



**Mercury IT Governance Center™
Mercury Change Management Extension
for Oracle® E-Business Suite™ Guide**

Version: 6.0.0



MERCURY™

This manual, and the accompanying software and other documentation, is protected by U.S. and international copyright laws, and may be used only in accordance with the accompanying license agreement. Features of the software, and of other products and services of Mercury Interactive Corporation, may be covered by one or more of the following patents: United States: 5,511,185; 5,657,438; 5,701,139; 5,870,559; 5,958,008; 5,974,572; 6,137,782; 6,138,157; 6,144,962; 6,205,122; 6,237,006; 6,341,310; 6,360,332, 6,449,739; 6,470,383; 6,477,483; 6,549,944; 6,560,564; 6,564,342; 6,587,969; 6,631,408; 6,631,411; 6,633,912; 6,694,288; 6,738,813; 6,738,933; 6,754,701; 6,792,460 and 6,810,494. Australia: 763468 and 762554. Other patents pending. All rights reserved.

Mercury, Mercury Interactive, the Mercury logo, the Mercury Interactive logo, LoadRunner, WinRunner, SiteScope and TestDirector are trademarks of Mercury Interactive Corporation and may be registered in certain jurisdictions. The absence of a trademark from this list does not constitute a waiver of Mercury's intellectual property rights concerning that trademark.

All other company, brand and product names may be trademarks or registered trademarks of their respective holders. Mercury disclaims any responsibility for specifying which marks are owned by which companies or which organizations.

Mercury
379 North Whisman Road
Mountain View, CA 94043
Tel: (650) 603-5200
Toll Free: (800) TEST-911
Customer Support: (877) TEST-HLP
Fax: (650) 603-5300

© 1997–2005 Mercury Interactive Corporation. All rights reserved.

If you have any comments or suggestions regarding this document, please send email to documentation@mercury.com.

Table of Contents

List of Figures	xv
List of Tables	xix
Chapter 1: Introduction.....	23
About This Document.....	24
Who Should Read This Document	25
Prerequisite Documents	26
Related Documents.....	26
Overview of the Extension.....	27
General Overview of Extension for Oracle E-Business Suite	27
What's New in Release 6.0.0.....	29
Object Type Compatibility with Oracle ADADMIN Changes	29
Pre-configured Dashboard Pages	29
Improved Control of Oracle ADPATCH Utility.....	30
Improved Patch Capture	30
Reports Accessible Only from Standard Interface.....	30
Licensing.....	30
Enhanced OraApps Release 11i Upgrade Project Template.....	30
Improved OraApps Critical Requests Summary Report.....	31
Prerequisite Knowledge and Experience	31
Integration of the Extension with Oracle E-Business Suite	33
Business Concepts Used in the Oracle Environment	36
Business Roles	36
Project Manager.....	36
Project Sponsor	37
Track Owner.....	37
DBA (Database Administrator).....	38

Business Lead.....	38
Team Lead	39
Developer	39
Analyst.....	40
Additional Overview Information.....	40
Business Area.....	40
Source.....	40
Track	40
Chapter 2: Installing and Configuring the Extension.....	41
Overview of Installation.....	42
System Requirements	42
Oracle E-Business Suite Instances.....	42
Mercury IT Governance Center	45
Installing or Upgrading to Release 6.0.0.....	45
Installed Entities.....	46
Preparing for Installation.....	47
Downloading and Installing the Extension	50
Downloading the Installation Files.....	50
Installation Procedure.....	50
Installing the Extension’s Pre-configured Dashboard Pages.....	53
Logs Generated During Installation.....	55
Configuring the Extension.....	57
Configuring server.conf Parameters	57
Configuring Database Links	58
Configuring Environments.....	59
App Code Loader.....	65
Remote Execution of Object Migrator Concurrent Requests.....	66
Configuring Oracle Applications Users	70
Configuring Security Groups	71
Configuring Request Types	72
Configuring Workflows.....	74
Configuring Object Types.....	77
AOL and GL Object Types.....	77
AOL:Single Entity Object Types	78
OraApps Oracle Workflow Object Type	79
OraApps AKLOAD Migrate AK Setups Object Type	80
Deleting Installation Files.....	81
Managing Integration with Oracle E-Business Suite.....	82
Changing the Integration Method to Local	82
Changing the Integration Method to Linked.....	84
Managing Linked Integrations During Oracle E-Business Suite Downtime	86
Impact on WatchDog Functionality.....	86
Impact Depending on Oracle E-Business Suite Release.....	87

Chapter 3: Upgrading the Extension	89
Overview of Upgrade	90
Product Features Introduced in Release 5.5.0.....	91
Product Renaming.....	91
Support for Oracle E-Business Suite Release 11.5.9.....	91
Support for Oracle Patch Unified Drivers	92
Improvements to Reports.....	92
Additions to AOL Object Type Parameters.....	93
Miscellaneous Changes.....	93
Product Features Introduced in Release 6.0.0.....	93
General Upgrade Impacts and Guidelines.....	94
Impacts of Upgrading from Release 5.0.0	94
Support for Oracle Patch Unified Drivers	95
Improvements to Reports.....	96
Additions to AOL Object Type Parameters.....	97
Impacts of Upgrading from Release 5.5.0	98
Object Type Compatibility with Oracle ADADMIN Changes	98
Dashboard Menus and Tab Templates (Pre-configured Dashboard Pages).....	99
Improved Control of Oracle ADPATCH Utility.....	100
Improved Patch Capture	101
Enhanced OraApps Release 11i Upgrade Project Template.....	102
Improved OraApps Critical Requests Summary Report.....	102
Other New Functionality.....	102
Upgrade Process	102
Deleting Upgrade Files	103
Chapter 4: Extension Interface.....	105
Menus	106
Oracle Applications Menu.....	107
Issue Management Submenu	107
Project Management Submenu.....	108
Change Management Submenu.....	108
Instance Management Submenu	109
Patch Management Submenu.....	110
Oracle Applications Roles Menu.....	111
Project Sponsor Submenu	112
Project Manager Submenu	113
Track Owner Submenu	115
Apps DBA Submenu	116
Pre-configured Dashboard Pages.....	117
Overview of Pages and Portlets	117
Adding Pre-configured Dashboard Pages.....	118
Oracle Apps Project Manager Page and Its Portlets.....	119

Oracle Apps 11i: Overall Status Portlet.....	120
OraApps: Critical Activities Portlet.....	121
OraApps: Activity by Priority Portlet.....	121
Apps: Activities Summary (Past 2 Weeks) Portlet.....	122
Oracle Apps Project Sponsor Page and Its Portlets	123
Oracle Apps 11i: Overall Status Portlet.....	124
OraApps: Critical Activities Portlet.....	124
Oracle Apps 11i: All Phases Portlet.....	125
Oracle Apps Reports Track Owner Page and Its Portlets	126
Oracle Apps 11i: Reports Track Status Portlet	127
Reports: Critical Activities Portlet.....	127
Reports: Activity by Priority Portlet.....	128
Oracle Apps DBA Page and Its Portlets.....	129
Oracle Apps: Patch Deployment Status Portlet.....	130
Bugs Reported to Oracle Portlet	130
OraApps Config. Deployments Portlet.....	131
Oracle Apps: Instance Cloning Queue Portlet.....	131
Chapter 5: Managing Issues	133
Overview of Issues	134
Capturing and Tracking Issues	135
Processing User Requests.....	136
Reports About Issues	138
OraApps Critical Requests Summary Report.....	138
OraApps Apps Issues Detail Report.....	138
OraApps Apps Issues Summary Report.....	139
OraApps IT Demand Summary Report.....	139
Chapter 6: Managing Projects	141
Overview of Projects.....	142
Planning an Upgrade Strategy.....	143
Including Activities in a Project.....	143
Integrating Mercury Project Management and Demand Management.....	144
Performing a Gap Analysis	145
Using the OraApps Release 11i Upgrade Project Template.....	146
Overview of Project Template.....	146
Project Template Structure	146
Phase I: Project Startup	148
Phase II: Business Operations Analysis.....	149
Phase III: Prototype, GAP Analysis, High Level Design.....	149
Phase IV: Development and System Test	150
Phase V: Integration and Performance Testing.....	151
Phase VI: Simulations.....	151

Phase VII: Cut Over	152
Phase VIII: Post Implementation Support.....	152
Customizing Project Plans.....	153
Selecting Information to Track and Display	153
Configuring Exception Behavior	155
Configuring Project Plan Indicators.....	156
Configuring Scheduling Options	157
Configuring Security Restrictions.....	157
Reporting Status	158
OraApps Status Update Request Request Type.....	158
OraApps Status Update Request Workflow	159
Reports About Project-Related Status	159
OraApps Critical Requests Summary Report.....	160
OraApps IT Demand Summary Report	160
OraApps Ready Task List Report	161
OraApps Unassigned Ready Tasks Report.....	161
Configuring Project Management.....	162
Configuring Request Types	162
OraApps GAP Analysis Request Request Type	162
OraApps Status Update Request Request Type.....	163
Configuring Workflows.....	164
OraApps GAP Analysis Process Workflow	164
OraApps Status Update Request Workflow	165
Configuring Report Types.....	166
Chapter 7: Managing Changes	167
Overview of Changes	168
Requesting Changes.....	169
Requesting New Reports	169
Requesting New Interfaces.....	170
Requesting Conversions	171
Requesting Enhancements.....	172
OraApps Enhancement Request Type	172
OraApps Design & Development Request Type.....	173
Requesting Setup Changes.....	174
Deploying Oracle Changes.....	175
Overview of AOL and GL Migration	177
AOL or GL Migration Execution	179
AOL or GL Concurrent Request Log.....	179
Chapter 8: Managing Instances.....	181
Overview of Instances.....	182
Cloning Requirements	184

Cloning Instances	185
Cleaning Up after Cloning.....	187
Configuring Instance Management.....	188
Configuring Environments.....	189
Required Field for the Source Environment	190
Required Fields for the Destination Environment.....	191
OraApps Cloning Request Request Type	193
OraApps Cloning-Related Workflows.....	194
OraApps Cloning Process Workflow.....	194
OraApps 11i Cloning Workflow	195
OraApps 11i Cloning Object Type	196
ksc_oa_copy_clone_scripts Special Command.....	196
Chapter 9: Managing Patches.....	197
Overview of Patches.....	198
Analyzing Patch Information.....	199
OraApps Patch Detail Report	200
OraApps Patch Analysis Report.....	200
OraApps Patch Data Capture Subworkflow	201
Applying Patches.....	202
Overview of Applying Patches	202
Automating ADPATCH and ADADMIN	204
Limitations.....	205
Execution Logs.....	205
Merging Patches.....	206
Reports About Patches.....	207
OraApps Patch Detail Report	208
OraApps Patch Analysis Report	208
Patch Application Comparison Report	209
Patches Applied to an Environment Report.....	209
Pending Patches Report.....	209
Oracle Applications Implementation and Patching Scenarios.....	210
Configuring Patch Management.....	212
System Requirements	212
Configuring the OA - Oracle Patch Admin Security Group	213
Configuring Environments.....	213
Patch Stage Environment	214
Patch Destination Environment	218
SERVER_ENV_NAME Environment.....	221
Configuring Object Types.....	222
R10 Oracle Patch Object Type	222
R11 Oracle Patch Object Type	223
OraApps R11i Merge Patch Object Type	224
Object Types That Automate the Use of the ADADMIN Utility	224

Configuring Workflows.....	225
OraApps Patch Deployment Workflow.....	226
OraApps Patch Data Capture Subworkflow.....	227
OA - Capture Patch Workflow Step Source	228
Creating a Workflow to Merge Patches	228
Configuring Report Types.....	229
OraApps Patch Analysis Report	229
OraApps Patch Detail Report	229
Special Commands	230
ksc_oa_capture_patch	230
ksc_oa_purge_temp_patch.....	230
Maintaining a Mercury IT Governance System.....	230
Maintaining a Mercury IT Governance Server	231
Maintaining a Mercury IT Governance Database.....	231
Appendix A: Object Types	233
Overview of Object Types.....	234
Reference Object Types.....	234
List of Object Types	235
Application Data Migration Object Types.....	239
AOL and GL Object Types.....	240
Description.....	240
Migration Order	240
Field Definitions.....	241
AOL:Single Conc Mgr Entry Object Type.....	243
Description.....	243
Configuration Consideration	243
Field Definitions.....	244
AOL:Single GUI Menu Entry Object Type	245
Description.....	245
Configuration Consideration	245
Field Definitions.....	246
AOL:Single Report Group Unit Rel 10 Object Type	247
Description.....	247
Configuration Consideration	247
Field Definitions.....	248
AOL:Single Report Group Unit Rel 11 Object Type	249
Description.....	249
Configuration Considerations.....	249
Field Definitions.....	250
OraApps AKLOAD Migrate AK Setups Object Type	251
Description.....	251
Configuration Considerations.....	251
Field Definitions.....	253
OraApps Oracle Workflow Object Type	255

