate location path for JAgent. Use this path when you create the extract process definitions from a model.

Related Topics

"Creating a Data Server" in the "Setting Up a Topology" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator

Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator

7.7 Datatype Editor

Use to define the datatypes associated with the technology.

These datatypes allow Oracle Data Integrator to generate handling scripts.

The technologies that store the formatted data allocate to each of them a type that defines their nature. For example: numeric, character, date, and so forth.

Some data server access drivers allow you to reverse the datatypes automatically from the Technologies tree in Topology Navigator.

To reverse the datatypes from the Technologies tree in Topology Navigator:

1. Right-click the technology.
2. Select **Datatypes Reverse-engineering...**

The Datatype Editor has the following tabs:

- [Definition](#)
- [Converted to](#)
- [Converted from](#)
- [Version](#)
- [Privileges](#)
- [Flexfields](#)

7.7.1 Definition

Properties	Description
Code	Code used in Oracle Data Integrator to refer to this datatype. This code is unique for this technology. The code is stored in models and flows in order to refer to the datatype.
Name	Name of the datatype, as it appears in the graphic interface and the reports.
Reversed Code	Code returned by the driver or a SQL reverse query.
Length Default	Default value for the Length parameter in a data type.
Length Min	Minimum accepted value for the Length parameter in a data type.
Length Max	Maximum accepted value for the Length parameter in a data type.
Scale Default	Default value for the Scale parameter in a data type.
Scale Min	Minimum accepted value for the Scale parameter in a data type.
Scale Max	Maximum accepted value for the Scale parameter in a data type.

SQL Generation Parameters

Properties	Description
Create Table Syntax	Syntax used to create a column of this type. The tags %L (Length) and %P (Precision) can be used in this syntax.
Writable Datatype Syntax	Alternate syntax used to create a column that stores data for a writable datatype. The tags %L (Length) and %P (Precision) can be used in this syntax. If the type is writable, then the Writable Datatype syntax and Create Table syntax should be the same.

Graphical Parameter

Properties	Description
Letter used for the icon	Letter used to represent the datatype (graphic ID code for this datatype).

Writable

Select this check box if this datatype can be used in an INSERT or UPDATE command. This is not the case, for example, for datatypes for which values are automatically set and cannot be modified such as IDENTITY columns.

Columns that are reversed with a datatype that is not writable are flagged read-only.

A datatype that is not writable should have an alternate Writable Datatype syntax defined, to enable Oracle Data Integrator to create temporary tables that store data of this type.

7.7.2 Converted to

The Converted to tab allows you to specify for the other technologies, the datatype corresponding to the current type. This datatype will be used if Oracle Data Integrator

needs to create a work table on another technology that is a reflection of a datastore in the current technology.

For example:

The Oracle datatype VARCHAR2 will be transformed into VARCHAR on a Microsoft SQL Server data server.

For the data flow of an Oracle instance towards a SQL Server instance, the language generation engine may require conversion in both directions (for example, from Oracle to SQL Server and from SQL Server to Oracle).

New Datatype Conversion Rule Dialog

Use to create a new datatype conversion rule.

Click the green plus sign to open the New Datatype Conversion Rule dialog.

Properties	Description
Technology	Name of the technology.
Datatype	Defines to which data type, your current data type, should be converted while data transformation.
Condition	<p>If it is not clear which is the best match, the Condition field allows you to provide a Java expression when this conversion will happen. The expression may use the variables SIZE and SCALE, which represent the length and scale of the source column. The expression is evaluated, after substituting the length and scale values for the SIZE and SCALE variables. If true for the given source datatype and target technology, then the target datatype may be used when converting from source technology to target technology.</p> <p>For example:</p> <p>When you open a String datatype in the File technology and switch to the 'Converted To' tab, add the condition, <code>SIZE != null && SIZE <= 4000</code> in the line for Oracle - VARCHAR2. Then add another line for Oracle with CLOB as the datatype and in the Condition column add <code>SIZE != null && SIZE > 4000</code>. With this change, if the length of a String column in the File technology is greater than 4000, then the corresponding column created in Oracle technology will be CLOB.</p>
Length Expression	<p>Allows to constrain the length of the datatype to be its maximum size, without taking up additional space.</p> <p>This field allows you to type an expression like this.</p> <p><code>min(SIZE, 38)</code></p> <p>This will use SIZE as the length unless SIZE is greater than 38, in which case, 38 will be used.</p>
Scale Expression	<p>Allows to constrain the length of the datatype to be its maximum size, without taking up additional space.</p> <p>This field allows you to type an expression like this.</p> <p><code>min(SIZE, 38)</code></p> <p>This will use SIZE as the length unless SIZE is greater than 38, in which case, 38 will be used.</p>

7.7.3 Converted from

This tab is used to view, in read-only mode, the data types of the other technologies that correspond to the current data types.

Related Topics

Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

7.8 Enterprise Scheduler Dialog

Use to define an enterprise scheduler.

Use the button **Test Connection** on the toolbar to test the connection of the enterprise scheduler.

Properties	Description
Name	Name of the Enterprise Scheduler.
Provider Factory	JNDI Provider to access Metadata Service and Runtime Service JNDI Resources.
URL	JNDI URL to connect to the Metadata Service and Runtime Service.

Design JNDI Parameters

Properties	Description
Metadata Service JNDI Name	JNDI Resource to access the Metadata Service (MDS).
Username	User name to connect to this metadata service.
Password	Password of the user.

Execution JNDI Parameters

Properties	Description
Runtime Service JNDI Name	JNDI Resource to access the runtime service.
Asynchronous Request JNDI Name	JNDI Resource to callback for reporting job completion.
Use same credentials as Metadata Service	Select to use the same username and password as specified in the Design JNDI Parameters section.
Username	User name to connect to this runtime service
Password	Password of the user.

Related Topics

Oracle Fusion Middleware Administering Oracle Data Integrator

7.9 Export Execution Environment Dialog

Use to export the execution environment either as .xml files into a directory or as a zip file that contains the .xml files.

The export of the execution environment includes the technologies, data servers, contexts, generic actions, load balanced agents, physical schemas, and agents.

In the Export Execution Environment Dialog, set the export parameters as follows:

Properties	Description
Export to directory	Directory in which the export files are created.
Export to zip file	Select if you want to generate a zip file.
Zip File Name	Name of the zip file.

Encryption

For more information about Encryption see, [Export Key fields in Export Dialog](#)

Advanced Options

Use these options to customize the XML output file format. It is recommended that you use the default values.

Properties	Description
XML Character Set	Encoding specified in the export file. The encoding parameter in the XML file header. <?xml version="1.0" encoding="ISO-8859-1"?>
Java Character Set	Java character set used to generate the file.

Related Topics

"Export/Import Topology and Security Settings" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

7.10 Export Logical Topology Dialog

Use to export the logical topology either as .xml files into a directory or as a zip file that contains the .xml files.

The export of the logical topology includes the technologies (connection, datatype, or language information), logical agents, logical schemas, actions, and action groups.

In the Export Logical Topology Dialog, set the export parameters as follows:

Properties	Description
Export to directory	Directory in which the export files are created.
Export to zip file	Select if you want to generate a zip file.
Zip File Name	Name of the zip file.

Encryption

For more information about Encryption see, [Export Key fields in Export Dialog](#)

Advanced Options

Use these options to customize the XML output file format. It is recommended that you use the default values.

Properties	Description
XML Character Set	Encoding specified in the export file. Parameter encoding in the XML file header. <?xml version="1.0" encoding="ISO-8859-1"?>
Java Character Set	Java character set used to generate the file.

Related Topics

"Export/Import Topology and Security Settings" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

7.11 Export Master Repository Dialog

Use to export the master repository and its topology and security settings either as .xml files into a directory or as a zip file that contains the .xml files.

The objects that are exported when exporting the master repository are objects, methods, profiles, users, languages, versions (if option selected), solutions (if option selected), open tools, password policies, entities, links, fields, lookups, technologies, datatypes, datatype conversions, logical agents, contexts, and the child objects.

In the Export Master Repository Dialog, set the export parameters as follows:

Properties	Description
Export to directory	Directory in which the export files are created.
Export to zip file	Select if you want to generate a zip file.
Zip File Name	Name of the zip file.

Encryption

For more information about Encryption see, [Export Key fields in Export Dialog](#)

Advanced Options

Use these options to customize the XML output file format. It is recommended that you use the default values.

Properties	Description
XML Character Set	Encoding specified in the export file. The encoding parameter in the XML file header. <?xml version="1.0" encoding="ISO-8859-1"?>
Java Character Set	Java character set used to generate the file.
Export versions	Select if you want to export all stored versions of objects that are stored in the repository. Unselect this option in order to reduce the size of the exported repository, and to avoid transferring irrelevant project work.
Export solutions	Select if you want to export all stored solutions that are stored in the repository. Unselect this option in order to reduce the size of the exported repository, and to avoid transferring irrelevant project work.

Related Topics

"Exporting and Importing the Master Repository" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

7.12 Export Topology Dialog

Use to export the full topology either as .xml files into a directory or as a zip file that contains the .xml files.

The export of the topology includes the full topology (logical and physical architectures including the local repository, data servers, agents, generic actions, technologies, datatypes, logical schemas, and contexts).

In the Export Topology Dialog, set the export parameters as follows:

Properties	Description
Export to directory	Directory in which the export files are created.
Export to zip file	Select if you want to generate a zip file.
Zip File Name	Name of the zip file.

Encryption

For more information about Encryption see, [Export Key fields in Export Dialog](#)

Advanced Options

Use these options to customize the XML output file format. It is recommended that you use the default values.

Properties	Description
XML Character Set	Encoding specified in the export file. The encoding parameter in the XML file header. <?xml version="1.0" encoding="ISO-8859-1"?>
Java Character Set	Java character set used to generate the file.

Related Topics

"Export/Import Topology and Security Settings" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

7.13 Import Execution Environment Dialog

Use to import the execution environment.

Importing the execution environment imports the technologies, data servers, contexts, generic actions, load balanced agents, physical schemas, and agents.

In the Import Execution Environment Dialog, set the import parameters as follows:

Properties	Description
Import Mode	Select the Import Mode. Possible values are: Synonym Mode INSERT, Synonym Mode UPDATE, and Synonym Mode INSERT_UPDATE.
Import From a Folder	Select to import the logical topology from a folder.
Import From a Zip File	Select to import the logical topology from a zip file.

In the text field, enter the file import folder or zip file. Note that you can browse for the import folder or zip file.

Related Topics

"Import Types" and "Export/Import Topology and Security Settings" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

7.14 Import Logical Topology Dialog

Use to import the logical topology.

Importing the logical topology imports the technologies (connection, datatype, or language information), logical agents, logical schemas, actions, and action groups.