Description.....	255
Configuration Considerations.....	256
Field Definitions.....	256
Patch and Administration Object Types.....	258
R10 Oracle Patch and R11 Oracle Patch Object Types	259
Description.....	259
Field Definitions.....	259
OraApps R11i Merge Patch Object Type	263
Description.....	263
Configuration Considerations.....	263
Field Definitions.....	264
OraApps ADADMIN Compile APPS Schema Object Type	266
Description.....	266
Configuration Considerations.....	266
Field Definitions.....	267
OraApps ADADMIN Compile Flexfields Object Type	269
Description.....	269
Field Definitions.....	269
OraApps ADADMIN Generate Jar Files Object Type	271
Description.....	271
Field Definitions.....	271
OraApps ADADMIN Generate Messages Object Type	273
Description.....	273
Field Definitions.....	274
Instance Management Object Type.....	276
OraApps 11i Cloning Object Type	276
Description.....	276
Configuration Considerations.....	277
Field Definitions.....	278
Appendix B: Request Types.....	281
Overview of Request Types	283
Overview of the Request Process	284
Reference Request Types.....	285
List of Request Types.....	286
OraApps Application Issue Request Type	287
Description	287
Configuration Considerations.....	288
Field Definitions	288
OraApps Cloning Request Request Type.....	291
Description	291
Configuration Consideration	291
Field Definitions	292

OraApps Conversion Request Request Type.....	294
Description	294
Configuration Consideration	294
Field Definitions	295
OraApps Design & Development Request Type	297
Description	297
Configuration Considerations.....	298
Field Definitions	298
OraApps Enhancement Request Request Type	300
Description	300
Configuration Consideration	300
Field Definitions	301
OraApps GAP Analysis Request Request Type.....	303
Description	303
Configuration Consideration	303
Field Definitions	304
OraApps Interface Request Request Type	306
Description	306
Configuration Consideration	306
Field Definitions	307
OraApps Report Request Request Type.....	309
Description	309
Configuration Consideration	309
Field Definitions	310
OraApps Setup Change Request Request Type	312
Description	312
Configuration Consideration	312
Field Definitions	313
OraApps Status Update Request.....	315
Description	315
Configuration Considerations.....	316
Field Definitions	316
Appendix C: Workflows	319
Overview of Workflows.....	321
Reference Workflows	321
List of Workflows	321
General Configuration Considerations.....	323
OraApps Application Issue Workflow	324
Workflow Diagram and Step Descriptions.....	325
Workflow Steps with Predefined Security	328
OraApps Cloning Process Workflow	329

Workflow Diagram and Step Descriptions.....	330
OraApps Conversion Process Workflow	331
Workflow Diagram and Step Descriptions.....	332
Steps with Predefined Workflow Security	334
OraApps Customization/Configuration Deployment Workflow	335
Configuration Considerations.....	336
Workflow Diagram and Step Descriptions.....	337
Steps with Predefined Workflow Security	340
OraApps Design & Development Workflow	341
Workflow Diagram and Step Descriptions.....	342
Steps with Predefined Workflow Security	343
OraApps Enhancement Process Workflow	344
Workflow Diagram and Step Descriptions.....	345
Steps with Predefined Workflow Security	347
OraApps GAP Analysis Process Workflow.....	348
Workflow Diagram and Step Descriptions.....	349
Steps with Predefined Workflow Security	351
OraApps Interfaces Process Workflow	352
Workflow Diagram and Step Descriptions.....	353
Steps with Predefined Workflow Security	355
OraApps Patch Deployment Workflow	356
Configuration Considerations.....	357
Workflow Diagram and Step Descriptions.....	358
Steps with Predefined Workflow Security	360
OraApps Reports Process Workflow	362
Workflow Diagram and Step Descriptions.....	363
Steps with Predefined Workflow Security	365
OraApps Setup Change Process Workflow	366
Workflow Diagram and Step Descriptions.....	366
OraApps Status Update Request Workflow.....	368
Workflow Diagram and Step Descriptions.....	369
Steps with Predefined Workflow Security	371
OraApps 11i Cloning Workflow.....	373
Workflow Diagram and Step Descriptions.....	375
Appendix D: Report Types.....	379
Overview of Report Types	380
Reference Report Types.....	381
List of Report Types.....	381
General Configuration Considerations.....	382

OraApps Apps Issues Detail Report	383
Description	383
Configuration Considerations	383
Field Definitions	383
OraApps Apps Issues Summary Report	387
Description	387
Configuration Considerations	387
Field Definitions	388
OraApps Critical Requests Summary Report	391
Description	391
Configuration Considerations	391
Field Definitions	391
OraApps IT Demand Summary Report	393
Description	393
Configuration Considerations	393
Field Definitions	394
OraApps Ready Task List Report	396
Description	396
Configuration Consideration	396
Field Definitions	396
OraApps Unassigned Ready Tasks Report	398
Description	398
Configuration Consideration	398
Field Definitions	398
Patch-Related Reports	400
OraApps Patch Analysis Report	400
Description	400
Field Definitions	400
OraApps Patch Detail Report	403
Description	403
Configuration Considerations	403
Field Definitions	404
Patch Application Comparison Report	406
Description	406
Field Definitions	406
Patches Applied to an Environment Report	408
Description	408
Field Definitions	409
Pending Patches Report	410
Description	410
Field Definitions	411
Appendix E: Tokens	413
Overview of Tokens	414

Table of Contents

List of Tokens	414
Index	417

List of Figures

Figure 1-1	Local integration of Oracle E-Business Suite and the Extension	34
Figure 1-2	Linked database integration of Oracle E-Business Suite and the Extension	35
Figure 2-1	Environment window, Host tab, sample data.....	59
Figure 2-2	Environment window, Oracle Applications subtab, sample data	60
Figure 2-3	Sample setup for remote execution of Object Migrator	66
Figure 2-4	User window, Oracle Applications subtab.....	70
Figure 2-5	Workflow step configuration example	75
Figure 4-1	OraApps: Critical Activities portlet	117
Figure 4-2	Oracle Apps Project Manager page	119
Figure 4-3	Oracle Apps Project Sponsor page	123
Figure 4-4	Oracle Apps Reports Track Owner page.....	126
Figure 4-5	Oracle Apps DBA page	129
Figure 5-1	OraApps Application Issue workflow	136
Figure 6-1	OraApps Release11i Upgrade project template.....	147
Figure 7-1	Migrate Concurrent Program (AOL:Conc Prog object type) fields.....	177
Figure 7-2	Package including AOL objects.....	178
Figure 7-3	Mercury Change Management execution log sample data	180
Figure 8-1	Environment window, Host tab, sample data.....	189
Figure 8-2	Environment window, Oracle Applications subtab, sample data	190
Figure 9-1	Patch Stage Environment window, Host tab, sample data.....	216
Figure 9-2	Patch Stage Environment window, Oracle Applications subtab, sample data.....	217
Figure 9-3	Patch destination environment window, Oracle Applications subtab, sample data.....	221

List of Figures

Figure A-1	Object Type window, Ora Apps tab.....	240
Figure A-2	AOL:Conc Prog object type sample data.....	241
Figure A-3	Migrator fields from Define Concurrent Programs form in Oracle.....	242
Figure A-4	AOL:Single Conc Mgr Entry object type sample data	244
Figure A-5	AOL:Single GUI Menu Entry object type sample data.....	246
Figure A-6	AOL:Single Report Group Unit Rel 10 object type sample data.....	248
Figure A-7	AOL:Single Report Group Unit Rel 11 object type sample data.....	250
Figure A-8	OraApps AKLOAD Migrate AK Setups object type sample data.....	253
Figure A-9	OraApps Oracle Workflow object type sample data.....	256
Figure A-10	R10 Oracle Patch object type sample data	260
Figure A-11	R11 Oracle Patch object type sample data	260
Figure A-12	OraApps R11i Merge Patch object type sample data	264
Figure A-13	OraApps ADADMIN Compile APPS Schema object type sample data.....	267
Figure A-14	OraApps ADADMIN Compile Flexfields object type sample data	270
Figure A-15	OraApps ADADMIN Generate Jar Files object type sample data	272
Figure A-16	OraApps ADADMIN Generate Messages object type sample data.....	274
Figure A-17	OraApps 11i Cloning object type sample data.....	278
Figure B-1	OraApps Application Issue request creation.....	289
Figure B-2	OraApps Cloning Request creation.....	292
Figure B-3	OraApps Conversion Request creation.....	295
Figure B-4	OraApps Design & Development request creation	298
Figure B-5	OraApps Enhancement Request creation	301
Figure B-6	OraApps GAP Analysis Request creation.....	304
Figure B-7	OraApps Interface Request creation.....	307
Figure B-8	OraApps Report Request creation.....	310
Figure B-9	OraApps Setup Change Request creation	313
Figure B-10	OraApps Status Update Request creation	316
Figure C-1	OraApps Application Issue workflow	325
Figure C-2	OraApps Cloning Process workflow	330
Figure C-3	OraApps Conversion Process workflow	332
Figure C-4	OraApps Customization/Configuration Deployment workflow	337
Figure C-5	OraApps Design & Development workflow.....	342
Figure C-6	OraApps Enhancement Process workflow.....	345
Figure C-7	OraApps GAP Analysis Process workflow.....	349

Figure C-8	OraApps Interfaces Process workflow	353
Figure C-9	OraApps Patch Deployment workflow	358
Figure C-10	OraApps Patch Data Capture Subworkflow	358
Figure C-11	OraApps Reports Process workflow	363
Figure C-12	OraApps Setup Change Process workflow.....	366
Figure C-13	OraApps Status Update Request workflow.....	369
Figure C-14	OraApps 11i Cloning workflow.....	375
Figure D-1	OraApps Apps Issues Detail Report	384
Figure D-2	OraApps Apps Issues Summary Report.....	388
Figure D-3	OraApps Critical Requests Summary Report.....	392
Figure D-4	OraApps IT Demand Summary Report	394
Figure D-5	OraApps Ready Task List Report.....	397
Figure D-6	OraApps Unassigned Ready Tasks Report.....	399
Figure D-7	OraApps Patch Analysis Report	401
Figure D-8	OraApps Patch Detail Report.....	404
Figure D-9	Patch Application Comparison Report.....	407
Figure D-10	Patches Applied to an Environment Report.....	409
Figure D-11	Pending Patches Report	411

List of Tables

Table 2-1	Pre-configured Dashboard pages for the Extension and their security groups...	53
Table 2-2	Logs generated during installation.....	55
Table 2-3	Environment window, Host tab and Oracle Applications subtab field definitions.....	60
Table 2-4	Change Management request type/workflow pairs	73
Table 2-5	Variables for scripts to convert to local integration.....	84
Table 3-1	Product name changes in release 5.5.0	91
Table 4-1	Issue Management items in Oracle Applications menu.....	107
Table 4-2	Project Management items in Oracle Applications menu	108
Table 4-3	Change Management items in Oracle Applications menu.....	108
Table 4-4	Instance Management items in Oracle Applications menu	109
Table 4-5	Patch Management items in Oracle Applications menu	110
Table 4-6	Project Sponsor items in Oracle Applications Roles menu	112
Table 4-7	Project Manager items in Oracle Applications Roles menu	113
Table 4-8	Track Owner items in Oracle Applications Roles menu	115
Table 4-9	Apps DBA items in Oracle Applications Roles menu	116
Table 4-10	Oracle Apps 11i: Overall Status portlet filter fields.....	120
Table 4-11	OraApps: Critical Activities portlet filter fields.....	121
Table 4-12	OraApps: Activity by Priority portlet filter field	121
Table 4-13	Apps: Activities Summary (Past 2 Weeks) portlet filter fields	122
Table 4-14	Oracle Apps 11i: Overall Status portlet filter fields.....	124
Table 4-15	OraApps: Critical Activities portlet filter fields.....	124
Table 4-16	Oracle Apps 11i: All Phases portlet filter fields.....	125

List of Tables

Table 4-17	Oracle Apps 11i: Reports Track Status portlet filter fields.....	127
Table 4-18	Reports: Critical Activities portlet filter fields	127
Table 4-19	Reporting Track: Activity by Priority portlet filter field.....	128
Table 4-20	Oracle Apps Patches Deployment Status portlet filter fields	130
Table 4-21	Bugs Reported to Oracle portlet filter fields.....	130
Table 4-22	OraApps Config. Deployments portlet filter fields	131
Table 4-23	Oracle Apps: Instance Cloning Queue portlet filter fields	131
Table 6-1	OraApps Release11i Upgrade project template fields.....	147
Table 6-2	OraApps Release11i Upgrade project template planning fields	154
Table 6-3	OraApps Release11i Upgrade project template activity fields	154
Table 6-4	OraApps Release11i Upgrade project template exception behavior	155
Table 6-5	OraApps Release11i Upgrade project template exception rules	156
Table 6-6	OraApps Release11i Upgrade project template project plan indicators.....	156
Table 6-7	OraApps Release11i Upgrade project template scheduling options.....	157
Table 8-1	Host tab required field for source environment.....	190
Table 8-2	Host tab and Oracle Applications subtab required fields for destination environment.....	191
Table 9-1	Host tab and Oracle Applications subtab required fields for Patch Stage environment.....	215
Table 9-2	Host tab and Oracle Applications subtab required fields for patch destination environment	218
Table 9-3	OraApps ADADMIN-related object types	224
Table A-1	Object types included in the Extension.....	236
Table A-2	AOL:Single Conc Mgr Entry object type field definitions.....	244
Table A-3	AOL:Single GUI Menu Entry object type field definitions	246
Table A-4	AOL:Single Report Group Unit Rel 10 object type field definitions	248
Table A-5	AOL:Single Report Group Unit Rel 11 object type field definitions	250
Table A-6	OraApps AKLOAD Migrate AK Setups object type field definitions	254
Table A-7	OraApps Oracle Workflow object type field definitions	257
Table A-8	R10 Oracle Patch and R11 Oracle Patch object types field definitions	261
Table A-9	OraApps R11i Merge Patch object type field definitions.....	265
Table A-10	OraApps ADADMIN Compile APPS Schema object type field definitions	268
Table A-11	OraApps ADADMIN Compile Flexfields object type field definitions.....	270
Table A-12	OraApps ADADMIN Generate Jar Files object type field definitions.....	272
Table A-13	OraApps ADADMIN Generate Messages object type field definitions	275

Table A-14	OraApps 11i Cloning object type field definitions	279
Table B-1	Request types included in the Extension	286
Table B-2	OraApps Application Issue request type field definitions.....	289
Table B-3	OraApps Cloning Request request type field definitions.....	293
Table B-4	OraApps Conversion Request request type field definitions.....	296
Table B-5	OraApps Design & Development request type field definitions.....	299
Table B-6	OraApps Enhancement Request request type field definitions	302
Table B-7	OraApps GAP Analysis Request request type field definitions.....	305
Table B-8	OraApps Interface Request request type field definitions.....	308
Table B-9	OraApps Report Request request type field definitions.....	311
Table B-10	OraApps Setup Change Request request type field definitions	314
Table B-11	OraApps Status Update Request request type field definitions	317
Table C-1	Workflows included in the Extension.....	322
Table C-2	OraApps Application Issue workflow predefined security.....	328
Table C-3	OraApps Conversion Process workflow predefined security.....	334
Table C-4	OraApps Customization/Configuration workflow predefined security.....	340
Table C-5	OraApps Design & Development workflow predefined security	343
Table C-6	OraApps Enhancement Process workflow predefined security	347
Table C-7	OraApps GAP Analysis Process workflow predefined security	351
Table C-8	OraApps Interfaces Process workflow predefined security	355
Table C-9	OraApps Patch Deployment workflow predefined security.....	360
Table C-10	OraApps Reports Process workflow predefined security	365
Table C-11	OraApps Status Update Request workflow predefined security	371
Table D-1	Report types included in the Extension	381
Table D-2	OraApps Apps Issues Detail Report field definitions.....	385
Table D-3	OraApps Apps Issues Summary Report field definitions	389
Table D-4	OraApps Critical Requests Summary Report field definitions	392
Table D-5	OraApps IT Demand Summary Report field definitions.....	395
Table D-6	OraApps Ready Task List Report field definitions	397
Table D-7	OraApps Unassigned Ready Tasks Report field definitions	399
Table D-8	OraApps Patch Analysis Report field definitions.....	401
Table D-9	OraApps Patch Detail Report field definitions.....	404
Table D-10	Patch Application Comparison Report field definitions.....	407
Table D-11	Patches Applied to an Environment Report field definitions	409

List of Tables

Table D-12	Pending Patches Report field definitions.....	411
Table E-1	Tokens included in the Extension.....	415

Chapter 1 Introduction

In This Chapter:

- *About This Document*
 - *Who Should Read This Document*
 - *Prerequisite Documents*
 - *Related Documents*
 - *Overview of the Extension*
 - *General Overview of Extension for Oracle E-Business Suite*
 - *What's New in Release 6.0.0*
 - *Prerequisite Knowledge and Experience*
 - *Business Concepts Used in the Oracle Environment*
 - *Business Roles*
 - *Additional Overview Information*
-

About This Document

This document provides information about installing or upgrading to release 6.0.0 of Mercury Change Management Extension for Oracle E-Business Suite™ (often referred to hereafter as “the Extension”), and it provides conceptual, procedural, and reference information about the product.

You can access this document from the Mercury IT Governance Center™ Documentation home page and elsewhere, as described in the *Guide to Documentation*.

This document is organized as follows:

- [Chapter 1, *Introduction*, on page 23](#) (this chapter) describes who should read this document, includes information about prerequisite and related documents, and provides an overview of the Extension for this release.
- [Chapter 2, *Installing and Configuring the Extension*, on page 41](#) provides overview and detailed information about installing and configuring the Extension and managing integration with Oracle E-Business Suite.
- [Chapter 3, *Upgrading the Extension*, on page 89](#) provides information about upgrading the Extension from release 5.0.0 or release 5.5.0 to release 6.0.0.
- [Chapter 4, *Extension Interface*, on page 105](#) provides information about pre-configured Mercury IT Governance Dashboard™ pages, environment definitions, and other interface elements of the Extension.
- [Chapter 5, *Managing Issues*, on page 133](#) provides information about the Extension’s issue management functionality.
- [Chapter 6, *Managing Projects*, on page 141](#) provides information about the Extension’s project management functionality.
- [Chapter 7, *Managing Changes*, on page 167](#) provides information about the Extension’s functionality for managing changes to Oracle E-Business Suite.
- [Chapter 8, *Managing Instances*, on page 181](#) provides information about the Extension’s instance management functionality.
- [Chapter 9, *Managing Patches*, on page 197](#) provides information about the Extension’s patch management functionality.
- [Appendix A: *Object Types* on page 233](#) provides reference information about the Extension’s object types.

- [Appendix B: Request Types on page 281](#) provides reference information about the Extension's request types.
- [Appendix C: Workflows on page 319](#) provides reference information about the Extension's workflows.
- [Appendix D: Report Types on page 379](#) provides reference information about the Extension's report types.
- [Appendix E: Tokens on page 413](#) provides reference information about the Extension's tokens.

Who Should Read This Document

This document is intended for the following audiences:

- People responsible for setting up the database schema and installing, upgrading, or maintaining Mercury IT Governance Center and integrating it with one or more Oracle E-Business Suite instances
- People responsible for maintaining access and security for, or supporting use of, Mercury Change Management Extension for Oracle E-Business Suite
- People responsible for managing Oracle-related projects
- People responsible for patching activities for Oracle instances, including analysis, application and evaluation, and reporting
- Developers or others responsible for creating customizations of Oracle-related objects and promoting them to different instances
- People responsible for managing or participating in the development process or issue management process

For More Information

For more information about audience types as defined for Mercury IT Governance Center, see the *Guide to Documentation*.

For more information about business roles in connection with Oracle E-Business Suite, see [Business Roles on page 36](#).

Prerequisite Documents

You should be familiar with the information in the following documents:

- *What's New in Release 6.0*
- *Release Notes*
- *Key Concepts*
- *Getting Started*
- *System Administration Guide and Reference*
- *Mercury Change Management: Configuring a Deployment System*
- *Mercury Change Management User's Guide*
- *Security Model Guide and Reference*

Before performing an installation or upgrade, check the latest version of the *Release Notes* for release 6.0 on the Mercury IT Governance Download Center for any last-minute information or issues you need to understand.

Related Documents

See the *Guide to Documentation* for:

- Descriptions of all of the Mercury documents for Mercury IT Governance Center and Mercury Change Management Extensions, and how to access each document
- Definitions of audience types, reflecting the roles played by various personnel in using Mercury IT Governance Center and Mercury Change Management Extensions
- Descriptions of typographical conventions used in the documentation set
- Additional resources for support in using Mercury IT Governance Center and Mercury Change Management Extensions

The following information may be of particular interest:

- Additional Extension and Migrator documentation for Oracle environments:
 - *Mercury Object Migrator Guide*
 - *Mercury GL Migrator Guide*
 - *Mercury Change Management Extension for Oracle Technology Guide*

Overview of the Extension

Mercury offers prepackaged change management products to automate migrations, upgrades, and other changes in a variety of leading database, system, and application environments. These products are called Mercury Change Management Extensions.

Extensions are installed on the Mercury IT Governance Server.

General Overview of Extension for Oracle E-Business Suite

Mercury Change Management Extension for Oracle E-Business Suite helps to automate change management in the Oracle E-Business Suite environment.

The Extension enhances the functionality of Mercury Change Management™ by providing predefined entities that are unique to Oracle E-Business Suite environments, such as specialized object types, request types, workflows, report types, commands, and tokens.

For information about which Oracle releases the Extension supports, see the *System Requirements and Compatibility Matrix* document.

Organizations use Oracle E-Business Suite to manage a variety of business processes. Mercury Change Management Extension for Oracle E-Business Suite (“the Extension”) lowers costs and shortens implementation time for Oracle E-Business Suite by providing automated processes built upon known best practices for:

- Issue management, including project and ongoing operational requests
- Project management, including implementation and upgrade of Oracle E-Business Suite and gap analysis
- Change management, including, for example, managing:
 - Requests for enhancements
 - Requests for new report types
 - Requests for conversion and importation of data from third-party applications into Oracle
 - Deployment of changes
- Instance management, including instance cloning
- Patch management, including impact analysis and patch deployment

To help manage a wide range of activities related to Oracle E-Business Suite, the Extension provides entities that are unique to Oracle environments, such as specialized workflows, request types, project templates, object types, and reports (beyond Oracle E-Business Suite’s own reports).

In addition, the Mercury Object Migrator™ and Mercury GL Migrator products work in conjunction with the Extension. See the *Mercury Object Migrator Guide* and the *Mercury GL Migrator Guide*.

This manual documents release 6.0.0 of Mercury Change Management Extension for Oracle E-Business Suite.

Before using Mercury IT Governance Center release 6.0 or the Extension at release 6.0.0 for productive work, you must do one of the following:

- Install Mercury IT Governance Center release 6.0, then install Extension release 6.0.0
- Upgrade Mercury IT Governance Center to release 6.0, then upgrade the Extension to release 6.0.0

What's New in Release 6.0.0

Release 6.0.0 introduces the Extension functionality described in the following sections.

For More Information

For more information about release 6.0.0 (and Mercury IT Governance Center release 6.0), see the *What's New in Release 6.0* document.

For descriptions of supported upgrade paths, see [Overview of Upgrade on page 90](#). For general information about the impacts of upgrading from any release to release 6.0.0, see [General Upgrade Impacts and Guidelines on page 94](#). For specific information about the impacts of upgrading from release 5.0.0, see [Impacts of Upgrading from Release 5.0.0 on page 94](#). For specific information about the impacts of upgrading from release 5.5.0, see [Impacts of Upgrading from Release 5.5.0 on page 98](#).

For references to other documentation that describes Extension functionality, see [Prerequisite Documents on page 26](#) and [Related Documents on page 26](#).

Object Type Compatibility with Oracle ADADMIN Changes

To support the menu structure changes that Oracle made to the ADADMIN utility after Release 11i minipack AD.H (specifically, in the AD Post-H Consolidated 12/01 patch), the following object types have been updated in release 6.0.0:

- OraApps ADADMIN Compile APPS Schema object type
- OraApps ADADMIN Compile Flexfields object type

For More Information

For more information, see [Object Type Compatibility with Oracle ADADMIN Changes on page 98](#) in the discussion of upgrade impacts.

Pre-configured Dashboard Pages

As with Mercury IT Governance Center release 6.0, Extension-related pre-configured Dashboard pages replace the tab templates used in earlier releases and provide the same functionality.

Improved Control of Oracle ADPATCH Utility

In release 6.0.0, users can control whether the Oracle ADPATCH utility applies patches to Oracle environments when file system names and database system names are different.

For More Information

For more information, see [Improved Control of Oracle ADPATCH Utility on page 100](#) in the discussion of upgrade impacts.

Improved Patch Capture

Oracle patch data captured between the installation of Extension release 5.5.0 and any subsequent patch level or upgrade of the Extension could be incomplete, affecting reporting and analysis.

For More Information

For more information, see [Improved Patch Capture on page 101](#) in the discussion of upgrade impacts.

Reports Accessible Only from Standard Interface

As with Mercury IT Governance Center release 6.0, Extension-related reports are accessible only from the standard interface.

Licensing

Extension licensing has been updated, consistent with the new Mercury IT Governance Center release 6.0 licensing strategy. For more information, see the *Security Model Guide and Reference*.

Enhanced OraApps Release 11i Upgrade Project Template

In release 6.0.0, the OraApps Release 11i Upgrade project template supports multiple currencies and multiple calendars, consistent with the added functionality in IT Governance Center release 6.0.

Improved OraApps Critical Requests Summary Report

The OraApps Critical Requests Summary Report has two new parameters in release 6.0.0. A new reference (REFERENCE) version is provided, and any existing non-reference version with the name OraApps Critical Requests Summary is renamed.

For More Information

For more information, see [Improved OraApps Critical Requests Summary Report on page 102](#) in the discussion of upgrade impacts.

Prerequisite Knowledge and Experience

To install, upgrade, configure, or maintain release 6.0.0 of the Extension, you need to understand the following:

- Particular key concepts described in the *Key Concepts* document:
 - Change management
 - Environments
 - Mercury IT Governance Workbench
 - Object types
 - Packages
 - Portlets
 - Software deployment
 - Tokens
 - Workflows and workflow steps
 - Request types
 - Report types

- Fundamental concepts of processing concurrent requests, and administration and cloning of Oracle E-Business Suite
- Fundamental concepts governing creation, maintenance, and use of database links
- The Mercury IT Governance Center entities installed by the Extension
- The AOL or GL objects used in conjunction with the Extension

In addition, you must have practical experience in:

- Using Oracle E-Business Suite and Mercury IT Governance Center
- Installing, upgrading, configuring, and using Mercury IT Governance Center, if you are responsible for configuring the Extension

For More Information

For more information about documents and resources that help you gain the required knowledge and experience, see the *Guide to Documentation*.

Integration of the Extension with Oracle E-Business Suite

Mercury Change Management Extension for Oracle E-Business Suite, which is installed on the Mercury IT Governance Server, integrates tightly with Oracle E-Business Suite to submit and monitor Object Migrator and GL Migrator concurrent requests, as well as certain CONCSUB requests.