In the Import Logical Topology Dialog, set the import parameters as follows:

Properties	Description
Import Mode	Select the Import Mode. Possible values are: Synonym Mode INSERT, Synonym Mode UPDATE, and Synonym Mode INSERT_UPDATE.
Import From a Folder	Select to import the logical topology from a folder.
Import From a Zip File	Select to import the logical topology from a zip file.

In the text field, enter the file import folder or zip file. Note that you can browse for the import folder or zip file.

Related Topics

"Import Types" and "Export/Import Topology and Security Settings" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

7.15 Import Master Repository Dialog

Use to import the master repository.

Importing the master repository imports objects, methods, profiles, users, languages, versions (if option was selected for export), solutions (if option was selected for export), open tools, password policies, entities, links, fields, lookups, technologies, datatypes, datatypes conversions, logical agents, contexts, and the child objects.

In the Import Master Repository Dialog, set the import parameters as follows:

Properties	Description
Import Mode	Select the Import Mode. Possible values are: Synonym Mode INSERT, Synonym Mode UPDATE, and Synonym Mode INSERT_UPDATE.
Import From a Folder	Select to import the logical topology from a folder.
Import From a Zip File	Select to import the logical topology from a zip file.

In the text field, enter the file import folder or zip file. Note that you can browse for the import folder or zip file.

Related Topics

"Import Types" and "Exporting and Importing the Master Repository" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

7.16 Import Topology Dialog

Use to import the topology.

Importing the topology imports the logical and physical architectures including the local repository, data servers, hosts, agents, generic actions, technologies, datatypes, logical schemas, and contexts.

In the Import Topology Dialog, set the import parameters as follows:

Properties	Description
Import Mode	See Import Modes in <i>Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator</i> for more information.
Import From a Folder	Select to import the topology from a folder.
Import From a Zip File	Select to import the topology from a zip file.

In the text field, enter the file import folder or zip file. Note that you can browse for the import folder or zip file.

Related Topics

"Import Types" and "Export/Import Topology and Security Settings" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

7.17 Index Type Editor

Use to create and edit index types.

The Index Type Editor has the following tabs:

- [Definition](#)
- [Version](#)
- [Privileges](#)

7.17.1 Definition

Properties	Description
Name	Name of the index type.
Code	Code of the index type.
Description	Detailed description of the index type.
SQL Keyword	Refer to your database documentation.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

7.18 Language Editor

Use to define the languages used by Oracle Data Integrator.

Oracle Data Integrator uses computer languages to access technologies. These languages are used to generate the procedures executed on these technologies. The correct definition of the characteristics of a language is therefore essential for the correct generation and execution of the procedures.

In Oracle Data Integrator, a language is described by its language elements assembled in sub-languages.

The languages are used for:

- Determining the language elements available in the Expression Editor.
- Managing the separators and delimiters of language words when generating procedures.

Concepts

A language is made up of three types of words:

- **Objects:** Named entities handled by the language (tables, schemas, columns).
- **Literals:** Values that the language handles (for example, content of a record column in a table, for SQL language).
- **Reserved words:** Specific to the language, and are usually the names of functions, commands, and so forth.

The Language Editor has the following tabs:

- [Definition](#)
- [Version](#)
- [Privileges](#)

7.18.1 Definition

Properties	Description
Name	Name of the language.

Delimiters

This section allows the characters used as delimiters to be defined for the language.

Properties	Description
Objects	<p>Pairs of characters that allow the names of objects to be delimited. This type of character is generally used to protect the case (upper/lower) or to protect the possible separators of special words or characters used in object names.</p> <p>Each pair is made up of a start character and an end character, and there cannot be more than two characters.</p> <p>Examples:</p> <ul style="list-style-type: none"> ■ Square brackets ([]) <p>For example: [Active Customers]</p> <p>The square bracket is necessary to protect the space and the case in the name of the object Active Customers.</p> <ul style="list-style-type: none"> ■ Double quotes (" ") <p>For example: "Account" ."Licenses"</p> <p>The quotation marks are used to protect the accents and the case in the name of the objects Account (schema) and Licenses (datastore).</p>
Literals	<p>Pairs of characters allowing literals (values) to be delimited.</p> <p>Each pair is made up of a start character and an end character, and there cannot be more than two characters.</p> <p>For example:</p> <p>" : 'Welcome to Paris !'. The quotation marks are used to protect the accents, the case, and the special characters in this literal.</p>

Case-Sensitive

Determines the case-sensitivity for the language.

Properties	Description
Objects	Indicates whether the language differentiates upper and lower case for the names of the objects it processes, such as the names of columns, tables, and schemas.
Reserved keywords	Indicates whether the language differentiates upper and lower case for its reserved words, such as SELECT or INSERT.

Word Separators

The characters used to separate the words of the language.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

7.19 Language Element Editor

Use to create and configure language specific elements.

A language element is an operator, a function, or a keyword of a language. This language element can have several implementations depending on the technology the language is implemented on.

For example:

The function that returns the current date in a SQL language query is declared as the language element CURDATE, but is implemented in the technologies supporting SQL

language as TODAY (Sybase, Informix) or CURDATE (Progress, DB2/400, and so forth).

The Language Element Editor has the following tabs:

- [Definition](#)
- [Implementation](#)
- [Version](#)
- [Privileges](#)

7.19.1 Definition

Properties	Description
Name	Name of the language element.
Expression	Generic syntax of the language element. The implementation syntax for a technology is described on the Implementation tab. However, if no implementation exists for the technology and if the latter must have access to the language element (Universal or Standard-type), then this is the default syntax used by the Expression Editor.
Group function	Shows that this language element handles a group of elements. (For example: MAX, MIN, and so forth).
Universal	Shows that this element is a universal type and should be inherited by all technologies supporting this language.
Standard	Shows that this element is a standard type and should be inherited by all technologies with the compatible check box selected for this language.
Help	Help text and description of the element.

7.19.2 Implementation

Properties	Description
Name	Name of the implementation of the language element.
Expression	Syntax of the language element in this implementation. This syntax is the one displayed in the Expression Editor.
Technology	Technology in which this implementation takes place.
Exception	If this box is selected, then the language element does not exist for the given technology, even if it is universal or standard-type.
Help	Help text for a specific implementation of a language element. Defaults to the help text for the implementation-independent element definition.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

7.20 Logical Agent Editor

Use to create and define the logical agents.

A logical agent is an alias that allows a unique name to be given to all the physical agents with the same function in different contexts.

- The aim of the logical agent is to simplify going into production and scheduling on several contexts.
- To be usable, a logical agent must be declared in a context. Declaring a logical agent in a context consists in indicating the physical agent it corresponds to in the given context.

The Logical Agent Editor has the following tabs:

- [Definition](#)
- [Privileges](#)

7.20.1 Definition

Properties	Description
Name	Name of the logical agent. The name should suggest the function of the agent or OS it is based on. Caution: This name is used as ID code by the objects contained in the work repositories accessed by other Oracle Data Integrator Navigators. Uncontrolled modification of this name may cause manual readjustment of the references from the user interfaces of the Oracle Data Integrator Navigators.
Context	List of the contexts declared in Oracle Data Integrator.
Physical Agent	Shows the physical agent corresponding to the logical agent in this context. An undefined value indicates that the logical agent does not exist in the context.

Related Topics

"Creating a Logical Agent" in the "Setting Up a Topology" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

7.21 Logical Oozie Engine

Use to create and define a logical Oozie engine corresponding to a physical Oozie engine within a context.

The Logical Oozie Engine has the following tabs:

- [Definition](#)
- [Privileges](#)

7.21.1 Definition

Properties	Description
Name	Name of the logical Oozie engine.
Context	List of the contexts declared in Oracle Data Integrator.
Physical Agents	Shows the physical Oozie engine corresponding to the logical Oozie engine in this context. An undefined value indicates that the logical Oozie engine does not exist in the context.

Related Topics

"Executing Oozie Workflows" in *Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator*

7.22 Logical Schema Editor

Use to define the logical schema.

A logical schema is an alias that allows a unique name to be given to all the physical schemas containing the same datastore structures.

- The aim of the logical schema is to ensure the portability of the procedures and models on the different physical schemas. In this way, all developments in Designer are carried out exclusively on logical schemas.
- A logical schema can have one or more physical implementations on separate physical schemas, but they must be based on data servers of the same technology. A logical schema is always directly linked to a technology.
- To be usable, a logical schema must be declared in a context. Declaring a logical schema in a context consists of indicating which physical schema corresponds to the alias - logical schema - for this context.

For example:

- The logical schema LEDGER is the set of Sybase tables required for the functioning of the accounting application. These tables are stored in a physical schema for each installation of the accounting application. The application was installed once in Boston and twice in Seattle (in production and in test).
- Work in Designer or Operator is always done on the logical schema LEDGER. Only the context allows the physical schema on which the operations are actually done to be determined. Thus, the user can switch from one physical environment to another in a single action.

The following table illustrates this example.

Name of the logical schema	Context	Physical Schema
LEDGER	Boston	Sybase Boston LDG
LEDGER	Seattle Production	Sybase SEATTLE PROD LDG
LEDGER	Seattle Test	Sybase SEATTLE TEST LDG

The Logical Schema Editor is made up of the following tabs:

- [Definition](#)
- [Privileges](#)
- [Flexfields](#)

7.22.1 Definition

Properties	Description
Name	<p>Name of the logical schema. It is recommended to give a name that suggests the functional content of the schema (program or application name). For example, LEDGER, CRM, ACCOUNTING.</p> <p>Caution: This name is used as ID code by the objects contained in the work repositories accessed by other Oracle Data Integrator Navigators. Uncontrolled modification of this name may cause a manual readjustment of the references from the user interfaces.</p>
Context	List of the contexts declared in Oracle Data Integrator.
Process Type	<p>Type of the process that performs Changed Data Capture in Oracle GoldenGate.</p> <ul style="list-style-type: none"> ▪ Capture: Reads the changes from database log files and writes them to trail files, a common Oracle GoldenGate format. This process is database dependent, it needs to understand the format of log files corresponding to the database. One process per database. A capture process can extract from any number of tables the single access details in the configuration. ▪ Delivery: Reads the content from a common trail file and populates the target database. This process can be database dependent or common. A delivery process can update only one database, but it can update multiple tables accessed through the same account.
Target DB Logical Schema	Database schema where the selected process type writes content to.
Physical Schema	Shows the physical schema that corresponds to the logical schema in this context. An undefined value shows that the logical schema does not exist in the context.

Related Topics

"Creating a Logical Schema" in the "Setting Up a Topology" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

7.23 Master Repository Editor

Use to configure the master repository connection details.

The master repository is a data structure that contains information on the topology of a company's IT resources, on security and on version management of projects and data models. This repository is stored on a relational database accessible in client/server mode from the different navigators.

Generally, only one master repository is necessary.