To be fully functional, the Extension must communicate and integrate with the instance of Oracle E-Business Suite on which Object Migrator and, optionally, GL Migrator are installed and run by default. This instance is known as the primary Object Migrator host. Values in the Mercury IT Governance Center configuration file (`server.conf`), for example `CONC_REQUEST_USER`, refer to this Oracle E-Business Suite instance.

(A remote Object Migrator host is another Oracle E-Business Suite instance on which Object Migrator and, optionally, GL Migrator are installed. For more information, see [Remote Execution of Object Migrator Concurrent Requests on page 66.](#))

There are three methods for integrating the Extension with Oracle E-Business Suite:

- Local integration (single database)
- Linked database integration
- No integration

Before beginning installation of the Extension, decide upon the integration method to be used. The integration method can be changed after installation, but changing it requires stopping and restarting the Mercury IT Governance Server. For information about changing the current integration method, see [Changing the Integration Method to Local on page 82](#) or [Changing the Integration Method to Linked on page 84](#).

The following sections describe integration methods. For information about how to install Object Migrator and how to set up database links, see the *Mercury Object Migrator Guide*.

Local Integration (Single Database)

In local integration, Mercury IT Governance Center and the primary Object Migrator host use the same database, as shown in [Figure 1-1](#).

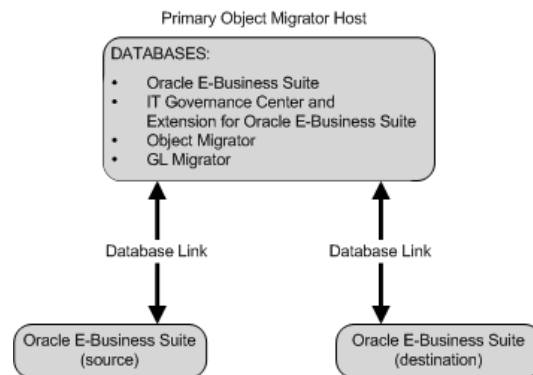


Figure 1-1. Local integration of Oracle E-Business Suite and the Extension

This configuration provides the most efficient integration of Mercury IT Governance Center with Oracle E-Business Suite, and should provide the best performance in high-volume situations. However, having two separate applications using the same database may conflict with your company's policies, or you may find that there are conflicts between the needs of Mercury IT Governance Center (for example, database patch levels or parameters) and how you want to maintain your Oracle E-Business Suite instance. In this case, Mercury recommends a different integration option.

Linked Database Integration

In linked database integration, Mercury IT Governance Center and the primary Object Migrator host use separate databases, and they communicate using database links, as shown in *Figure 1-2*.

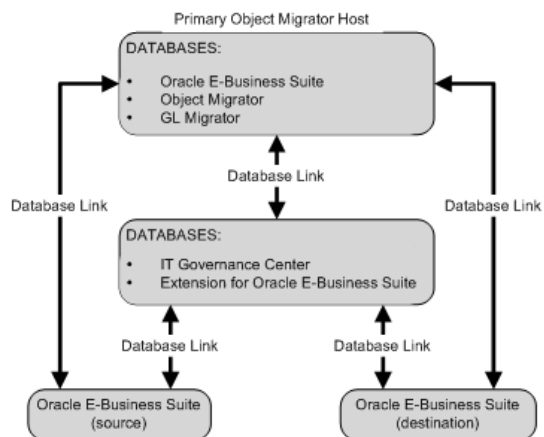


Figure 1-2. Linked database integration of Oracle E-Business Suite and the Extension

This configuration provides more flexibility than local integration, but it may be less efficient because of reliance on SQL*Net for database link connectivity. If the primary Object Migrator host is unavailable (due to downtime, network configuration issues, or other reasons), some Extension functionality (including integration with Object Migrator or GL Migrator) is unavailable. Linked integration may also require more administration to manage downtime windows (see *Managing Linked Integrations During Oracle E-Business Suite Downtime* on page 86).

No Integration

The only time you should consider not having any integration is when Object Migrator or GL Migrator are not yet installed. Without integration, some Extension functionality is unavailable. If the Extension is installed without integration and you later need to establish integration, you must define the linkage to Oracle E-Business Suite and stop and restart the Mercury IT Governance Server.



Note

These integration options do not affect the configuration or use of remote Object Migrator functionality. Use a remote Object Migrator host instead of the primary Object Migrator host as necessary to accomplish specific migrations. See *Remote Execution of Object Migrator Concurrent Requests* on page 66.

Business Concepts Used in the Oracle Environment

This section describes business roles and other business concepts that apply to the Oracle E-Business Suite environment.

Business Roles

To manage their activities, organizations operating in the Oracle E-Business Suite environment define business roles that are differentiated by their skill sets and specific responsibilities. The roles usually vary by project or implementation size. This section describes the typical business roles used in managing Oracle E-Business Suite with Mercury Change Management Extension for Oracle E-Business Suite.

Project Manager

A project manager has overall responsibility for a project, the accuracy of the project plan, and delivery of the project on time and within budget. A person in this role defines the project structure and is responsible for ensuring that adequate resources (that is, people and equipment) are available to the project at the time they are required.

The project manager actively manages project issues. If necessary, the project manager escalates issues to the project sponsor and works with appropriate IT and business resources (inside or outside of the project) to achieve resolution.

The project manager is usually a project management or IT role, and there is often more than one person involved in this activity. This person interacts both with the project sponsor and also external organizations such as operations, networking, and vendors. When using Mercury Change Management Extension for Oracle E-Business Suite, the project manager manages tasks using Mercury Project Management™, and may own all risk- or project-related issues.

Project Sponsor

A project sponsor is the overall sponsor for the project. The person in this role is usually a high-level business representative, often a director or vice president in a business area directly impacted by Oracle E-Business Suite. The project sponsor is responsible for ensuring overall company buy-in to the new or changed system, and for ensuring that any critical process or policy issues are addressed in a timely manner.

Major process changes in an organization (which usually entail changes in policy, or require some type of external reporting) usually require project sponsor approval. The project sponsor is often responsible for the overall project budget, approval of implementation cutover, and “go” or “no-go” decisions.

Project sponsors may have varying degrees of interaction with the project team, depending on project size and organizational structure, and are likely to be the primary approvers of initial project scope and changes to project scope.

Track Owner

A track owner is the overseer of activities that make up a group of related tasks on the project, such as conversions, financials, interfaces, or reporting. Track definitions are project specific. A track may involve more than one departmental business process. The person in this role is responsible for detailed project plans for the governed area, and identification and resolution of issues relating to the track.

A track owner is usually in a project management or IT role. Responsibilities include:

- Proactively providing input to the project manager
- Defining standards or templates to be used during track activities
- Coordinating cross-functional dependencies within the track
- Identifying and coordinating cross-track dependencies
- Understanding the relationship of the track's efforts to the larger project

For mid-sized projects, track owners may be responsible for design review, ownership of the code, and resource sharing across different tracks. Track owners may also prioritize and assign incoming requests to developers.

DBA (Database Administrator)

A DBA (database administrator) manages Oracle databases. This role can also have responsibility for the following activities:

- Administering the entire Oracle E-Business Suite installation, including maintenance and patching
- Determining the Oracle configuration to be used and implementing the configuration
- Ensuring the overall safety and security of the system's data, including mechanisms for backup and recovery
- Developing and enforcing performance- or data model-related coding standards for SQL or PL/SQL

Business Lead

A business lead is the representative of one or more related business processes or sectors within the company. Larger organizations may have more than one business lead.

The primary responsibility of business leads is to determine the accuracy of business requirements given to the project team, and to prioritize those business requirements, both within their own area of expertise and across the project. Business leads are directly involved in project scoping, including review and approval of scope changes and late enhancement requests. They help to ensure that a new system meets business requirements.

Business leads are usually responsible for the acceptance of functional designs, approval of test outcomes, and providing input to “go” or “no-go” decisions. They identify project risks and issues, and actively work to resolve them. They are business counterparts of track owners and work closely with them. They often contribute to or write functional test cases.

Team Lead

A team lead directs the activities of a (generally small) group of developers or analysts during various phases of the project. The team lead focuses on a relatively narrow functional area of a project such as accounts payable, EFT interfaces, or order management. Team leads are often technical experts in their areas of focus, providing guidance and help to team members as they analyze and develop solutions.

A team lead is usually responsible for technical and functional review of designs and code, and may have input to templates and standards in use by the team. This role's responsibilities may include:

- Coordinating low-level team activities and proactively identifying issues or risks
- Understanding the relationship of the team's efforts to the larger project
- Generating design documents (usually for small projects)
- Conducting gap analysis and solution proposal
- Generating accurate time estimates for proposed solutions or development activities

The distinction between track owner and team lead is based on the scale of a project. Larger projects may include separate roles for functional lead (track owner) and technical lead (team lead).

Developer

A developer is responsible for creating a particular segment of functionality (for example, a report or form). This role may include analysis tasks if a separate analyst role does not exist in the organization. Developers must be technically skilled in the development tools used on the project and have technical understanding of the data structures that are integrated with the functionality. Developers must also possess a basic functional understanding of the business process related to their area of development.

Developer responsibilities often include:

- Creating design documents
- Code development and unit testing
- Technical gap analysis and solution proposals
- Detailed effort estimates

Analyst

An analyst is responsible for translating business requirements into integrated functional definitions. This role is usually responsible for mapping requirements to Oracle functionality, and then identifying and analyzing gaps. These individuals may also provide level-of-effort estimates for development. In some organizations this role is combined with the developer role.

Additional Overview Information

This section describes business areas, sources, and tracks.

Business Area

A business area is a major business process or group in operation at a company, such as order management, finance, or shipping. The business area usually identifies a grouping of functional areas for which there is a particular business or IT owner responsible for prioritizing and managing requests. These functional areas are usually groupings of multiple, related applications. A business lead is usually responsible for one or more business areas.

Source

A source refers to the type of activity that was underway when a problem was identified. Examples of sources are integration test and production use.

Track

A track is a group of related activities within a project, such as conversions, financials, interfaces, and reporting.

Installing and Configuring the Extension

In This Chapter:

- *Overview of Installation*
 - *System Requirements*
 - *Installing or Upgrading to Release 6.0.0*
 - *Installed Entities*
- *Preparing for Installation*
- *Downloading and Installing the Extension*
 - *Downloading the Installation Files*
 - *Installation Procedure*
 - *Installing the Extension's Pre-configured Dashboard Pages*
 - *Logs Generated During Installation*
- *Configuring the Extension*
 - *Configuring server.conf Parameters*
 - *Configuring Database Links*
 - *Configuring Environments*
 - *Configuring Oracle Applications Users*
 - *Configuring Security Groups*
 - *Configuring Request Types*
 - *Configuring Workflows*
 - *Configuring Object Types*
- *Deleting Installation Files*
- *Managing Integration with Oracle E-Business Suite*
 - *Changing the Integration Method to Local*
 - *Changing the Integration Method to Linked*
 - *Managing Linked Integrations During Oracle E-Business Suite Downtime*

Overview of Installation

This section discusses the following overview subjects:

- System requirements
- Known issues
- Installing or upgrading to release 6.0.0
- Installed entities

In addition, this chapter discusses the following:

- Preparing for installation
- Downloading and installing the Extension
- Configuring the Extension
- Deleting installation files
- Managing integration of the Extension with Oracle E-Business Suite

System Requirements

Mercury IT Governance Center release 6.0 and Mercury Change Management Extension for Oracle E-Business Suite release 6.0.0 are installed on the same system and have the same system requirements. For more information about version level and other general system requirements, see *Mercury IT Governance Center on page 45* and the *System Requirements and Compatibility Matrix* document, which is available from the Mercury IT Governance Download Center:

<http://itg.merc-int.com/support/download/login.jsp>

To use the Extension, Oracle E-Business Suite environments (instances) and Mercury IT Governance Center must meet the requirements described in the following sections.

Oracle E-Business Suite Instances

This section describes system requirements for the following Oracle E-Business Suite instances:

- The primary Object Migrator host (the instance of Oracle E-Business Suite that hosts Object Migrator and GL Migrator)
- Source or destination instances of Mercury IT Governance Center activity

Primary Object Migrator Host

The primary Object Migrator host must meet the following criteria:

- The Oracle E-Business Suite release must be as specified in the *System Requirements and Compatibility Matrix* document.
- Application Object Library (AOL) must be installed and fully functional.
- If you are using GL Migrator, Oracle General Ledger must already be installed.
- For continuity of the Object Migrator and GL Migrator installations, the database must not be refreshed.
- The primary Object Migrator host can use the same database as Mercury IT Governance Center or a different database.
 - For local integration, where the databases for the primary Object Migrator host and Mercury IT Governance Center reside on the same server, both databases must meet the Mercury IT Governance Center's general database specifications. See the *System Administration Guide and Reference*.
 - For linked integration, where the databases for the primary Object Migrator host and Mercury IT Governance Center reside on different servers, the Mercury IT Governance Center schema must have a fully functional database link to the APPS account of the primary Object Migrator host.

You must decide upon the integration method to be used before you install the Extension.

For more information about integration methods, see [Integration of the Extension with Oracle E-Business Suite on page 33](#).

- Any remote Object Migrator hosts must allow connectivity using a database link from the Mercury IT Governance schema.

For more information about remote migrations, see [Remote Execution of Object Migrator Concurrent Requests on page 66](#).

- FTP or SCP connectivity must be allowed for retrieval of concurrent request logs.

Oracle E-Business Suite Instances as Source or Destination of IT Governance Activities

Oracle E-Business Suite instances that are the source or destination of activities controlled by Mercury IT Governance Center must meet the following criteria:

- The Oracle E-Business Suite release must be as specified in the *System Requirements and Compatibility Matrix* document.

Specific Extension functionality may have additional release requirements.

- You must allow connectivity using the database link from the Mercury IT Governance schema.
- You must allow “dumb terminal” login at the operating system level with a bash-compatible shell.
- To migrate Mercury workflows related to the Oracle environment, you must have a UTL_FILE_DIR definition in the database `init.ora` file.
- FTP or SCP connectivity must be provided for file transfer.
- The logon used for servers must have:
 - Read/write access to Oracle E-Business Suite directories.
 - A method to set an Oracle E-Business Suite context from the command line in a bash-compatible shell.

For information about instance cloning, see [Chapter 8, *Managing Instances*, on page 181](#).

For potential additional requirements related to Object Migrator and GL Migrator, see the following documents:

- *System Administration Guide and Reference*
- *Mercury Object Migrator Guide*
- *Mercury GL Migrator Guide*

Mercury IT Governance Center

Mercury Change Management Extension for Oracle E-Business Suite is installed on the same server as Mercury IT Governance Center. The Extension requires the following:

- Mercury IT Governance Center release 6.0 must be installed and configured.
- Connectivity must be provided to each Oracle E-Business Suite database using a database link from the Mercury IT Governance Center.
- The `open_links` database parameter (which is set in the `init.ora` file) must support the number of database links defined in the Mercury IT Governance Center schema.
- For installation or upgrade, Mercury IT Governance Server must be running in restricted mode.
- All Object Migrator and GL Migrator concurrent requests initiated from Mercury IT Governance Center must be processed to completion prior to starting a Mercury IT Governance Center upgrade.

Installing or Upgrading to Release 6.0.0

Before installing or upgrading the Extension, you must install (or upgrade to) and configure Mercury IT Governance Center release 6.0. For information about those procedures, see the *System Administration Guide and Reference*.

After you have installed Mercury IT Governance Center release 6.0, you can install Mercury Change Management Extension for Oracle E-Business Suite release 6.0.0 or upgrade to release 6.0.0 from either of the following releases:

- Mercury Change Management Extension for Oracle E-Business Suite release 5.5.0
- Kintana™ Accelerator for Oracle E-Business Suite release 5.0.0

The Mercury IT Governance Center release 6.0 upgrade process detects the existence of either release of the Extension in your environment and advises you that you will need to upgrade the Extension. [Chapter 3, *Upgrading the Extension*, on page 89](#) provides information about upgrading the Extension from either release 5.5.0 or release 5.0.0.

If you are upgrading the Extension from any release earlier than Kintana Accelerator for Oracle E-Business Suite release 5.0.0, you must first upgrade it to release 5.0.0, then upgrade it to release 6.0.0. The Mercury IT Governance Center upgrade process enforces this rule. For information about upgrades to (and features of) releases earlier than release 5.0.0, see the documentation for all of the releases from your current product release through release 5.0.0 as necessary.

Upgrade the Extension before using Mercury IT Governance Center release 6.0 for productive work; see [Chapter 3, *Upgrading the Extension*, on page 89](#).

Installed Entities

The installation process for Extension release 6.0.0 (described in [Downloading and Installing the Extension on page 50](#)) copies various entities into the Mercury database, including:

- Menus in the standard interface and pre-configured Dashboard pages that are specifically designed for Oracle E-Business Suite
- Oracle E-Business Suite tabs in the Environment, Object Type, and User windows
- Project templates
- Object types
- Report types
- Request types
- Workflows
- Security groups
- Validations

The Extension upgrade process replaces all existing Extension reference (REFERENCE) entities of the same name. For information about the impact of this process, see [General Upgrade Impacts and Guidelines on page 94](#).

Preparing for Installation

To prepare for installation of the Extension:

1. Purchase the Extension, if you have not already done so.

You will not be allowed to download the Extension installation files from the Mercury IT Governance Download Center unless you have the username and password given to you at the time of purchase.

2. Decide which Oracle E-Business Suite instance will be your primary Object Migrator host and whether it will use local integration or linked integration with Mercury IT Governance Center.

The integration method can be changed after installation, but changing it requires stopping and restarting the Mercury IT Governance Server.

The Extension can also be installed with no integration, but Object Migrator- and GL Migrator-related functions will not be available.

For more information about integration methods, see [Integration of the Extension with Oracle E-Business Suite on page 33](#).

3. Install and configure Object Migrator on the primary Object Migrator host, if it is not already installed. If the current release of Object Migrator is not installed, Mercury recommends that you upgrade to the current release before or after upgrading the Extension.

For more information about installing Object Migrator, see the *Mercury Object Migrator Guide*.

4. Install and configure GL Migrator on the primary Object Migrator host, if it is not already installed. If the current release of GL Migrator is not installed, Mercury recommends that you upgrade to the current release before or after upgrading the Extension.

For more information about installing GL Migrator, see the *Mercury GL Migrator Guide*.

5. Decide where to put the installation file (`mitg-600-OracleApps.jar`).

The installation script looks in one of these locations for this file:

- `ITG_Home` directory (this is the recommended location)
 - If the `ITG_DEPLOYMENT_HOME` environment variable is set: `ITG_Home/Deployment_Home` directory
 - If you use the `-D` option on the `sh` command to start the installation procedure: `Deployment_Home` directory
6. In the Mercury IT Governance Center schema, define a database link to the APPS account of the primary Object Migrator host, and test it.
 7. Decide whether separate tablespaces will be used for patch data captured by the Extension. If so, create them.

To reduce the potential for performance impacts or space contention, specify separate tablespaces for patch analysis and grant unlimited access on the tablespaces to the Mercury IT Governance schema.

The required size for the tablespaces depends on your expected patching activity. For example, patch detail data for the 11.5.7 maintenance pack requires 50 megabytes if no other patch data is included.

Mercury recommends initial allocations of 100 megabytes for tables and 125 megabytes for indexes.

8. Collect the following information, which you will need to supply during the installation process:

- The username and password you were given at the time you purchased the Extension
- The database password for the Mercury IT Governance Center schema
- A Mercury IT Governance Center logon username

The username (typically “admin”) must belong to a security group that has the following access grants:

- SysAdmin: Migrate ITG Objects
- SysAdmin: Server Administrator
- The password for the Mercury IT Governance Center logon username
- For local integrations, the database name and password for the Oracle E-Business Suite APPS user on the primary Object Migrator host.

For linked integrations, the name of the database link, defined in the Mercury IT Governance schema, that connects to the APPS account of the primary Object Migrator host.

For more information about integration methods, see [Integration of the Extension with Oracle E-Business Suite on page 33](#).

- (Optional) The tablespace names used by the tables and indexes created by the Extension for patch analysis

9. Stop the Mercury IT Governance Server.

10. Set the Mercury IT Governance Server to restricted mode by running the following script:

```
setServerMode.sh Restricted
```

For more information about this script, see the *System Administration Guide and Reference*.

11. Start the Mercury IT Governance Server.

Downloading and Installing the Extension

This section describes the download and installation procedures to install or upgrade the Extension.

Downloading the Installation Files

To download the installation files for Mercury Change Management Extension for Oracle Applications:

1. Go to the Mercury IT Governance Download Center:

<http://itg.merc-int.com/support/download/login.jsp>

2. Log on using the username and password your company was given when it purchased the Extension.
3. In the ITG Download > License window:
 - a. Read the Software End User License Agreement.
 - b. Select the **I accept these terms** radio button.
 - c. Click **Submit**.
4. In the ITG Download > Main window, click the **Installs** link under the name.
5. To download the Extension, click **mitg-600-OracleApps.jar**.

Installation Procedure

To install the Extension:

1. Be sure you have completed all the steps in *Preparing for Installation on page 47*.

In particular, be sure Mercury IT Governance Center is running in restricted mode. This procedure is documented in *step 10 on page 49*.

2. Be sure you have completed all the steps in *Downloading the Installation Files on page 50*.

3. Copy the `mitg-600-OracleApps.jar` file to the `ITG_Home` directory.

For information about using another location for this file, see [step 5 on page 48](#).



Note

You do not need to unpack the installation file. The installation process automatically unpacks it.

4. Navigate to the `ITG_Home/bin` directory.

5. Run the script:

```
sh kDeploy.sh -i OracleApps
```



Note

The `kDeploy` script can upgrade one Extension. For more information, see the *System Administration Guide and Reference*.

If you have upgraded Mercury IT Governance Center to release 6.0 and if one or more Mercury Change Management Extensions were previously installed on your server at the release 5.0.0 or 5.5.0 level, you can upgrade multiple Extensions by running one script—download one `.jar` file for each Extension to the `ITG_Home` directory, and run the `mitg_extensions.sh` upgrade script from the `upgrade_600` directory.

6. As `kDeploy.sh` runs, respond to prompts for the information you collected in [step 8 on page 49](#) (for example, the database password for the Mercury IT Governance Center schema, and the Mercury IT Governance Center logon name and password).

Files are installed in various subdirectories under `ITG_Home`. Data is also placed in the Mercury IT Governance database.

When the installation procedure is complete, the following message appears:

```
Deployment OracleApps has been successfully installed.
```

7. Using a Web browser, view and check the installation summary report, which is located in:

```
ITG_Home/logs/deploy/600/OracleApps/log_x/installLog.html
```

where `log_x` is a random number generated by `kDeploy.sh` to make each log file name unique. The number increments by one each time the installation script is run, so the log file for the most recent run is the one with the highest log number.

The summary report lists all Mercury entities installed as part of the Extension installation process.

Each entity that was installed correctly is marked as “Completed.” If there is an error for a particular entity, the report contains a direct link to another log file (HTML page) with additional information.

Installation of the Extension generates some or all of the logs listed and described in [Table 2-2](#), depending on the installation options.

Correct the errors, if necessary, and repeat the installation procedure. Otherwise, proceed to [step 8](#).

8. (Optional) To verify that Mercury Change Management Extension for Oracle E-Business Suite is listed among the installed release 6.0.0 Mercury Extensions, issue the following command:

```
sh kDeploy.sh -l
```

The name `OracleApps` should appear in the list of installed Mercury Extensions.

For example, if you have both Mercury Change Management Extension for Oracle E-Business Suite and Mercury Change Management Extension for PeopleSoft Enterprise™ installed, you could see a table similar to the following:

Deployment	Version	Deployed	Description
OracleApps	600	2005-02-15	Oracle Apps Extension
PeopleSoft	600	2005-03-22	PeopleSoft Extension

9. (Optional) Install other Mercury Extensions your company has purchased, if you have not already done so.

For specific installation procedures for the other Mercury Extensions, see the appropriate Extension document described in the *Guide to Documentation*.

10. Define any server configuration parameters required by the Extension.

For a list of the parameters, see [Configuring server.conf Parameters on page 57](#). For descriptions of the parameters, see the *System Administration Guide and Reference*.

11. Stop and restart the Mercury IT Governance Center in normal mode, as follows:

- a. Stop the Mercury IT Governance Server.

- b. Run the script:

```
setServerMode.sh NORMAL
```

For more information about this script, see the *System Administration Guide and Reference*.

- c. Start the Mercury IT Governance Server.

Installing the Extension's Pre-configured Dashboard Pages

You can install the Extension's pre-configured Dashboard pages (also known as modules) listed in [Table 2-1](#). Users belonging to the indicated security group can then add these pages to their Dashboards:

Table 2-1. Pre-configured Dashboard pages for the Extension and their security groups

Pre-configured Dashboard Page	Associated Security Group
Oracle Apps Project Manager	OA - Management
Oracle Apps Project Sponsor	OA - Sponsor
Oracle Apps Reports Track Owner	OA - Track Owner
Oracle Apps DBA	OA - Oracle Patch Admin

If you need to install these pre-configured Dashboard pages, perform the following steps:

1. Enable all the Extension's request types:
 - a. Open the Workbench.
 - b. From the shortcut bar, click **Demand Mgmt** and click the **Request Types** icon.
 - c. Open the request type.
 - d. In the Enabled field in the Request Type window, click **Yes**.
2. Create a copy of the (REFERENCE) OraApps Release 11i Upgrade Project Template:
 - a. From the Workbench shortcut bar, click **Project Mgmt** and click the **Project Templates** icon.
 - b. Select the (REFERENCE) OraApps Release 11i Upgrade Project Template.
 - c. Copy this project template and rename the copy:
OraApps Release 11i Upgrade Project Template
3. Install (import) the pre-configured Dashboard pages:
 - a. From the *ITG_Home/bin* directory, run the following script:

```
sh oraapps_modules.sh
```

As the script runs, respond to its prompts.
 - b. When the installation procedure is complete, the console log shows that it was successful.

Logs Generated During Installation

Depending on the installation options that were chosen, the logs listed and described in [Table 2-2](#) can be generated during installation and saved in the `ITG_Home/logs/deploy/600/OracleApps` directory. The log number (#####) is a random number (generated by `kDeploy.sh` unless otherwise noted) to make each log file name unique.