However, in exceptional circumstances, it may be necessary to create several master repositories in one of the following cases:

- Project construction over several sites not linked by a high-speed network (for example, off-site development).
- Necessity to clearly separate the mappings' operating environments (development, test, production), including on the database that contains the master repository. This may be the case if these environments are on several sites.

Master repository domains

The master repository has two functional domains:

- **Topology:** This domain is compulsory. It contains the description of the technologies, the data servers, and the agents. The information in this domain can be modified through the Topology Navigator.
- **Security:** This domain is compulsory. It contains Oracle Data Integrator internal metadata, and the storage structure for information about the users and their privileges. The information in this domain can be modified through the Security Navigator.

The Master Repository Editor has the following tabs:

- [Definition](#)
- [Version](#)
- [Privileges](#)

7.23.1 Definition

Properties	Description
Name	Name of the master repository.
External ID	Unique ID code for the repository.
Successful Installation	Shows if the installation has finished successfully. If this indicator is not positioned, it is probably because the installation has not finished correctly.
Repository Version	Version of the master repository.
Global Upgrade Key	Converts old IDs to a global unique identifier.
Authentication Mode	Displays the authentication mode used by this master repository: <ul style="list-style-type: none"> ■ Use ODI Authentication manages users using ODI's internal security system ■ Use External Authentication uses an external enterprise identity store, such as Oracle Internet Directory, to manage user authentication

Related Topics

"Creating the Master Repository" in the "Administering Repositories" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

7.24 New Action Line Dialog

Use to insert a new action line.

Enter the name of the new action line.

Click **OK** to add the new action and open the Action Line tab of the Action Editor.

Related Topics

"Using Substitution Methods in Actions" in the "Introduction to the Substitution API" chapter in *Oracle Fusion Middleware Developing Knowledge Modules with Oracle Data Integrator*

7.25 Oozie Runtime Engine

Use to create and define the Oozie runtime engine.

The Oozie Runtime Engine toolbar provides the following menu items:

Menu Item	Description
Test	Tests the connections and configurations of the Oozie runtime engine and the associated Hadoop data server.
Initialize	Initializes the Oozie runtime engine.

The Oozie Runtime Engine has the following tabs:

- [Definition](#)
- [Properties](#)
- [Version](#)
- [Privileges](#)
- [Flexfields](#)

7.25.1 Definition

Use to configure the connectivity parameters to the Oozie runtime engine and customize the ODI log retrieval workflow parameters.

Properties	Description
Name	Name of the Oozie runtime engine that appears in Oracle Data Integrator.
Host	Network name or IP address of the machine on which the Oozie runtime engine has been launched.
Port	Listening port used by the Oozie runtime engine. By default, this port is 11000.
Web application context	Name of the web application context defined when deploying the Oozie runtime engine in the application server. Type <code>oozie</code> as the value of this field.
Protocol	Protocol used for the connection. Possible values are <code>http</code> or <code>https</code> . Default is <code>http</code> .

Hadoop Connection

Properties	Description
Hadoop Server	Name of the Hadoop server where the Oozie engine is installed. This Hadoop server is associated with the Oozie runtime engine.
Poll Frequency	Frequency at which the Hadoop audit logs are retrieved and stored in ODI repository as session logs. The poll frequency can be specified in seconds (s), minutes (m), hours (h), days (d), and years (y). For example, 5m or 4h.
Lifespan	Time period for which the Hadoop audit logs retrieval coordinator stays enabled to schedule audit logs retrieval workflows. Lifespan can be specified in minutes (m), hours (h), days (d), and years (y). For example, 4h or 2d.

Properties	Description
Schedule Frequency	Frequency at which the Hadoop audit logs retrieval workflow is scheduled as an Oozie Coordinator Job. Schedule workflow can be specified in minutes (m), hours (h), days (d), and years (y). For example, 20m or 5h.

7.25.2 Properties

Use to configure the property associated with the Oozie runtime engine:

- **OOZIE_WF_GEN_MAX_DETAIL:** Limits the maximum detail (session level or fine-grained task level) allowed when generating ODI Oozie workflows for an Oozie engine. The options available in the Value column are SESSION and TASK.

Related Topics

"Executing Oozie Workflows" in *Oracle Fusion Middleware Integrating Big Data with Oracle Data Integrator*

7.26 Physical Agent Editor

Use to create and define the physical agents.

The agent is the ODI component that orchestrates the run-time execution of ODI scenarios.

It is a Java service which can be placed as listener on a TCP/IP port.

This service allows:

- execution on demand of jobs (model reverses, packages, scenarios, mappings, and so forth), from navigators.
- execution of scheduled scenarios, in addition to executions on demand. The physical agent contains a scheduler that allows scenarios to be launched automatically according to predefined scheduling.

The Physical Agent Editor toolbar provides the following menu items:

Menu item	Description
View Schedule	Displays the agent's schedule.
Update Schedule	Updates the agent's schedule.
Test	Verifies that the agent is running correctly.
Generate Server Template	Starts the Server Template Generation Wizard.
Apply Settings	Applies the changes in the agent cache configuration by refreshing them from repository.

The Physical Agent Editor has the following tabs:

- [Definition](#)
- [Datasources](#)
- [Properties](#)
- [Load balancing](#)

- [Version](#)
- [Privileges](#)
- [Flexfields](#)

7.26.1 Definition

Properties	Description
Name	Name of the agent used in the user interfaces. Note: Avoid using Internal as agent name. Oracle Data Integrator uses the Internal agent when running sessions using the internal agent and reserves the Internal agent name.
Host	Network name or IP address of the machine the agent has been launched on. In some TCP/IP configurations, it is preferable to specify an IP address (for example, 195.10.10.5) rather than a machine name.
Port	Listening port used by the agent. By default, this port is the 20910.
Web application context	Name of the web application context defined when deploying this agent in the application server.
Protocol	Protocol to use for the agent connection. Possible values are http or https. Default is http.
Maximum number of sessions	Maximum number of session allowed on this agent. This value is used when using Load Balancing.
Maximum number of threads	Maximum number of threads allowed for all the session executions on an agent. The maximum threads value should be greater or equal to the maximum sessions value.
Maximum threads per session	Maximum number of threads allowed per session. If the value is set to 0, no limits per session thread are imposed.
Maximum cache entries	Maximum number of session blueprints instances cached in memory. Caching Session Blueprint reduces the corresponding scenario invocation time. Default value is 100; value 0 means no limit on cache size.
Unused Blueprint Lifetime (Sec)	Timeout value in seconds indicating the maximum idle time for a cached session blueprint. Default value is 600 seconds. 0 indicates no timeout.

7.26.2 Datasources

Use to define the data servers or the Oracle Data Integrator repositories that you want to manage as Java EE datasources.

When this agent is deployed on a Java EE server, it connects to the datasource specified on this tab.

Properties	Description
Dataserver	Name of the data server or repository you wish to manage as a Java EE datasource.
JNDI Name	Datasource name. This name must match the name in your application server.

Properties	Description
JNDI Standard	Select JNDI Standard, if you want to use the environment naming context (ENC). If JNDI Standard is selected, Oracle Data Integrator automatically prefixes the datasource name with the string <code>java:comp/env/</code> to identify it in the Java EE server's JNDI directory. Note: The JNDI Standard is not supported by Oracle Weblogic Server and for global data sources.

Add DataServer or Work/Execution Repository Dialog

Use to add a DataServer or Work/Execution Repository accessed by this agent using as J2EE DataSources.

7.26.3 Properties

Use to configure properties.

- **Use Pre-Emptive Load Balancing:** If checked, each time a session finishes, sessions are re-balanced.
- **Scenario Naming Convention:** Defines the default naming pattern for new scenarios created for objects via the Studio or the tools.
- **JDBC Connection Timeout:** Specifies number of seconds to wait during database connections. Default: 3000 ms.
- **Rollback All Open Transactions On Step Failure:** If checked, all open transactions are rolled back immediately when a step fails. If unchecked, you can commit open transactions via KO path. If session completes in Error, all open transactions are rolled back.
- **Keep Alive Interval in Seconds:** Specifies the number of seconds the connection is to be kept alive. The default value is 60 seconds.

7.26.4 Load balancing

Oracle Data Integrator implements load balancing between physical agents.

Each physical agent is defined with:

- A maximum number of sessions it can execute simultaneously.
- Optionally, a number of linked physical agents to which it can delegate sessions' executions.

An agent's load is determined at a given time by the ratio (Number of running sessions / Maximum number of sessions) for this agent.

Determining the maximum number of sessions

The maximum number of sessions is a value that must be set depending on the capabilities of the machine running the agent. It can also be set depending on the processing power of the Data Integrator agent.

Delegating sessions

When a session is started on an agent with linked agents, Oracle Data Integrator determines which one of the linked agents is less loaded, and the session is delegated to this linked agent.

If the user parameter "Use pre-emptive load balancing" is in use, sessions are also rebalanced each time a session finishes. This means that if an agent runs out of sessions, it will possibly be reallocated some from another agent.

Notes:

- An agent can be linked to itself. An agent not linked to itself is only able to delegate sessions to its linked agents, and does not execute a session.
- Delegation works on cascades of linked agents. Besides, it is possible to make loops in agents links. This option is not recommended.

Agent unavailable

When for a given agent the number of running sessions is equal to its maximum number of sessions, the agent sets incoming sessions in a "queued" status until the number of running sessions falls below the maximum of sessions for this agent.

Setting up load balancing

To set up load balancing:

1. Define a set of physical agents, and link them to a root agent. More
2. Start the root and the linked agents.
3. Run the executions on the root agent. Oracle Data Integrator balances the load of the executions between its linked agents.

Note:

- The execution agent for a session can be seen in the session editor in Operator Navigator.
 - For load balancing to work between agents, it is necessary to have named the agents, that is start them with the -NAME parameter. More
-
-

Properties	Description
Linked	Indicates if the given physical agent can receive delegated sessions from the current agent.
Agent Name	Name of the physical agent that can receive sessions from the current agent.

Related Topics

"Creating a Physical Agent" in the "Setting Up a Topology" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

7.27 Physical Schema Editor

Use to define the physical schema.

The physical schema is a decomposition of the data server, allowing the datastores (tables, files, and so forth) to be classified. Objects stored in data servers with this mode of classification can be accessed by specifying the name of the schema attached to the object name.

Examples:

- Oracle classifies its tables by schema (or user). Each table is linked to a schema, thus, SCOTT.EMP represents the EMP table in the SCOTT schema.
- Microsoft Access does not have schemas.
- DB2/400 has schemas called Libraries.
- Microsoft SQL Server has a schema called Owner for each database. By default the owner is named: dbo. A table is accessed under the form HR.dbo.EMP to access the EMP table belonging to the user dbo in the HR database.

Notes:

- To access the data in a data server, all the schemas containing the datastores used in your project must be declared.
- Technologies that do not really have schemas must still undergo a default physical schema creation.