Table 2-2. Logs generated during installation

Directory or File	Description
<code>ddlDriver.#####.log</code>	Contains information about data model changes made during installation.
<code>fnd_apps_grants.#####.log</code> (Might or might not exist, depending on installation options that were chosen)	Contains information about the creation of database grants from the APPS account of a local integration to the Mercury IT Governance schema.
<code>fnd_drop_dummy.#####.log</code> (Might or might not exist, depending on installation options that were chosen)	Contains information about whether existing data structures in the Mercury IT Governance schema need to be dropped, and drops them if required. These structures are used to integrate Mercury IT Governance Center with the primary Object Migrator primary instance.
<code>fnd_knta_synonyms.#####.log</code> (Might or might not exist, depending on installation options that were chosen)	Contains information about the definition of synonyms in the Mercury IT Governance Center schema when the primary Object Migrator host shares the same database as IT Governance Center. If the connectivity to the primary Object Migrator host was linked but is now local, this log contains a message to that effect.
<code>fnd_linked_synonyms.#####.log</code>	Contains information about the refresh of connectivity to a linked primary Object Migrator host residing in a database different from Mercury IT Governance Center, using the database link provided at the start of installation. If the database link used for connectivity changes, this log contains a message to that effect.
<code>jarxvf.#####.log</code>	Contains information from the procedure that unpacks the <code>jar</code> file.
<code>packageDriver.#####.log</code>	Contains information about the installation of database code, for example, reports.
<code>postXMLDriver.#####.log</code>	Contains information about the application of SQL scripts required after the installation of Mercury-supplied data.

Table 2-2. Logs generated during installation [continued]

Directory or File	Description
preXMLDriver.#####.log	Contains information about the application of SQL scripts required prior to the installation of Mercury-supplied data, such as the definition for Mercury Change Management Extension for Oracle E-Business Suite.
recompile_invalid.#####.log	Lists any objects made invalid by prior scripts and recompiled.
modules.check.#####.log	Indicates whether or not prerequisites were met for installing the pre-configured Dashboard pages (modules) using the <code>oraapps_modules.sh</code> script, where ##### is a random number generated by this script.
log_n/installLog.html	Contains log for importing the modules if the prerequisites were met, where <i>n</i> is a sequential number defining a new subdirectory for each time the <code>oraapps_modules.sh</code> script is run.

Configuring the Extension

The Extension provides request types, workflows, and object types used for change management activities in an Oracle environment. These Mercury entities can be configured with information specific to your particular site, but they are disabled by default. Before you use them, you must enable and configure them. You must also set up database links, Mercury IT Governance Server configuration parameters, Oracle Applications users, security groups, and environments. These activities are described in the following sections.

Configuring `server.conf` Parameters

Successful use of Object Migrator and GL Migrator object types requires setting parameters in the `server.conf` file in Mercury IT Governance Center as follows:

1. Stop the Mercury IT Governance Server.
2. Set `CONC_REQUEST_USER` and `CONC_REQUEST_PASSWORD` to a user with access to the log and output directories for concurrent requests for the primary Object Migrator host.
3. Set `CONC_LOG_TRANSFER_PROTOCOL` to an allowed protocol to retrieve files from the primary Object Migrator host's concurrent request directories.

The default value is `FTP`.

4. Set `ORACLE_APPS_ENABLED` to `True`.
5. Set `SERVER_ENV_NAME` to the environment name of the IT Governance Server. The default is `KINTANA_SERVER`.
6. Restart the Mercury IT Governance Server.

For More Information

For more information about the parameters in the `server.conf` file, see the *System Administration Guide and Reference*.

Configuring Database Links

A valid database link must be defined in the Mercury IT Governance schema to the APPS account of each Oracle E-Business Suite instance that is a source or destination of Object Migrator migrations.

The Extension must integrate with the Oracle E-Business Suite instance that runs Object Migrator (and optionally GL Migrator) requests by default, that is, the primary Object Migrator host. Otherwise, users may be unable to complete the environment setups required to submit Object Migrator requests, or they may select invalid data for the setups.

The required setup is primarily at the database level, and Mercury IT Governance Server must be restarted whenever changes are made in this configuration.

For information about integration configurations, see [Integration of the Extension with Oracle E-Business Suite on page 33](#).

Configuring Environments

In the Extension, you must specify an environment for each of the following:

- The primary Object Migrator host
- Each Oracle E-Business Suite instance that is a source or destination for Object Migrator, a workflow, or an AK Web Setup migration
- An environment corresponding to the SERVER_ENV_NAME configuration parameter, to represent Mercury IT Governance server and to move log files to it
- A single common repository for all Oracle E-Business Suite patch archives. The Extension looks here to find the patch to be applied. The environment name must be Patch Stage.

A sample Environment window with the **Host** tab selected is shown in [Figure 2-1](#).

The screenshot shows the 'Environment : OA - OA9U 11i local' window. The 'Host' tab is selected, and the 'Enabled' checkbox is checked. The configuration is as follows:

Section	Field	Value
Server	Name	server.company.com
	Type	Sun Solaris
	Username	applmgr
	Password	*****
	Base Path	/export/home/applmgr
Client	Name	server.company.com
	Type	Sun Solaris
Database	Server Type	Oracle Server
	Host Name	server.company.com
	Connect String	OA9U
	Username	apps
	Password	*****
	Port Number	1528
Oracle SID	OA9U	
DB Link		
Version		
JDBC URL		

Buttons at the bottom: Check..., OK, Save, Cancel. Status: Ready.

Figure 2-1. Environment window, Host tab, sample data

A sample Environment window with the **Extension Data** tab and **Oracle Applications** subtab selected is shown in [Figure 2-2](#).

Selected fields for the **Host** tab and the **Oracle Applications** subtab are defined in [Table 2-3](#).

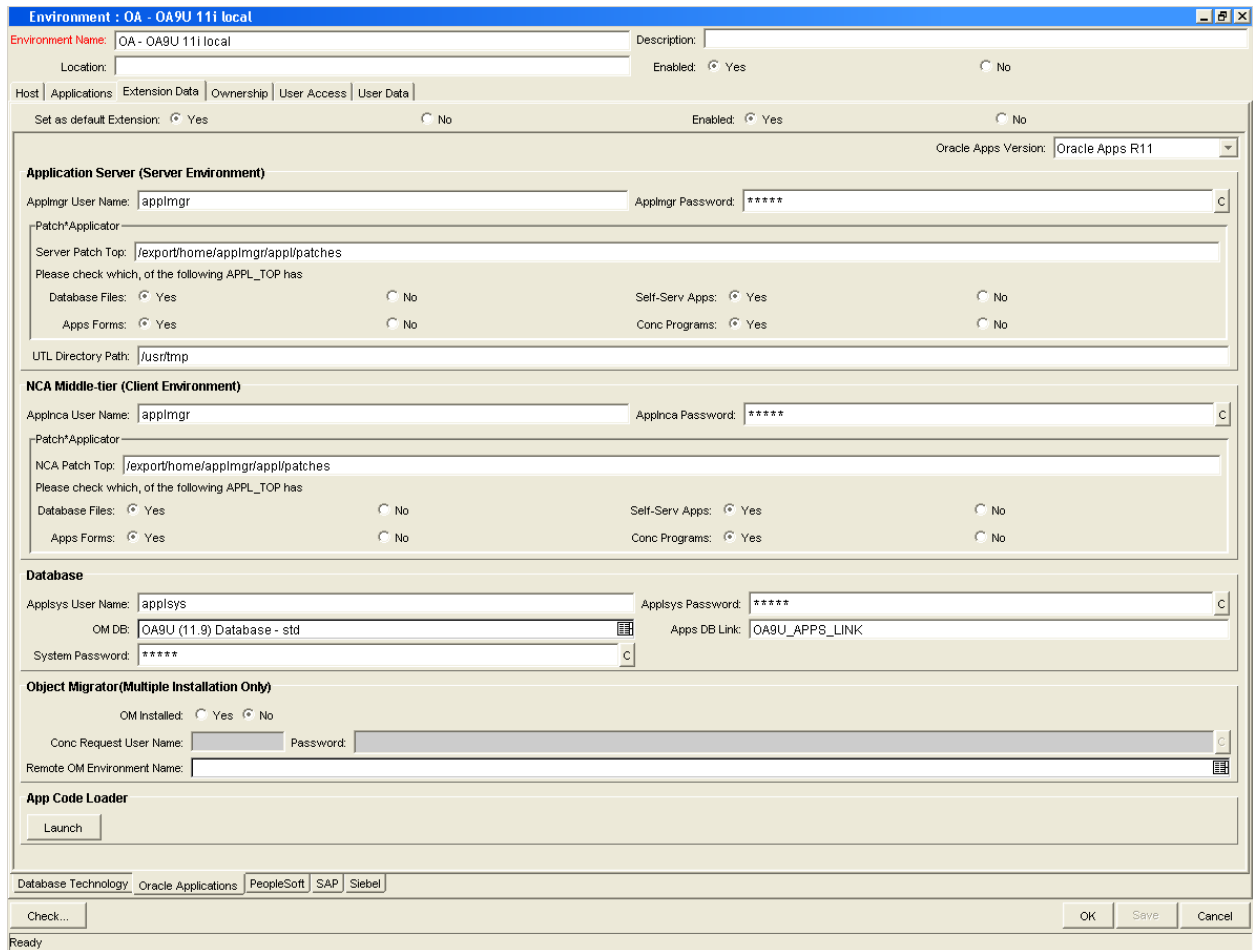


Figure 2-2. Environment window, Oracle Applications subtab, sample data

Table 2-3. Environment window, Host tab and Oracle Applications subtab field definitions

Field Name	Definition
Host tab, Server section	
Name	Host name of the server to be used for workflow and AK data migrations.
Type	Type of hardware and operating system for the host.
Username	Valid Username that has read/write access to the database UTL_FILE_DIR directory (for workflow migration) and has rights to execute WFLOAD and AKLOAD. Typically, this would be APPLMGR.
Password	Password for the Username.

Table 2-3. Environment window, Host tab and Oracle Applications subtab field definitions

Field Name	Definition
Base Path	Login directory for the user, or the APPL_TOP for the instance. The user must have read/write rights to the directory and should be able to set the Oracle context information.
Connection Protocol	Connection protocol to use when connecting to this host.
Transfer Protocol	Transfer protocol to use when moving files.
Host tab, Database section	
Server Type	Oracle Server or SQL Server . Fields in this section change according to this setting, and are described here for the Oracle Server setting.
Host Name	Host name of the server where the database runs. Used by AKLOAD to define a JDBC connection.
Connect String	Connect string for the database. Used by the OraApps Oracle Workflow object type.
Username	Database schema and user used for the APPS account. Passed to the AKLOAD and WFLOAD utilities.
Password	Password for the database user. Passed to the AKLOAD and WFLOAD utilities.
Oracle SID	SID for the database. Used by AKLOAD to define a JDBC connection. Used by the OraApps Oracle Workflow object type to set Oracle context.
Port Number	Port being monitored by the database listener. Used by AKLOAD to define a JDBC connection.
Extension Data tab, Oracle Applications subtab	
Set as default Extension	Whether or not Oracle Applications (Oracle E-Business Suite) is the default subtab to show for this window's Extension Data tab.
Enabled	Whether or not the fields on this tab are available to be configured.
Oracle Apps Version	Version of Oracle E-Business Suite running in this environment. Fields within the Oracle Applications subtab automatically change based on the selected version number. Oracle Apps R11 applies to both R11 and R11i Oracle E-Business Suite instances.
Extension Data tab, Oracle Applications subtab, Application Server (Server Environment) section: The fields in this section are used primarily by Mercury IT Governance Center during patch application and AD Administration tasks.	
Applmgr User Name	Oracle E-Business Suite file system user (usually APPLMGR) on the server host.
Applmgr Password	Password for the Oracle E-Business Suite file system user.

Table 2-3. Environment window, Host tab and Oracle Applications subtab field definitions

Field Name	Definition
Server Patch Top	Top-level directory used to store patch files for use with Oracle Patch Applicator.
Database Files**	Whether or not APPL_TOP has database files installed on the server host. This value is used during execution of ADPATCH and ADADMIN utilities on Release 11, and to determine whether patch detail information will be captured for a given Release 11 or 11i patch. Patch detail information is captured only for database tiers.
Self-Serve Apps**	Whether or not APPL_TOP has self-serve applications installed on the server host.
Apps Forms**	Whether or not APPL_TOP has forms installed on the server host.
Conc Programs**	Whether or not APPL_TOP has concurrent programs installed on the server host.
UTL Directory Path	Directory in which to place files for uploading or downloading Oracle workflows in an Oracle E-Business Suite instance. Used by the OraApps Oracle Workflow object type, it must correspond to a UTL_FILE_DIR entry in the database's <code>init.ora</code> file.
Extension Data tab, Oracle Applications subtab, NCA Middle-Tier (Client Environment) section: The fields in this section are used primarily by Mercury IT Governance Center during patch application and AD Administration tasks.	
Applnca User Name	Oracle E-Business Suite file system user (usually APPLNCA) on the middle tier host.
Applnca Password	Password for the Oracle E-Business Suite file system user on the middle tier host.
NCA Patch Top	Top-level directory used to store NCA patch files for use with Patch Applicator.
Database Files**	Whether or not APPL_TOP has database files installed on the middle tier host.
Self-Serve Apps**	Whether or not APPL_TOP has self-serve apps installed on the middle tier host
Apps Forms**	Whether or not APPL_TOP has forms installed on the middle tier host.
Conc Programs**	Whether or not APPL_TOP has concurrent programs installed on the middle tier host.