The Physical Schema Editor has the following tabs:

- [Definition](#)
- [Context](#)
- [Process Definition](#)
- [Version](#)
- [Privileges](#)
- [Flexfields](#)

7.27.1 Definition

Properties	Description
Name	Name of the physical schema, as it appears in the graphic interface. It is calculated automatically and is read-only.
Datasource (Catalog)	Only appears if the Technology selected supports Catalogs. Name of the catalog in the data server.
(Schema)	Name of the schema in the data server. Schema, owner, or library where the required data is stored. Caution: Oracle Data Integrator lists all the schemas present in the data server. Sometimes, Oracle Data Integrator cannot draw up this list. In this case, you should enter the schema name, respecting the case.
Datasource (Work Catalog)	Only appears if the Technology selected supports Catalogs. Indicate the catalog in which you want to create these objects. For some data validation or transformation operations, Oracle Data Integrator may require work objects to be created.

Properties	Description
(Work Schema)	<p>Indicate the schema in which you want to create these objects. For some data validation or transformation operations, Oracle Data Integrator may require work objects to be created.</p> <p>It is recommended that you create a specific schema dedicated to any work tables. By creating a schema named <i>SAS</i> or <i>ODI</i> in all your data servers, you ensure that all Oracle Data Integrator activity remains totally independent from your applications.</p> <p>Caution: Oracle Data Integrator lists all the schemas present in the data server. Sometimes, Oracle Data Integrator cannot draw up this list. In this case, you should enter the schema name, respecting the case.</p> <p>Note: If you do not set a Work Schema, it defaults to the Schema during the execution.</p>
Default	<p>If this check box is selected, then the physical schema is the data server default schema, when no schema has been specified. Only one physical schema can be marked by default.</p>

Work Tables Prefix

The following table details the work table prefixes that Oracle Data Integrator is likely to create in the work schema of this physical schema.

Properties	Description
Errors	Prefix used to create the tables that contain erroneous data. These tables are created or updated during a data quality control and can be consulted from the graphic interface.
Loading	Prefix used to create the objects (tables, views, files, and so forth) that allow data loading between two data servers.
Integration	Prefix used to create the objects (tables, files, and so forth) dedicated to data integration during execution of an mapping.
Temporary Indexes	Prefix used for the temporary indexes.

Journalizing Elements Prefixes

The following table details the prefixes that Oracle Data Integrator is likely to use to create elements for journalizing in this schema.

Properties	Description
Datstores	Prefix used to create the journalizing datstores (containing the change indications).
Views	Prefix used to create the views that link journalizing tables and the data tables.
Triggers	Prefix used to create triggers on the data tables that enable the journalizing tables to be updated.

Naming Rules

The following table shows how to locate and name (term) the data containers for this technology.

Properties	Description
Local Object Mask	<p>Shows how to name an object in the physical schema, when you are connected to the data server on which the object is based.</p> <p>For example, for Oracle, you should indicate %SCHEMA.%OBJECT to symbolize the syntax "SCOTT.EMP" giving access to the EMP table belonging to the user SCOTT.</p> <p>In Oracle Data Integrator, the tags available for this mask are:</p> <ul style="list-style-type: none"> ■ %CATALOG to symbolize the catalog name (property of the physical schema). ■ %SCHEMA to symbolize the schema name (property of the physical schema). ■ %OBJECT to symbolize the datastore name. <p>Caution: These tags are case-sensitive.</p>
Remote Object Mask	<p>Shows how to name an object in the physical schema, when you are connected to a different data server from the one on which this object is based.</p> <p>For example, for Oracle, you should indicate %SCHEMA.%OBJECT@DSERVER to symbolize the syntax SCOTT.EMP@NYORK allowing access to the EMP table of the user SCOTT of the instance NYORK (remote data server).</p> <p>The tags available for this mask are the same as for local objects, to which must be added %DSERVER that represents the name of the data server (property of the connection).</p>
Partition Mask	<p>Shows how to name a partition in a SELECT statement.</p> <p>In Oracle Data Integrator, the tags available for this mask are:</p> <ul style="list-style-type: none"> ■ %CATALOG to symbolize the catalog name (property of the physical schema). ■ %SCHEMA to symbolize the schema name (property of the physical schema). ■ %OBJECT to symbolize the datastore name. ■ %PARTITION to symbolize a partition name. <p>For example: %SCHEMA.%OBJECT PARTITION(%PARTITION)</p>
Sub-Partition Mask	<p>Shows how to name a sub-partition in a SELECT statement.</p> <p>In Oracle Data Integrator, the tags available for this mask are the same as for the Partition Mask.</p> <p>For example: %SCHEMA.%OBJECT SUBPARTITION(%PARTITION)</p>
Local Sequence Mask	<p>Shows the SQL code to use when partitioning a native sequence that is on the current (local) data server.</p> <p>For example: %SCHEMA.%OBJECT.nextval</p>
Remote Sequence Mask	<p>Shows the SQL code to use when partitioning a native sequence that is on the remote data server.</p> <p>For example: %SCHEMA.%OBJECT.nextval@%DSERVER</p>
Local Sequence Current Value Mask	<p>Shows the SQL code to use when accessing a native sequence current value that is on the current (local) data server.</p> <p>For example: %SCHEMA.%OBJECT.currval</p>
Remote Sequence Current Value Mask	<p>Shows the SQL code to use when accessing a native sequence current value that is on the remote data server.</p> <p>For example: %SCHEMA.%OBJECT.currval@%DSERVER</p>

Configuration

This section details the configuration related fields of the SOAP Web Service data server, which represent the mapping to a WSDL binding and enables you to specify OWSM security policies. It is only displayed when defining a physical schema for an SOAP Web Service data server.

Properties	Description
Service	Unique name that differentiates a service from the other services defined in the WSDL document.
Port	WSDL port to which you want to connect.
EndPoint URL	EndPoint URL of the service that is being invoked.
Binding	Binding WSDL port to which you want to connect.
User	User name used to connect to the service. Note: If a user name is not specified, use the user name defined at the data server level.
Password	Password linked with the user name.
Override EndPoint URL	Option to override the EndPoint URL specified in the WSDL.
OWSM Policy Configuration	Displays the OWSM security policies. You can add, edit, remove, enable, or disable OWSM security policies by clicking the corresponding toolbar icon.

7.27.2 Context

In a project, the datastores of a physical schema are always accessed by specifying the logical schema and the context.

Caution: To be able to use a physical schema in Oracle Data Integrator, it is imperative that it is associated with a physical schema in a given context.

Properties	Description
Context	Shows the contexts in which this physical schema is represented.
Logical Schema	Shows the name of the logical schema through which you can access the physical schema for the specified context. Note: If there is no suitable logical schema name in the list, entering a new name automatically creates a new logical schema.

7.27.3 Process Definition

Use process definition tab to perform Changed Data Capture in Oracle GoldenGate.

Properties	Description
Process Type	Type of the process that performs Changed Data Capture in Oracle GoldenGate. <ul style="list-style-type: none"> ■ Capture: Reads the changes from database log files and writes them to trail files, a common Oracle GoldenGate format. This process is database dependent, it needs to understand the format of log files corresponding to the database. One process per database. A capture process can extract from any number of tables the single access details in the configuration. ■ Delivery: Reads the content from a common trail file and populates the target database. This process can be database dependent or common. A delivery process can update only one database, but it can update multiple tables accessed through the same account.
Name	Name of the physical schema in Oracle Data Integrator. Process name cannot exceed 8 characters and only upper case is allowed.

Delivery Process Properties

Properties	Description
Trail File Path	Location of trail file. The file name part allows only two characters.
Discard File Path	Location of the discard file.
Definition File Path	Location of the definition file.
AllowTargetDups	Allows the same source table to be replicated to more than one target table.
ReportDetail	Enables report information to include any collision counts.
Report Count Frequency	Reports the total operations count at specific intervals. If the interval is not specified the entry is not added to the parameter file.

Capture Process Properties

Properties	Description
Trail File Path	Location of the trail file. The file name part allows only two characters.
Remote Trail File Path	Target trail file path used by the Pump process. Content from this trail file is consumed by a Delivery process.
Trail File Size	Maximum size of the trail file. If the trail file size exceeds the limit, this process creates another trail file.
Report Fetch	Enables the Oracle GoldenGate REPORTFETCH parameter.
Report Count Frequency	Reports the total operations count at specific intervals. If the interval is not specified, the entry is not added to the parameter file.

Additional Options

Properties	Description
Select a parameter	List of available parameters. Select a parameter and click Add . A template of the selected parameter is added to the text box.

Related Topics

"Creating a Physical Schema" in the "Setting Up a Topology" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

7.28 Schedule Dialog

Use to visualize the agents' scheduled tasks.

The Scheduling Information is retrieved from the agent's schedule. Note that the Agent must be started and its schedule refreshed in order to display accurate schedule information.

Properties	Description
Selected Agent	Agent for which the scheduling is displayed. You can also display the scheduling of all agents by selecting All Agents .
Selected Work Repository	Only the scenarios and load plans executed in the selected Work Repository are displayed in the schedule. Default is All Work Repositories.
Scheduling from ... to ...	Time range for which the scheduling is displayed. Click Refresh to refresh this schedule.
Update	Click Update to update the schedule for the selected agent(s)
Time Range	Allows you to center the diagram on the current time plus the time range specified (1 hour, 2 hours) here. This feature provides a vision of the sessions in progress plus the incoming sessions. You can use the arrows to move the range forward or backward.
Zoom in/out	Zooms in the Gantt diagram. Note that you can also zoom by selecting a zone in the diagram.
Gantt Diagram	Displays as a Gantt diagram the schedule for the selected agent.
Jobs details	Displays the details and execution statistics for each scheduled scenario or load plan.

Using the Diagram

If you select a zone in the diagram (keep the mouse button pressed), you automatically zoom on the selected zone.

Right-click within the diagram to open the context menu for zooming, saving the diagram as an image file, printing, or editing the display properties.

Related Topics

"Displaying the Schedule" in the "Running Integration Processes" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

7.29 Select Repository Dialog

Use to select a work repository to which you want to connect.

From the Work Repositories list, select a work repository and click **OK**.

Related Topics

"Connecting to a Work Repository" in the "Administering Repositories" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

7.30 Select Repositories Dialog

Use to select the work repositories containing the schedules you want to update.
From the work repositories list, select the Work Repositories and click **OK**.

Related Topics

"Administering Repositories" in *Oracle Fusion Middleware Administering Oracle Data Integrator*

7.31 Server Template Generation Wizard

Use to create a template to deploy a run-time agent for Oracle WebLogic Server.

How?

The Server Template Generation Wizard has the following steps:

- [Step 1 - Agent Information](#)
- [Step 2 - Libraries and Drivers](#)
- [Step 3 - Datasources](#)
- [Step 4 - Template Target and Summary](#)

7.31.1 Step 1 - Agent Information

Use to review and, if needed, to modify the agent information.

General

- **Agent Name:** Displays the name of the agent that you want to deploy.

Master Repository Connection

Datasource JNDI Name: The name of the Master Repository to which the agent is connected. Default is `jdbc/odiMasterRepository`.

Connection Retry Settings

- **Connection Retry Count:** Number of retry attempts in case the Agent cannot establish the connection to the Master Repository. Default is 0.
- **Retry Delay (milliseconds):** Interval (in milliseconds) between each attempt in case that the Agent cannot establish the connection to the Master Repository. Default is 7000.

Supervisor Authentication

Supervisor Key: Name of the key in the application server credential store that contains the login and the password of an ODI user with Supervisor privileges. This agent uses this user credentials to connect to the repository.

Click **Next** to move to the next step of the WLS Template Generation Wizard.

7.31.2 Step 2 - Libraries and Drivers

Use to select the external libraries and drivers you want to deploy with this Agent.
From the list, select the external libraries and drivers to deploy with this Agent.

The libraries can be any jar or zip file that is required for this agent. JDBC drivers or libraries for accessing the source and target data servers must be selected here.

You can use the corresponding buttons in the toolbar to select or deselect all libraries or drivers in the list.

Click **Next** to move to the next step of the Server Template.

7.31.3 Step 3 - Datasources

Use to select the datasources you want to deploy with this Agent.

From the list, select the datasources to deploy with this Agent.

Note: Only the datasources defined in the Topology for this agent are available in this list and that by default, the master connection datasource is added to this list.

Click **Next** to move to the next step of the Server Template.

7.31.4 Step 4 - Template Target and Summary

Use to specify the target location for the template generation.

Enter the target template path.

Click **Finish** to complete the WLS Template Generation Wizard and generate the WLS template in the target template path.

Related Topics

"Deploying an Agent in a Java EE Application Server" in the "Setting Up a Topology" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

7.32 Sub-language Editor

Use to create and configure sub-languages.

A sub-language is a group of language elements which all have a common type, a use within a language.

For example, the Aggregation sub-language groups all the aggregation operators in SQL language.

The Sub-language Editor has the following tabs:

- [Definition](#)
- [Version](#)
- [Privileges](#)

7.32.1 Definition

Properties	Description
Name	Name of the sub-language.

Properties	Description
Type	<p>Types of language elements in the sub-language. The possible types are:</p> <ul style="list-style-type: none"> ▪ F: Function, for example, elements accepting parameters and returning values. ▪ O: Operators, that put two words in relation. ▪ KW: Key words of the language. ▪ D: Default. ▪ DC: Dictionaries. ▪ BUIL: Built-in. ▪ FIL: Files. ▪ LS: Lists (in Jython language). ▪ MATH: Package math (in Jython language). ▪ OS: Package OS (in Jython language). ▪ SNPSFTP: Package SNPSFTP (in Jython language). ▪ SYS: Package SYS (in Jython language). ▪ TIME: Package time (in Jython language). ▪ S: Strings (in Jython language). ▪ You can specify a custom type.
Used in a Procedure	Shows that elements of the sub-language can be used in procedures.
Used in the Mapping	Shows that elements of the sub-language can be used in mapping.
Used in the From clause	Shows that elements of the sub-language can be used in FROM clauses.
Used in the Filter	Shows that elements of the sub-language can be used in filters.
Color	Color of the elements of the sub-language when displayed in the Expression Editor.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator

7.33 Technical Environment Dialog

Use to specify the parameters listed in [Table 7-1](#) for the export of the technical environment.

The export of the technical environment produces a comma separated (.csv) file in the directory of your choice, containing the details of the technical environment. This information is useful for support purposes.

Table 7-1 Technical Environment Export Parameters

Properties	Description
Export Directory	Directory in which the export file are created.
File Name	Name of the .csv export file.
Advanced options	Use these options to customize the XML output file format. It is recommended that you leave the default values.

Table 7-1 (Cont.) Technical Environment Export Parameters

Properties	Description
Character Set	Encoding specified in the export file. The encoding parameter in the XML file header. <?xml version="1.0" encoding="ISO-8859-1"?>
Field codes	The first field of each record produced contains a code identifying the kind of information present on the row. You can customize these codes as necessary. <ul style="list-style-type: none"> ■ Oracle Data Integrator Information Code: Code used to identify rows that describe the current version of Oracle Data Integrator and the current user. This code is used in the first field of the record. ■ Master, Work, Agent, and Technology Record Code: Code for rows containing information about the Master Repository, the work repositories, the running agents, or the data servers, their version, and so forth.
Record Separator and Field Separator	Separators define the characters used to separate records (lines) in the file, and fields within one record.
Text Separator	Defines the characters used to separate text within expressions, typically a quote or a double-quote character.

Click **OK** to start the export of the technical environment.

Encryption

For more information about Encryption see, [Export Key fields in Export Dialog](#)

Related Topics

"Exporting the Technical Environment" in the "Exporting and Importing" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

7.34 Technology Editor

Use to define a specific technology. In the Oracle Data Integrator terminology, this is any type of technology accessible by JDBC, ODBC, JMS, JNDI, JCA, or any operating system.

Oracle Data Integrator allows scripts to be generated and executed in the languages compatible with the technologies defined in its repository. Any type of technology can be declared and defined provided that Oracle Data Integrator can access it through appropriate middleware (JDBC, ODBC, JMS) or through an operating system command.

Example of technologies: Oracle, Sybase, Sybase IQ, DB2, Files, and so forth.

The Technology Editor has the following tabs:

- [Definition](#)
- [SQL](#)
- [Advanced](#)
- [Language](#)
- [Version](#)
- [Privileges](#)

- [Flexfields](#)

7.34.1 Definition

Properties	Description
Name	Name of the technology, as it appears in the different lists in the user interface.
Code	Technology code, which allows the technology to be referenced among the different repositories. Caution: Avoid changing this code because this could make some Oracle Data Integrator objects inconsistent if they reference a technology from another repository.
Technology type	Classification of the technology. The possible values are: <ul style="list-style-type: none"> ■ Database or files: Any technology accessible through ODBC or JDBC. Flat and XML files are part of this category, because Data Integrator includes a JDBC access driver for flat files, and a JDBC driver for XML files. ■ Operating system: Any operating system on which Oracle Data Integrator can launch commands via a Java virtual machine. ■ Topics (JMS): Any MOM accessible through JMS and including topic management (publication and subscription). ■ Queue (JMS): Any MOM accessible through JMS. ■ Oracle Data Integrator API: Technology allowing calls to the Oracle Data Integrator tools. ■ Oracle Data Integrator Connector: Technology allowing calls to a Java API. ■ Bean Scripting Framework: Technology allowing calls to a script interpreter. ■ Web Service Container: Technology allowing the deployment of web services.
Logical/Physical	Indicates if the technology is physical or logical. Three cases are possible: <ul style="list-style-type: none"> ■ Logical and Physical Technology: If a technology is both logical and physical, it can support logical and physical schemas. Its logical schemas can be mapped in the contexts exclusively on physical schemas of this technology only. ■ Logical Technology: If a technology is exclusively logical, it can support logical schemas only, which can be mapped in the contexts on physical schemas of any technology. For example: If you define an exclusively logical technology <code>GENERIC_ISO_SQL</code>, its logical schemas can give access to physical schemas based on Oracle, DB2, and other data servers. ■ Physical Technology: If a technology is exclusively physical, it can support physical schemas only, and its schemas can be accessed only through the logical schemas of an exclusively logical technology.

Data Handling

This section details the technology's data handling capacities.

Properties	Description
Select	The technology's capacity to construct a data array resulting from a query.
Where	The technology's capacity to filter this array according to conditions.
Join Support	<p>The technology's capacity to join several data arrays (two-dimensional data sets) into one only. This capacity can take several forms:</p> <ul style="list-style-type: none"> ■ If no check box is selected, the technology does not support joins. ■ If only Ordered is selected, the technology support joins in the ANSI-SQL syntax. These joins are specified in the FROM clause, as in the example below. For example: <code>FROM emp LEFT JOIN dept ON (emp.deptno = dept.deptno)</code> ■ If only Not Ordered is selected, the technology supports joins in a non-ordered syntax. These joins are typically specified in the WHERE clauses with an technology-specific join operator, as in the following example for Oracle Database: <code>WHERE emp.deptno (+) = dept.deptno</code> ■ If both Ordered and Not Ordered are selected, the technology supports both ordered and non-ordered joins. Usually, technologies supporting both syntaxes do not allow mixing them within the same statement. ■ To enable the Mixed syntax allowed option, select both Ordered and Not Ordered. If Mixed syntax allowed is selected, the technology supports mixing ordered and non-ordered joins within the same statement.
Partitioning Support	<p>The technology's capacity to support partitioning.</p> <ul style="list-style-type: none"> ■ No: The technology does not support partitioning. ■ Partitions: The technology supports partitions in SELECT statements. ■ SubPartitions: The technology supports partitions and subpartitions in SELECT statements.
Support SELECT lookup	The technology's capacity to support lookup as a SELECT statement in a SELECT clause.
Support Derived Table	The technology's capacity to support a SELECT statement in a FROM clause.
Support Set Operator	<p>The technology's capacity to support set-based operators.</p> <p>Comma-separated Operator List, lists the operators supported by the given technology. For example, for Oracle:</p> <pre>UNION, UNION ALL, MINUS, INTERSECT</pre>
Support Window Functions	The technology's capacity to support window functions in a SELECT clause, such as <code>MAX(...) OVER (PARTITION BY ... ORDER BY ...)</code> .
Constant Source Support	A boolean field. If set to <code>True</code> , the technology allows a query syntax to return a constant set of column values, without an actual table being included in the From clause. For example, Oracle technology has this field set to <code>True</code> , because Oracle supports a syntax <code>SELECT 1, 2, 3 FROM DUAL</code> , which returns a one row select result with constant column values, even though no source table is specified in the FROM clause.

Properties	Description
Constant Source Table Name	Specify a dummy table name for technologies that support constance source and use a dummy table name to specify a constant query. For example, Oracle has a dummy table name, <i>DUAL</i> , which is used for constant queries with no real source table in the FROM clause. Other technologies may support constant query with some other syntax that does not have a dummy table name, for example, by omitting the FROM clause altogether. In such a case, the Constant Source Support field will be <code>True</code> , but this field will have a value of <code>null</code> .

Naming Rules

This section shows how to locate and name the data containers for this technology.