Table 2-3. Environment window, Host tab and Oracle Applications subtab field definitions

Field Name	Definition
Extension Data tab, Oracle Applications subtab, Database section: The fields in this section describe general database information for the Oracle E-Business Suite installation and are used by multiple functions including patching, AD Administration tasks, and Object Migrator.	
Applsys Username	Oracle AOL database user (usually appls). Used for patching and AD Administration tasks.
Applsys Password	Database password for the AOL user. Used for patching and AD Administration tasks.
OM DB	Object Migrator entry representing the database in this environment being specified. Retrieve the value from the CLM_DATABASES value set in the primary Object Migrator host. This field links a Mercury IT Governance Center environment definition to a source and destination defined for use with Object Migrator.
Apps DB Link	Name of the DB Link from Mercury IT Governance Center schema to the APPS schema for the database in this environment being specified, if applicable. Used for Object Migrator migration and AKLOAD verifications.
System Password	Password for the SYSTEM user on this database. Used for patching and AD Administration tasks.
Extension Data tab, Oracle Applications subtab, Object Migrator (Multiple Installation Only) section: The fields in this section are used only for remote Object Migrator functionality.	
OM Installed	See Remote Execution of Object Migrator Concurrent Requests on page 66 if you are using this configuration.
Conc Request User Name	See Remote Execution of Object Migrator Concurrent Requests on page 66 if you are using this configuration.
Password	See Remote Execution of Object Migrator Concurrent Requests on page 66 if you are using this configuration.
Remote OM Environment Name	See Remote Execution of Object Migrator Concurrent Requests on page 66 if you are using this configuration.
Extension Data tab, Oracle Applications subtab, App Code Loader section	
Launch button	Launches App Code Loader (for information about this utility, see App Code Loader on page 65).

** Field names in the table marked with a double asterisk (**) are enabled only for Oracle Apps R11 environments. The settings are used during R11 patch applications and AD Administration tasks, and are informational only for R11i. In R11i, Oracle's patch and AD Administration programs derive this information automatically.



If you use environment groups in your workflows and migrate AOL or GL information:

- The environment groups must be configured for serial execution in order for the Object Migrator or GL Migrator dependencies to be considered.
- All environments in the group must use the same Object Migrator or GL Migrator instance. For example, a mixture of remote Object Migrator and local Object Migrator executions or a mixture of different Object Migrator hosts is not allowed.

To access the **Oracle Applications** subtab on the **Extension Data** tab in the Workbench Environment window:

1. Open the Workbench.
2. From the shortcut bar, click the **Environments** button and click the **Environments** icon.
3. Open a new environment or, if the environment of interest already exists, open it. The Environment window opens.
4. Click the **Extension Data** tab.
5. Click the **Oracle Applications** subtab at the bottom of the window.

App Code Loader

The App Code Loader utility can speed up the process of configuring environments. It can be used to generate new application codes or update the server and client base paths of existing application codes in an environment.

An Oracle E-Business Suite installation typically has many different applications, for example, OE and GL. Most Oracle E-Business Suite installations include configuration files that define the file system locations of each of these applications. These files usually have extensions of `.ora` or `.env`. The files contain entries that specify the top of the application's tree. For example, the following code segment specifies that OE's home directory is located under `/u2/apps/oe/11.5.0`:

```
OE_TOP=/u2/apps/oe/11.5.0
GL_TOP=/u2/apps/gl/11.5.0
```

App Code Loader reads these files and imports this information into an environment. To launch App Code Loader, go to the **Oracle Applications** subtab of the **Extension Data** tab of an environment and click **Launch**. Detailed instructions describing how to use App Code Loader are within the utility itself.



Note

App Code Loader can be launched only when working in a saved environment. To enable the **Launch** button while entering a new environment, click **Save**.



Note

For a Windows environment, refer to the environment file that is normally in the `APPL_TOP` directory, with a name such as `SID_NAME.env`. Based on this file, create a new file that has all the variables (Product Tops) you want to load using App Code Loader, but without the string `SET`. Path specifications should include backslashes (`\`) to be consistent with Windows. An example variable is:

```
FND_TOP=D:\OraApps\appl\TestAppl\wnd\11.5.0.
```

Remote Execution of Object Migrator Concurrent Requests

The Extension knows the instance of Oracle E-Business Suite on which Object Migrator requests run by default. This instance is called the primary Object Migrator host. Package line executions in Mercury Change Management are routed as concurrent requests to the primary Object Migrator host.

When network performance or geographical distance is an issue, or when organizations within a company are required to maintain their systems independently, it can be beneficial to install Object Migrator on certain remote Oracle E-Business Suite instances also. Such instances are called remote Object Migrator hosts. Then, instead of using the primary Object Migrator host to entirely manage the migration of objects among the remote Oracle E-Business Suite instances, you can use the Remote Execution feature of the Extension to make Mercury IT Governance Center submit concurrent requests to the remote Object Migrator host, which then manages the migration remotely and more efficiently.

Figure 2-3 shows a sample setup of remote migration. (It uses local integration. For more information, see [Integration of the Extension with Oracle E-Business Suite on page 33](#).) Oracle E-Business Suite is installed on every instance shown. This example is explained in detail later.

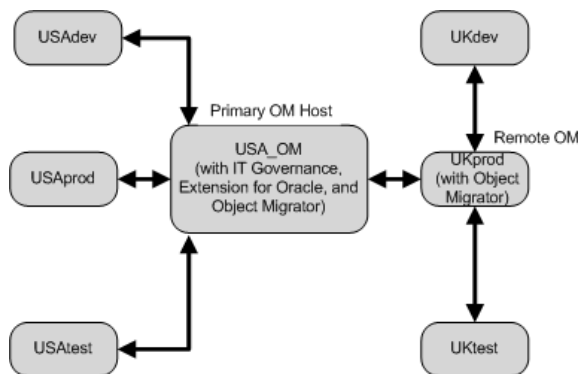


Figure 2-3. Sample setup for remote execution of Object Migrator

You must specify an environment in the Environment window (see [Figure 2-2 on page 60](#)) for the remote Object Migrator host and associated instances. If a particular environment (instance) is either a remote Object Migrator host or an instance for which a remote Object Migrator host is used to manage migrations, you must specify the fields in the Object Migrator (Multiple Installation Only) section of this window as follows:

- **OM Installed.** This field should be set to **Yes** only if this environment you are specifying is a remote Object Migrator host, that is, if it has Object Migrator installed and is intended to manage remote migrations.

This field should be set to **No** when you are specifying an environment (instance) that does not have Object Migrator installed, even if it has its migrations managed by a remote Object Migrator host, (or when the environment has Object Migrator installed but is not intended for use as a remote Object Migrator host).

In a typical remote configuration, Object Migrator is installed on only the remote Object Migrator host, and it manages migrations among several other remote Oracle E-Business Suite instances.

- **Conc Request User Name and Password.** These fields are used only when you are specifying the environment of a remote Object Migrator host, these fields are the operating system user name (usually APPLMGR) and password for a user who has permissions to FTP the concurrent request log and output files to the Mercury Change Management Server.

These fields should be left blank when you are specifying an environment (instance) that does not have Object Migrator installed, even if its migrations are managed by a remote Object Migrator host (or when the environment has Object Migrator installed but is not intended for use as a remote Object Migrator host).

- **Remote OM Environment Name.** This field is the name of the remote Object Migrator host that manages migrations for the environment you are specifying. Enter this name when you are specifying the environment of the remote Object Migrator host itself. Also, enter this name when you are specifying the environments of other remote instances for which the remote Object Migrator host manages migrations.

Note

In addition to specifying the environments, you must make sure that the CLM_DATABASES and CLM_DB_LINKS value sets in the primary Object Migrator host and all remote Object Migrator hosts are identical and have entries for all the databases associated with these Object Migrator hosts. Mercury strongly recommends that you first specify master CLM_DATABASES and CLM_DB_LINKS value sets on the primary Object Migrator host and then use the Value Set Migrator to copy them to all the remote Object Migrator hosts.

The example configuration in *Figure 2-3* includes a primary Object Migrator host in the United States (USA_OM), a remote Object Migrator host in the UK (UKprod), other instances in the US for development, production, and test, and other instances in the UK for development and test.

Using USA_OM to migrate an entity from one machine in the UK (UKdev) to another (UKtest) would take more time than using the Object Migrator installed on the remote Object Migrator host, UKprod, because latency across international networks is generally much higher than latency within a relatively local network.

Therefore, when migrating entities among any of the databases in the UK, UKprod should be used instead of the default Object Migrator, USA_OM.

To efficiently set up the remote migration example shown in *Figure 2-3*:

1. Install the same version of Object Migrator that is installed in USA_OM on UKprod.

Modify the CLM_DATABASES and CLM_DB_LINKS value sets in USA_OM to support all the instances (databases) in the USA and the UK. Then use the Value Set Migrator in Object Migrator to copy the value sets to UKprod.

2. Log on to Mercury IT Governance Center at USA_OM.
3. Open the Environment window for the UKprod environment and select the **Extension Data** tab, **Oracle Applications** subtab. (It is assumed that this environment does not yet have remote Object Migrator functionality configured, but has already been otherwise configured, including the OM DB and Apps DB Link fields.)
4. In the **Object Migrator (Multiple Installation Only)** section, set OM Installed to **Yes**.

5. Enter values for Conc Request User Name (usually **APPLMGR**) and Password. These are used to access the OUT and LOG files of the concurrent submission.
6. Click **Save** to save this environment.
7. In the Remote OM Environment Name field, select **UKprod**.
8. Click **OK** to save.
9. Open the environments of UKdev and UKtest, and perform the following steps for both environments. (It is assumed that these environments do not yet have remote Object Migrator functionality configured, but have already been otherwise configured, including the OM DB and Apps DB Link fields.)

(You can open multiple environment windows simultaneously by holding down the **Ctrl** key while selecting items.)

- a. Make sure that OM Installed is set to **No**.
- b. Set the Remote OM Environment Name field to **UKprod**.
- c. Click **OK** to save.

Now, whenever the destination is either UKdev or UKtest, UKprod will be used as the remote Object Migrator host.

In addition, you can use either the primary or remote Object Migrator host to manage migrations of objects from the USA to the UK or vice versa. For such migrations in either direction, the source and destination servers are so far apart that there is no performance advantage to using one Object Migrator host over the other.

Configuring Oracle Applications Users

The Extension allows an organization to include additional information specific to how a user accesses Oracle E-Business Suite. The Oracle Apps Username field is used when Mercury IT Governance Center submits concurrent requests, and it is reflected in concurrent request logs. [Figure 2-4](#) shows the Workbench User window, **Extension Data** tab, **Oracle Applications** subtab.

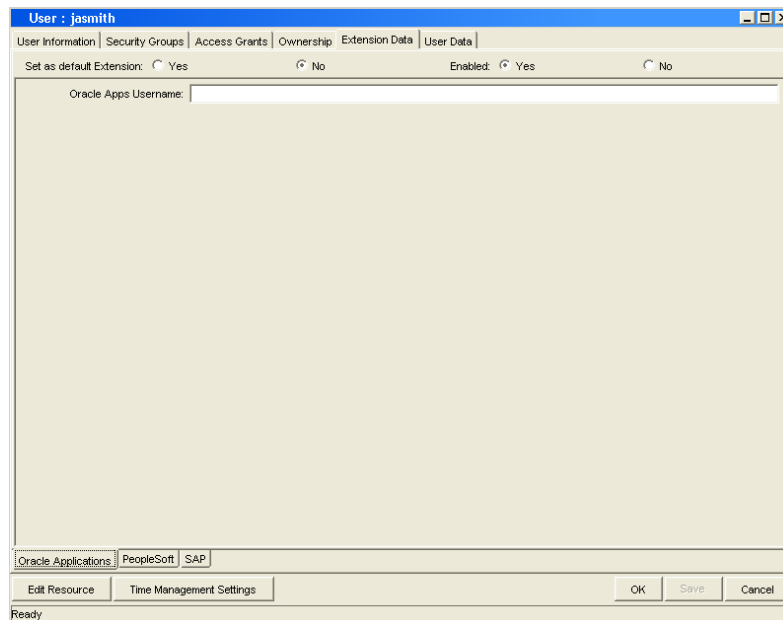


Figure 2-4. User window, Oracle Applications subtab

To access this combination of tabs:

1. Open the Workbench.
2. From the shortcut bar, click **Sys Admin** and click the **Users** icon.
3. Open a new user or, if the user of interest already exists, open it. The User window opens.
4. Click the **Extension Data** tab.
5. Click the **Oracle Applications** subtab at the bottom of the window.

For More Information

For more information about defining Mercury IT Governance Center users and privileges, see the *System Administration Guide and Reference*.

Configuring Security Groups

The Extension includes sample security groups representing common roles in organizations that use Oracle E-Business Suite. The workflows included in the Extension use these predefined Oracle-related security groups by default. You need to configure these security groups (and add any others your organization requires) for access, membership, and limitations, so they can act on workflow steps.

The predefined security groups for the Extension are:

- OA - Business Owner
- OA - Developer
- OA - Functional Designers
- OA - Functional Gap Reviewer
- OA - Management
- OA - Patch Admin
- OA - Sponsor
- OA - Technical Gap Reviewer
- OA - Track Owner

For More Information

For more information about security groups, see the *Security Model Guide and Reference*.

Configuring Request Types

The Extension includes request types containing sample values for field validations. You need to configure these validations to match the specifics of your organization. You should also assign a default workflow to each request type.