Properties	Description
File	Shows if this technology is based on the use of files.
Using "Data server"	Shows that the technology uses a data server whose name can be used to name objects. If the check box is selected, the term used to describe the data servers for the technology must be specified. For example, in Oracle technology, the data server is accessed by an Instance/DBLink.
Using "Catalog"	Shows that the technology uses a catalog to name its datastores. If the check box is selected, the term used to describe the catalog concept in the technology must be specified. For example, Microsoft SQL Server uses a catalog named <i>Database</i> , IBM AS/400 uses a catalog named <i>Library</i> , Oracle does not use any catalog.
Using "Schema"	Shows that the technology uses a schema to name its datastores. If the box is selected, the term used to describe the schema concept in the technology must be specified. For example, Microsoft SQL Server uses a schema named <i>Owner</i> , Oracle uses a schema named <i>Schema/User</i> , Microsoft Access does not use schema.
Character Encoding	Shows the character encoding to be used while reading and writing data.
Local object mask	Defines how an object is named when you are connected to the data server on which the object is based. For example, for Oracle, you should indicate <code>%SCHEMA.%OBJECT</code> to symbolize the syntax <code>SCOTT.EMP</code> giving access to the EMP table belonging to the user SCOTT. In Oracle Data Integrator, the tags available for this mask are: <ul style="list-style-type: none"> ■ <code>%CATALOG</code> to symbolize the catalog name (property of the physical schema). ■ <code>%SCHEMA</code> to symbolize the schema name (property of the physical schema). ■ <code>%OBJECT</code> to symbolize the datastore name. Caution: These tags are case-sensitive.

Properties	Description
Remote object mask	<p>Defines how an object is named when you are connected to a different data server from the one on which the object is based.</p> <p>For example, for Oracle, you should indicate <code>%SCHEMA.%OBJECT@DSEVER</code> to symbolize the syntax <code>SCOTT.EMP@NYORK</code> allowing access to the EMP table of the user SCOTT of the instance NYORK (remote data server).</p> <p>The tags available for this mask are the same as for local objects, to which must be added <code>%DSEVER</code> representing the name of the data server (property of the connection).</p>
Partition Mask	<p>Shows how to name a partition in a SELECT statement. This field is enabled only when partitions are selected.</p> <p>In Oracle Data Integrator, the tags available for this mask are:</p> <ul style="list-style-type: none"> ■ <code>%CATALOG</code> to symbolize the catalog name (property of the physical schema). ■ <code>%SCHEMA</code> to symbolize the schema name (property of the physical schema). ■ <code>%OBJECT</code> to symbolize the datastore name. ■ <code>%PARTITION</code> to symbolize a partition name. <p>For example: <code>%SCHEMA.%OBJECT PARTITION(%PARTITION)</code></p>
Sub-Partition Mask	<p>Sub-Partition Mask shows how to name a sub-partition in a SELECT statement. This field is enabled only when Sub-Partitions is selected in Partitioning Support.</p> <p>In Oracle Data Integrator, the tags available for this mask are the same as for the Partition mask.</p> <p>For example: <code>%SCHEMA.%OBJECT SUBPARTITION(%PARTITION)</code></p>
Supports Native Sequences	<p>Technology's capacity to support native sequences. If this option is selected, you need to provide at least a value for the Local Sequence Mask.</p>
Local Sequence Mask	<p>Shows the SQL code to use when accessing a native sequence that is on the current (local) data server.</p> <p>For example: <code>%SCHEMA.%OBJECT.nextval</code></p>
Remote Sequence Mask	<p>Shows the SQL code to use when accessing a native sequence that is on the remote data server.</p> <p>For example: <code>%SCHEMA.%OBJECT.nextval@%DSEVER</code></p>
Local Sequence Current Value Mask	<p>Shows the SQL code to use when accessing a native sequence current value that is on the current (local) data server.</p> <p>For example: <code>%SCHEMA.%OBJECT.currval</code></p>
Remote Sequence Current Value Mask	<p>Shows the SQL code to use when accessing a native sequence current value that is on the remote data server.</p> <p>For example: <code>%SCHEMA.%OBJECT.currval@%DSEVER</code></p>

7.34.2 SQL

The SQL properties are reserved for technologies with data or join filtering capacities (using the WHERE clause).

Where

This section details the syntax used for filter clauses. It is only displayed for technologies with a data array filtering capacity.

Properties	Description
Order by	Shows the type of syntax used in SQL for the technology, following an ORDER BY clause. The possible values are: <ul style="list-style-type: none"> ■ Complex Expression: The clause is followed by a complex expression or the column name. ■ Column Number: The ORDER BY clause is followed by the column numbers. ■ Alias: The ORDER BY clause is followed by the column aliases.
Group by	Shows the type of syntax used in SQL for the technology, following a GROUP BY clause. The possible values are the same as for an ORDER BY clause.
Having	Shows the type of syntax used in SQL for the technology, following a HAVING clause. The possible values are the same as for an ORDER BY clause.

Ordered Joins (ISO)

This section details the syntax used for sequenced joins (SQL ISO). It is only displayed for technologies with a join sequencing capacity.

Properties	Description
Clause Location	Shows whether the join clause is located on the "FROM" level or the "WHERE" level in the query syntax.
Brackets Supported in the ON clauses	Shows whether the technology accepts sub-sets delimited by parentheses (this is the case with SQL ISO syntax).
INNER	Shows that the technology supports inner joins. The key word specifying an inner join must be given, for example, "INNER JOIN".
CROSS	Shows that the technology supports Cartesian product. The key word specifying a product set must be given, for example, "CROSS JOIN" or ",".
LEFT OUTER	Shows that the technology supports left outer joins. The key word specifying a left outer join must be given, for example, "LEFT OUTER JOIN".
RIGHT OUTER	Shows that the technology supports right outer joins. The key word specifying a right outer join must be given, for example, "RIGHT OUTER JOIN".
FULL OUTER	Shows that the technology supports full outer joins. The key word specifying a full outer join must be given, for example, "FULL OUTER JOIN".
NATURAL	Show that the technology supports natural joins. The key word specifying a natural join must be given, for example, "NATURAL JOIN".

Not Ordered Join

The parameters of non-sequenced joins allow outer join clauses to be generated in the WHERE clause.

Properties	Description
Outer Keyword	Keyword or sign imposed by the syntax as a reminder that an element is considered external. For example, for Oracle the Outer Keyword is: (+).
Outer location	Shows if the OUTER Keyword is located outside (OUTER side) or inside (INNER side) the join syntax.

Properties	Description
Clause Location	Shows if the OUTER Keyword should be placed before or after the name of the object contributing to the join.

Specific Queries

Queries used by Oracle Data Integrator to perform specific tasks. These queries are technology-specific.

The SQL query for PKs, AKs, FKs, and Indexes is defined by the following parameters:

- CATALOG (if supported by technology)
- SCHEMA (if supported by technology)
- TABLE

The SQL query for native sequences is defined with CATALOG and SCHEMA.

Properties	Description
Current Date	Query that can be sent at anytime to a data server of this technology to return the current date and time of the server. This query is used internally to recover connections.
Reverse-Engineer PKs	<p>Query used by Common Format Designer to compare the data model with the metadata from the data server. This query retrieves primary key definitions. Note that this query is not used for standard or customized reverse-engineering. This query should return a record set (one row per primary key column) that contains the following VARCHAR fields:</p> <ul style="list-style-type: none"> ■ PK_NAME: Name of the primary key. ■ COLUMN_NAME: Name of the primary key column. <p>The record set should be ordered by PK_NAME and by the position of COLUMN_NAME in the primary key.</p>
Reverse-Engineer AKs	<p>Query is used by Common Format Designer to compare the data model with the metadata from the data server. This query is not used for standard or customized reverse-engineering. This query retrieves alternate key definitions. This query should return a record set (one row per alternate key column) that contains the following VARCHAR fields:</p> <ul style="list-style-type: none"> ■ AK_NAME: Name of the alternate key. ■ COLUMN_NAME: Name of the alternate key column. <p>The record set should be ordered by AK_NAME and by the position of COLUMN_NAME in the alternate key.</p>
Reverse-Engineer FKs	<p>Query is used by Common Format Designer to compare the data model with the metadata from the data server. This query is not used for standard or customized reverse-engineering. This query retrieves foreign key definitions. This query should return a record set (one row per foreign key column) containing the following VARCHAR fields:</p> <ul style="list-style-type: none"> ■ FK_NAME: Name of the foreign key. ■ COLUMN_NAME: Name of the foreign key column. <p>The record set should be ordered by FK_NAME and by the position of COLUMN_NAME in the foreign key.</p>

Properties	Description
Reverse-Engineer Indexes	<p>Query is used by Common Format Designer to compare the data model with the metadata from the data server. This query is not used for standard or customized reverse-engineering. This query reverse-engineers indexes. This query should return a record set (one row per index column) containing the following VARCHAR fields:</p> <ul style="list-style-type: none"> ■ INDEX_NAME: Name of the index. ■ COLUMN_NAME: Name of the index column. <p>The record set should be ordered by INDEX_NAME and by the position of COLUMN_NAME in the index.</p>
Reverse-Engineer Check Constraints	<p>Query is used by Common Format Designer to compare the data model with the metadata from the data server. This query is not used for standard or customized reverse-engineering. This query reverse-engineers check constraints. This query should return a record set (one row per check constraint) containing the following VARCHAR fields:</p> <ul style="list-style-type: none"> ■ CHECK_NAME: Name of the check constraint. ■ CHECK_TXT: SQL statement of the constraint. ■ CHECK_STATUS: Status of the check constraint - '1' for enabled, '0' for disabled. ■ CHECK_DESC: Description of the check constraint.
Reverse-Engineer Native Sequences	<p>Query used to retrieve definitions for the native sequences. This query should return a record set (one row per native sequence) that contains the following VARCHAR field:</p> <ul style="list-style-type: none"> ■ SEQUENCE_NAME: Name of the native sequence. <p>The record set should be ordered by SEQUENCE_NAME.</p>
Sequence Next Value In Non-binded Mode	<p>Query used only for native sequences using #_nextval. The result of this SQL query is used in #_nextval.</p>
Sequence Current Value In Non-binded Mode	<p>Query used only for native sequences current value using #_currval. The result of this SQL query is used in #_currval.</p>

7.34.3 Advanced

Parameters

This section includes additional parameters for the technology.

Properties	Description
Column-Alias Separator	<p>Element designed to separate a column from its alias in a SQL SELECT clause.</p> <p>In the ISO standard, this separator is "AS", however, it is not supported by all databases and may therefore be left incomplete.</p>
Table-Alias Separator	<p>Element designed to separate a table from its alias in a SQL FROM clause. This element may be left incomplete.</p>
Date Function	<p>Function allowing the date and the time to be returned. For example, getdate() for Microsoft SQL Server or sysdate under Oracle</p>
String Datatype Mask	<p>The syntax used to describe Chain-type data in DDL orders (table creation). The Oracle Data Integrator tags %L (data length) and %P (data precision) can be used in the syntax description.</p>
Date Datatype Mask	<p>Syntax used to describe Date-type data in DDL orders (table creation).</p>

Properties	Description
Numerical Datatype Mask	The syntax used to describe Numerical-type data in DDL orders (table creation). The Oracle Data Integrator tags %L (data length) and %P (data precision) can be used in the syntax description.
Longvarbinary Datatype Mask	Syntax used to describe longvarbinary-type data in DDL orders. To use like Numerical Datatype Mask.
DDL Null key word	Word used to describe a column that can contain empty values (column known as NULLABLE).
Maximum column name length	Maximum length of a column name in number of characters. All column names generated by Oracle Data Integrator are truncated to this size.
Maximum table name length	Maximum length of a datastore (table) name in number of characters. All table names generated by Oracle Data Integrator are truncated to this size.

Default Table Prefixes

This section includes the prefixes that are used for table names. You can define a prefix for the following tables:

- Error Table
- Temporary Index
- Integration Table
- Loading Table
- Journal Table
- Journal View
- Journal Trigger

7.34.4 Language

This tab describes the languages implemented by the technology and allows overwriting any of the properties of the language.

Note concerning User Functions:

To make a user function available for a specific technology:

1. In the Language column, select the language of the technology.
2. Select Default.
3. Make sure that you have selected the corresponding technology from the Technology type list on the Definition tab. The Oracle Data Integrator API does not work with user functions.

Properties	Description
Language	Language that is totally or partially implemented by the technology, selected from the list of defined languages.
Compatible	Shows that the language for the technology inherits all the language elements marked as standard. For more information, see the Language Editor topic.
Default	Shows that this is the default language for this technology.

Properties	Description
Object Delimiter	<p>Character used to delimit object names. Replaces the standard delimiter given in the language.</p> <p>Caution: If the two delimiters (before and after the object name) are different, then you should use both of them. If not, one delimiter is enough.</p> <p>Use a single double-quote " to generate code like this:</p> <pre>create table MySchema."My_Table" (...</pre> <p>Use two square brackets [] to generate code like this:</p> <pre>create table MySchema.[My_Table] (...</pre>
Word Separator	<p>Characters that can be used to separate language words (objects, key words, literals). The characters declared in this list cause the object (table/schema/column) to be enclosed in the characters declared for the 'Object Delimiter'. For example, if the table name contains a # character, some databases do not accept that table name to be used, except where enclosed in double quotes: MySchema.MyTable#1 becomes MySchema.'MyTable#1' if the ' (single quote) is declared as the object delimiter and # is declared as a word separator.</p>
Literal Delimiter	<p>Character used to delimit the literal names (values). Replaces the standard delimiter given in the language.</p> <p>Caution: If the two delimiters (before and after the literal) are different, then you should use both of them. If not, one delimiter is enough.</p>
Objects case-sensitive	<p>Object names in this technology are case-sensitive.</p> <p>For example, objects are not case-sensitive in the SQL Language by default. Therefore, code like this is generated to preserve names using upper and lowercase letters:</p> <pre>create table Snps_Temp.dbo.My_Table (My_Column1 VARCHAR(20) NULL, MY_COLUMN2 VARCHAR(20) NULL)</pre> <p>If Objects case-sensitive is selected, then code like the following is generated:</p> <pre>create table Snps_Temp.dbo."My_Table" ("My_Column1" VARCHAR(20) NULL, MY_COLUMN2 VARCHAR(20) NULL)</pre>
Words case-sensitive	Keywords of this language are case-sensitive.

Related Topics

Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator, Oracle Fusion Middleware Developing Knowledge Modules with Oracle Data Integrator, and Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator

7.35 Test Connection Dialog

Use to test the connection of a data server.

Properties	Description
Physical Agent	Agent that carries out the test. Local (No Agent) indicates that the local station will attempt to connect.
Transaction (only displayed if there is any On Connect/Disconnect command defined for this data server)	Transaction on which you want to execute the command. Note that the transactions from 0 to 9 and the Autocommit transaction correspond to connection created by sessions (by procedures or knowledge modules). Client Transaction corresponds to the client components (ODI Console and Studio).

Click **Test** to start the test.

The "Connection successful!" message is displayed if the test has worked. If not, an error message is displayed. Click **Detail** to obtain more information about the cause of the connection failure.

Related Topics

"Testing a Data Server Connection" in the "Setting Up a Topology" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

7.36 Test Connection Dialog for Enterprise Scheduler

Use to test connection of the Enterprise Scheduler.

Properties	Description
Physical Agent	Agent that carries out the test. Local (No Agent) indicates that the local station will attempt to connect.

Related Topics

Oracle Fusion Middleware Administering Oracle Data Integrator

7.37 Topology Navigator

Use to work with the topology of Oracle Data Integrator.

With the Topology Navigator, you can manage the topology of your information system, the technologies and their datatypes, the data servers linked to these technologies and the schemas they contain, the contexts, the languages, and the agents. In addition, the Topology Navigator allows you to manage the repositories.

The Topology Navigator stores this information in the Master Repository. This information is used in all the other navigators.

The Topology Navigator has the following navigation trees:

- [Physical Architecture](#)
- [Contexts](#)
- [Logical Architecture](#)
- [Languages](#)
- [Repositories](#)
- [Generic Action](#)

Physical Architecture

The Physical Architecture accordion contains the technology definitions with their associated data servers and physical schemas, and the physical agents.

Tip: The list of technologies that are displayed in the Physical Architecture accordion may be very long. To narrow the list of displayed technologies, you can hide unused technologies by selecting **Hide Unused Technologies** from the Topology Navigator toolbar menu.

Contexts

The Contexts accordion contains the list of contexts.

Logical Architecture

The Logical Architecture accordion contains the technologies with their associated technology definitions, logical schemas, and the logical agents.

Languages

The Languages accordion contains the languages associated to the technologies.

Repositories

The Repositories navigation trees includes the Master Repository and the attached Work Repositories.

Generic Action

The Generic Action accordion contains the list of generic actions.

Related Topics

"Setting Up a Topology" in *Oracle Fusion Middleware Administering Oracle Data Integrator*

7.38 Work Repository Editor

Use to configure work repositories.

The Work Repository is a data structure containing information about data models, projects, and their operation. This repository is stored on a relational database accessible in client/server mode from the different Oracle Data Integrator Navigators. Several work repositories can, if necessary, be declared with several master repositories. However, a work repository can be linked with only one master repository for version management purposes.

A work repository contains several functional domains:

- **Execution:** This domain is compulsory, it allows launching and monitoring of job operations executed locally or by an agent. This domain is generally accessed through the Operator Navigator. An execution repository contains only this domain.
- **Project/Model:** This domain is optional and it allows management of data models and projects (mappings, procedures, and so forth). This domain is generally accessed through the Designer Navigator. A development repository contains this domain as well as the execution domain.

The Work Repository Editor has the following tabs:

- [Definition](#)
- [Version](#)
- [Privileges](#)

7.38.1 Definition

Properties	Description
ID	<p>ID code for the work repository.</p> <p>Important note: You must assign a unique ID code (across the whole information system) for each work repository when it is created. This unique ID code allows:</p> <ul style="list-style-type: none"> ■ Work repositories to be referenced on several master repositories without conflict. ■ Objects in this repository to be referenced on the master repository for version management (by a company domain). ■ Objects to be transferred between work repositories without conflict. Since the unique ID code for the work repository enters the composition of the ID codes of these objects, this ensures their uniqueness among repositories. <p>For greater security, we recommend creating all work repositories from the same master repository.</p>
Type	Type of work repository. The possible types are Execution or Development.
Name	Name of the work repository.
Password	Work repository password. You can modify the password by clicking Change Password .
External ID	Unique ID code of the repository.

Related Topics

"Changing a Work Repository Password" in the "Administering Repositories" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

Security Navigator Editors

Oracle Data Integrator objects are created and edited in object Editors. Oracle Data Integrator Studio provides several navigator-specific object editors. This chapter lists the Oracle Data Integrator help topics for the object editors displayed in Security Navigator.

This chapter includes the following topics:

- [Change ODI User Password Dialog](#)
- [Clean Up Security Settings Dialog](#)
- [Export Security Settings Dialog](#)
- [Import Security Settings Dialog](#)
- [Method Editor](#)
- [Object Editor](#)
- [Object Instance Editor](#)
- [Password Policies Security Settings Dialog](#)
- [Profile Editor](#)
- [Profile Method Editor](#)
- [Roles Editor](#)
- [Security Navigator](#)
- [User Editor](#)
- [User Connections Dialog](#)
- [User Method Editor](#)

8.1 Change ODI User Password Dialog

Use to change the password of the selected ODI user and to replace an expired password.

Properties	Description
New Password	Enter the new ODI user password. New password must adhere to the guidelines defined in Password Policies Security Settings Dialog .
Confirm New Password	Enter the new ODI user password.

Properties	Description
Password options	This section allows you to specify a date for this account to expire. Select Allow expiration date and enter an expiration date and time. After this date, you will need to update the password when you log in. Note: The password options can only be set for non-supervisor users as the passwords of SUPERVISOR users do not expire.

Click **OK** to apply the changes and close the dialog box.

Related Topics

Oracle Fusion Middleware Administering Oracle Data Integrator

8.2 Clean Up Security Settings Dialog

Use to clean up any security settings defined on objects which no longer exist in any repositories.

Cleanable

Properties	Description
Clean	If checked, deletes this object instance records from the master repository.
Type	Name of the ODI artifact type.
Name	Name of the object deleted from the master repository.

Non-cleanable

Properties	Description
Type	Name of the ODI artifact type.
Name	Name of the existing object in the master repository.
Status	Shows if this privilege is still effective in one or more repositories.

Related Topics

"Cleaning up Unused Authorizations" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

8.3 Export Security Settings Dialog

Use to export the security settings either as .xml files into a directory or as a zip file that contains the .xml files.

The export of the security settings includes the objects, methods, users, profiles, and password policies.

In the Export Security Settings Dialog, set the export parameters as follows:

Properties	Description
Export to directory	Directory in which the export files are created.

Properties	Description
Export to zip file	Select if you want to generate a zip file.
Zip File Name	Name of the zip file.

Encryption

For more information about Encryption, see ["Export Key fields in Export Dialog"](#)

Advanced Options

Use these options to customize the XML output file format. It is recommended that you use the default values.

Properties	Description
XML Character Set	Encoding specified in the export file. The encoding parameter in the XML file header. <?xml version="1.0" encoding="ISO-8859-1"?>
Java Character Set	Java character set used to generate the file.

Related Topics

"Export/Import Topology and Security Settings" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

8.4 Import Security Settings Dialog

Use to import the security settings.

Importing the security settings imports the objects, methods, users, profiles, passwords, and policies.

In the Import Security Settings Dialog, set the import parameters as follows:

Properties	Description
Import Mode	Select the Import Mode. Possible values are: Synonym Mode INSERT, Synonym Mode UPDATE, and Synonym Mode INSERT_UPDATE. See "Import Modes" in the <i>Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator</i> for more information.
Import From a Folder	Select if you want to import the security settings from a folder.
Import From a Zip File	Select if you want to import the security settings from a zip file.

In the text field, enter the file import folder or zip file. Note that you can browse for the import folder or zip file.

Related Topics

"Import Types" and "Export/Import Topology and Security Settings" in the "Exporting and Importing Objects" chapter in *Oracle Fusion Middleware Developing Integration Projects with Oracle Data Integrator*

8.5 Method Editor

Use to edit object methods.

A method is an action that can be performed on an object.

The Method Editor has the following tabs:

- [Definition](#)
- [Version](#)
- [Privileges](#)

8.5.1 Definition

Properties	Description
Name	Label of the method that appears in the user interface.
Internal Java name	Name of the Oracle Data Integrator internal Java method implemented on this object.
Object creation method	Indicates that this method contributes to creating an object instance.
Default method granted for the object creator	If this check box is selected, this method is automatically granted to a user who creates an object of this type. Note: This check box only takes effect for objects for which authorizations by object instance are supported.
Concerned object	Internal ID of the object.

Related Topics

"Objects, Instances, and Methods" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

8.6 Object Editor

Use to view an object definition and to add or edit flexfields for this object.

An Object is a representation of an element that can be handled through Oracle Data Integrator (agents, models, datastores, projects, and so forth).

The Object Editor has the following tabs:

- [Definition](#)
- [Flexfields](#)
- [Version](#)
- [Privileges](#)

8.6.1 Definition

Properties	Description
Name	Label of the object that appears in the user interface.

8.6.2 Flexfields

Flexfields are user-defined fields that enable you to customize the properties of Oracle Data Integrator objects. Flexfields are defined for an object type (for example, the Project object type) in the Flexfield tab of the Object Editor. The flexfield values are provided for each object instance in the Flexfield tab of the instance's editor (for example, the *Datawarehouse* Project Editor). Flexfield values can be used programatically through Oracle Data Integrator substitution methods.

Note: Flexfield exists only for certain object type. Objects that do not have a Flexfield tab do not support them.

Properties	Description
Name	Label of the flexfield as it appears in the Flexfield tab of the Object Instance Editor.
Code	Code of the flexfield.
Technology	Technology for which the flexfield will be activated. For instance, a flexfield on a procedure only appears for procedures using this technology.
Type	Type of the flexfield (String or Numeric).
Default	Default value of the flexfield.

Related Topics

"Objects, Instances, and Methods" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

8.7 Object Instance Editor

Use to manage on a specific object instance the methods of a user.

An authorization by Object Instance is granted to a user on an object instance. It allows you to grant to this user certain methods of this object instance.

The Object Instance Editor has the following tabs:

- [Security](#)
- [Version](#)

8.7.1 Security

This tab contains a list of objects and associated methods. It contains not only the methods for the instance the authorization applies to, but also the methods for the object instances found under this instance in the tree view. For example the instance MODEL1 of the Model object is inserted, then the methods for the datastores, condition, and so forth contained in this model are also displayed.

No default selection is made in the list. To perform a selection, click the lines in the table. You can select a group of items by pressing the SHIFT key, or perform multiple selections by keeping the CTRL key pressed.

Properties	Description
Object	Object type to which the privilege applies.

Properties	Description
Method	Name of the object method.
Active	This three-state check box defines how the method is granted: <ul style="list-style-type: none"> ▪ Always: It is granted for all work repositories attached to this master, even the future ones. ▪ Never: It is denied for all work repositories attached to this master, even the future ones. ▪ By Repository: It is granted for the work repositories selected in the Repository field, and denied for other repositories.
Repository	This field contains either a list of work repositories or Not defined by the user is displayed. If By Repository is selected under Active, this field contains a list of work repositories. If Always or Never is selected, Not defined by the user is displayed, meaning this privilege is allowed or denied in all repositories

8.7.2 Version

Use to view the internal ID of the object instance.

The internal ID enables the identification of the object.

- The internal ID is unique for the object type within a repository.
- The internal ID for an object type contained in more than one repository is unique because it also contains the unique repository ID.

Related Topics

"Managing Privileges" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

8.8 Password Policies Security Settings Dialog

Use to specify password rules to be used by the various users. Standard rules are provided, but other password rules can be set up.

Properties	Description
Passwords are valid for (days)	Number of days left before the password expires.
New password must be different from old one	Select to set that the new password must be different from the old one.
Invalid Login Attempts Allowed	Number of invalid login attempts allowed.

Related Topics

"Enforcing Password Policies" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

8.9 Profile Editor

Use to create and configure a profile.

A profile contains a set of privileges for working with Oracle Data Integrator. One or more profiles can be assigned to a user to grant the sum of these privileges to this user.

The Profile Editor has the following tabs:

- [Definition](#)
- [Version](#)
- [Privileges](#)

8.9.1 Definition

Properties	Description
Name	Label of the profile that appears in the user interface.

Related Topics

"Managing Profiles" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

8.10 Profile Method Editor

Use to create and configure profile methods.

A profile method is an authorization granted to a profile on a method of an object type. Each method granted allows a user with this profile to perform an action (edit, delete, and so forth) on instance of an object type (project, model, datastore, and so forth).

The Profile Method Editor has the following tabs:

- [Definition](#)
- [Version](#)
- [Privileges](#)

8.10.1 Definition

Properties	Description
Generic privilege	When the check box is selected, the user with the profile is granted, by default (automatically), a privilege on the method on all the instances of this object. When it is not selected, the user does not have access to the method. The privilege must be shown case by case by an authorization by object instance.

Related Topics

"Managing Privileges" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

8.11 Roles Editor

Use to create and configure roles.

Roles Editor provides facility to search and map both enterprise users and enterprise roles to the ODI role.

A many to many relationship exists between an ODI role and enterprise role/user. It is possible to have multiple enterprise roles or enterprise users mapped to a single role. Similarly, any enterprise role or enterprise user can be mapped to multiple ODI roles.

The Roles Editor has the following tabs:

- [Definition](#)
- [Version](#)
- [Privileges](#)
- [Flexfields](#)

8.11.1 Definition

Properties	Description
Name	Enter the name of the new role.
Notes	Enter notes about the new role.

External Authentication Principals

Properties	Description
Add principals to Role	Specifies external enterprise users and enterprise roles mapped to this ODI role. To search and add an enterprise role click on the plus sign.

Supervisor Access Privileges

Properties	Description
Supervisor	Provides the ODI supervisor privilege to the mapped enterprise role and enterprise users.

User Profiles

Properties	Description
Profile	Specifies the ODI profiles mapped to the ODI role. The combination of all the access privileges of all the profiles would be available to the ODI role.

Related Topics

"Using the Roles Editor" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

8.12 Security Navigator

Use to manage the security in Oracle Data Integrator.

The Security Navigator is used to create users and profiles, and to assign privileges to these users and profiles.

The Security Navigator stores this information in the Master Repository.

The Security Navigator has the following navigation trees:

- [Profiles](#)
- [Users](#)
- [Roles](#)
- [Objects](#)

Profiles

The Profiles navigation tree contains the list of profiles that have been defined.

Roles

The Roles navigation tree appears only if the repository is in external authentication mode. Defines a mapping between external enterprise roles to the Oracle Data Integrator profiles.

Users

The Users navigation tree contains the list of users with their profiles and privileges.

Objects

The Objects navigation tree contains the complete list of Oracle Data Integrator objects.

Related Topics

"Managing Security in Oracle Data Integrator" in *Oracle Fusion Middleware Administering Oracle Data Integrator*

8.13 User Editor

Use to create and configure users.

A user is an Oracle Data Integrator user, and corresponds to the login name used for connecting to a repository.

The User Editor has the following tabs:

- [Definition](#)
- [Authorization](#)
- [Version](#)
- [Privileges](#)
- [Flexfields](#)

8.13.1 Definition

The user properties displayed depend on the authentication method currently in use in the master repository.

User

The Definition tab contains the following fields when internal authentication is used:

Properties	Description
Name	User name, used as login name.
Notes	Enter information about the user.

Account Expiration

Properties	Description
Allow expiration date	Allows you to set an expiration date for the user.

Supervisor Access Privileges

Properties	Description
Supervisor	<p>The Supervisor grants all privileges to the user and automatically overrides any privilege grant.</p> <p>Note: Only users with Supervisor Access Privileges have access to this check box.</p> <p>Caution: The Supervisor privilege should only be granted to a minimum number of users. It is granted by default to the user SUPERVISOR created when installing Oracle Data Integrator. It is recommended to grant this privilege only on this account, which should be used for maintenance and administration purposes.</p>

User Profiles

Lists the available profiles from the Master Repository. Select one or more of these profiles to get access privileges from multiple profiles.

8.13.2 Authorization

Use to select the objects and methods to grant to the user.

Properties	Description
Objects	Lists all Objects that can be chosen to grant access to the user. Select an object to see its methods in the Methods table.
Methods	<p>Lists of all methods for the selected object.</p> <p>Method column: Name of the method.</p> <p>Assign column: If checked, user is granted access to this method.</p>

Related Topics

"Managing Users" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

8.14 User Connections Dialog

Use to view, purge, and save user activity records.

Filter Criteria

Use to enter filter criteria to display relevant user connection information.

Properties	Description
Username	Enter or select the name of the user.
Module	Enter or select the module.
Host	Enter or select the host.

Properties	Description
Work Repository	Enter or select the work repository.
Login From	Enter or select the login from date.
Login To	Enter or select the login to date.
Connection Status	Select the connection status.
Logout From	Enter or select the logout from date. Note: This field is enabled if Inactive is selected in the Connection Status drop-down list.
Logout To	Enter or select the logout to date. Note: This field is enabled if Inactive is selected in the Connection Status drop-down list.
Filter	Click to view the relevant connections.

Connections

User connections are displayed based on the set filter criteria. Using the toolbar icons you can save or delete connections, or select the columns you want displayed.

Related Topics

"Managing Security in Oracle Data Integrator" in *Oracle Fusion Middleware Administering Oracle Data Integrator*

8.15 User Method Editor

Use to view and configure user methods.

A user method is a privilege granted to a user on a method of an object type. Each method granted allows the user to perform an action (edit, delete, and so forth) on instances of an object type (project, model, datastore, and so forth).

The User Method Editor has the following tabs:

- [Definition](#)
- [Version](#)
- [Privileges](#)

8.15.1 Definition

Properties	Description
Generic privilege	When this check box is selected, the user is granted, by default (automatically), a privilege on the method on all the instances of this object. When this check box is not selected, the user may add a privilege for each instance using an authorization by object instance.

Related Topics

"Managing Privileges" in the "Managing Security in Oracle Data Integrator" chapter in *Oracle Fusion Middleware Administering Oracle Data Integrator*