The request types for the Extension, described in detail in [Appendix B: Request Types on page 281](#), are:

- OraApps Application Issue request type
- OraApps Cloning Request request type
- OraApps Conversion Request request type
- OraApps Design & Development request type
- OraApps Enhancement Request request type
- OraApps GAP Analysis Request request type
- OraApps Interface Request request type
- OraApps Report Request request type
- OraApps Setup Change Request request type
- OraApps Status Update Request request type

To configure request types:

1. Open the Workbench.
2. From the shortcut bar, click **Demand Mgmt** and click the **Request Types** icon.
3. Open a new request type or, if the request type of interest already exists, open it. The Request Type window opens.
4. In the **Workflows** tab of the Request Type window, select the workflow that can be used with the request type.

Typically, only one workflow is attached. [Table 2-4](#) lists associated request types and workflows.

5. In the **Rules** tab of the Request Type window, select the default workflow.

[Table 2-4](#) contains the information required to make this selection.

6. (Optional) In the **Ownership** tab of the Request Type window, select the security groups allowed to modify the request type.
7. (Optional) In the **User Access** tab of the Request Type window, select the security groups allowed to create requests of this request type.

For example, you might want to do this if only some of the roles are allowed to request setup changes.

8. (Optional) If you want to restrict visibility or editability of one or more fields, define field-level security.

For example, you might want to limit edit access on analysis fields to developers and track owners.

9. Enable any Oracle request types you want to use that have not been enabled.

To enable request types:

- a. In the Workbench, open the request type.
- b. In the Enabled field in the Request Type window, click **Yes**.

Table 2-4. Change Management request type/workflow pairs

Request Type	Workflow
OraApps Design & Development	OraApps Design & Development
OraApps Enhancement	OraApps Enhancement Process
OraApps Interface	OraApps Interfaces Process
OraApps Report	OraApps Reports Process
OraApps Setup Change Request	OraApps Setup Change Process
OraApps Conversion Request	OraApps Conversion Process



In addition to the workflows listed in [Table 2-4](#), the OraApps Customization/Configuration Deployment workflow can be used at the Create Package and Wait steps of any of the request types to deploy the changes targeted to different instances.

For More Information

For more information about all of the request types, including those in [Table 2-4](#), see [Appendix B: Request Types on page 281](#).

Configuring Workflows

The Extension includes workflows that model proven real-world practices. Some workflows contain sample security information. You can configure the workflows to represent business processes for your specific environment. You also need to confirm or modify security and notification information.

The workflows for the Extension, described in detail in [Appendix C: Workflows on page 319](#), are:

- OraApps Application Issue workflow
- OraApps Cloning Process workflow
- OraApps Conversion Process workflow
- OraApps Customization/Configuration Deployment workflow
- OraApps Design & Development workflow
- OraApps Enhancement Process workflow
- OraApps GAP Analysis Process workflow
- OraApps Interfaces Process workflow
- OraApps Patch Deployment workflow
- OraApps Reports Process workflow
- OraApps Setup Change Process workflow
- OraApps Status Update Request workflow
- OraApps 11i Cloning workflow

[Figure 2-5](#) shows a sample workflow step configuration window.

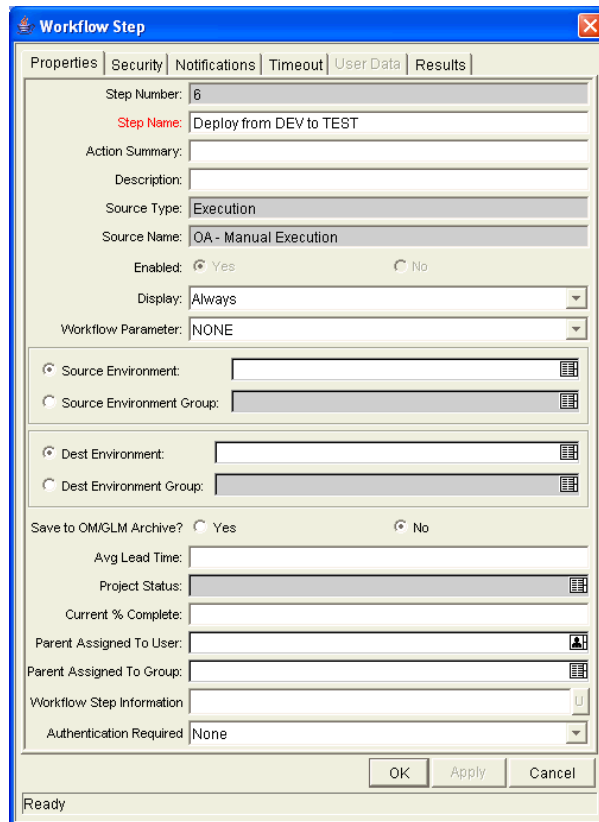


Figure 2-5. Workflow step configuration example

To configure an Oracle-specific workflow:

1. Open the Workbench.
2. From the shortcut bar, click **Configuration** and click the **Workflows** icon.
3. Open a new workflow or, if the workflow of interest already exists, open it. The Workflow window opens.
4. In the **Layout** tab of the Workflow window, right-click an execution step and click **Edit**.
5. Assign security groups to the workflow steps to make sure that only the appropriate set of users can act on a given step.

For example, you would probably want to limit which users are permitted to act on the Sign-Off Functional Design step to only those users authorized to approve functional designs.

The Extension includes predefined security groups on some workflow steps.

6. To prevent errant migrations, limit the object types that can be used with a particular workflow.

The Mercury Change Management object types can be used with any workflow that specifies valid source and destination environments. The OraApps Customization/Configuration Deployment workflow is specifically designed to handle these entities integrated with other activities.

7. Modify the workflow to reflect the Oracle E-Business Suite instances in use.

Set source and destination environments on the workflow execution steps. The OM DB fields from source and destination environments are used as the source and destination fields to Object Migrator or GL Migrator.

8. Use Mercury Object Archive when using Mercury Change Management to run the Object Migrator or GL Migrator. You can do this in one of the following ways:

- Create an environment (with a name such as Archive). In the **Extension Data** tab, **Oracle Applications** subtab, set the OM DB field to **Object Archive**.
- Set the Save to OM/GLM Archive field to **Yes** when defining your workflow migration step. Then Object Migrator and GL Migrator will both migrate to your destination environment and save to the Mercury Object Archive.

In order to save to the Mercury Object Archive, you must provide a value for the Version Label field on the package line Object Migrator or GL Migrator entry. Because the same entity cannot be archived twice in the same version, no more than one step on your workflow should have the Save to OM/GLM Archive field set to **Yes**.

9. Verify that the target workflow has been enabled.

For More Information

For more information about workflows, see [Appendix C: Workflows on page 319](#).

Configuring Object Types

The Extension includes many predefined object types that may need to be configured to reflect special logic required for your organization's migrations, such as preprocessing activities, integration with source control, and setup of the execution environment.

Extension-specific object types are categorized as follows and are described in detail in [Appendix A: *Object Types* on page 233](#):

- Application data migration object types, which are discussed in this section under the subcategories:
 - AOL and GL object types
 - AOL:Single *Entity* object types
 - OraApps Oracle Workflow object type
 - OraApps AKLOAD Migrate AK Setups object type
- Application patching and administration object types, which are discussed in [Patch and Administration Object Types on page 258](#).
- An instance management object type (the OraApps 11i Cloning object type), which is discussed in [Instance Management Object Type on page 276](#).

AOL and GL Object Types

To configure AOL or GL object types:

1. Enable the object type.
2. To modify the order of Object Migrator submissions to Oracle, adjust the Object Migrator or GL Migrator sequence value.

For More Information

For more information about these object types, see [AOL and GL Object Types on page 240](#).

AOL:Single Entity Object Types

AOL:Single *Entity* object types include the following:

- AOL:Single Conc Mgr Entry object type
- AOL:Single GUI Menu Entry object type
- AOL:Single Report Group Unit Rel 10 object type
- AOL:Single Report Group Unit Rel 11 object type

AOL:Single *Entity* object types allow users to do the following:

- Add a single specialization rule to an existing concurrent manager definition
- Add a single component to an existing GUI menu or report group.

You need to enable AOL:Single *Entity* object types before you can use them.

For More Information

For more information about these object types, see:

- [AOL:Single Conc Mgr Entry Object Type on page 243](#)
- [AOL:Single GUI Menu Entry Object Type on page 245](#)
- [AOL:Single Report Group Unit Rel 10 Object Type on page 247](#)
- [AOL:Single Report Group Unit Rel 11 Object Type on page 249](#)

OraApps Oracle Workflow Object Type

The OraApps Oracle Workflow object type allows users to migrate embedded Oracle workflow definitions between Oracle E-Business Suite Release 11i instances. By using Mercury products to migrate these definitions, users can increase the security, control, and visibility of these changes, and also integrate with a source code control system.

To configure the OraApps Oracle Workflow object type:

1. Prior to downloading or uploading Web setup data, set the Oracle instance context.

The object type assumes this context is automatically set at login. If not, amend the object type to set the context, for example, by calling the appropriate `APPSORA.env` file.

2. Enable the object type.

For More Information

For more information about this object type, see [OraApps Oracle Workflow Object Type on page 255](#).

OraApps AKLOAD Migrate AK Setups Object Type

The OraApps AKLOAD Migrate AK Setups object type allows users to automate the migration of custom Web page configuration data using the Oracle AKLOAD utility, saving effort, increasing auditability, and reducing data entry errors. The following types of data can be migrated:

- Attributes
- Objects
- Regions
- Flows

Be aware of the following configuration considerations:

- Only custom data should be migrated with this object type, because it does not validate or prevent migration of standard or Mercury-supplied data.
- This object type is configured by default to connect using JDBC. You can configure it to use other protocols supported by AKLOAD.
- Only data from Oracle E-Business Suite instances that are at the same point release and patch level should be migrated. This object type does not check the Oracle E-Business Suite version.
- You can put the extract file under a specific AppCode's base path instead of the server base path by specifying it in the package line definition.
- You need to apply Oracle patch #2594863 to your Oracle Applications instances, or the instance should be at Release 11.5.9 or later, for the following fields to get populated by the AKLOAD utility:
 - display_height, css_class_name, poplist_viewobject
 - poplist_display_attribute, poplist_value_attribute
 - css_label_class_name
- Any database views upon which your Web setup depends should already exist in the destination instance. The existence of these views is not validated by the object type.
- Migration of security data is not supported. Security information can be migrated using the AOL:Resp migrator.
- Migration of personalization values is not supported.

To configure the OraApps AKLOAD Migrate AK Setups object type:

1. Enable the object type.
2. The Oracle Applications instance context must be set prior to downloading or uploading Web setup data. The object type should be modified to do so if the existing logic is not adequate.
3. This object type uses a destination value of **\$NLS_LANG** during import. If you need a different value, change the object type logic.

For More Information

For more information about this object type, see [OraApps AKLOAD Migrate AK Setups Object Type on page 251](#).

Deleting Installation Files

After the Extension has been running successfully for a substantial period of time, you can optionally delete all of the installation or upgrade files. However, Mercury recommends that you retain (or copy) the log files.

Managing Integration with Oracle E-Business Suite

This section provides information about changing the integration method to local or linked, and managing Oracle E-Business Suite downtime in linked integrations. For an introduction to integration, see [Integration of the Extension with Oracle E-Business Suite](#) on page 33.



Never change the integration method while Mercury IT Governance Server is running. Doing so can lead to inconsistent and unexpected results.



All existing concurrent requests must be completed and processed within Mercury IT Governance Center before changing the integration method. Otherwise, request data may be lost.

Changing the Integration Method to Local

To change the integration method to local:

1. Be sure that Mercury IT Governance Center is sharing a database with the Oracle E-Business Suite instance you want to be the primary Object Migrator host.
2. Be sure that all package line executions related to Object Migrator and GL Migrator in Mercury IT Governance Center have completed.
3. Stop the Mercury IT Governance Server.



Never change the integration method while Mercury IT Governance Server is running, since this may lead to inconsistent and unexpected results or errors.

4. While logged into the database as the database user who owns Mercury IT Governance Center, run the `fnd_drop_dummy.sql` script.

This script drops any existing local tables and database packages that need to be refreshed.

You must run this script from `ITG_Home/deploy/600/OracleApps/phases/db/fnd/`

For example, run:

```
spool drop_dummy_log.log
@@fnd_drop_dummy.sql
```

5. Review the log for errors. Correct any problems found and run the script again.
6. From the `ITG_Home/ deploy/600/OracleApps/phases/db/fnd/` directory, and while logged into the database as the database user who owns Mercury IT Governance Center, run the `fnd_knta_synonyms.sql` script.

This script:

- Uses the `oraapps_schema` variable.
- Defines synonyms in the Mercury IT Governance Center account for certain objects in the APPS account

For example, run:

```
spool fnd.knta.synonyms.log
@@fnd_knta_synonyms.sql oraapps_schema
```

[Table 2-5](#) describes the `oraapps_schema` variable used in the script.

7. Review the log for errors. Correct any problems found and run the script again.
8. From the `ITG_Home/ deploy/600/OracleApps/phases/db/fnd/` directory, and while logged into the database as the database user who owns the Oracle E-Business Suite APPS schema (usually APPS), run the `fnd_apps_grants.sql` script.

This script:

- Uses the variables `itg_schema`, `oraapps_schema`, and `oraapps_password`.
- [Table 2-5](#) describes the variables used in the script.
- Installs the KINTANA_SUBMIT_REQUEST Package
- Grants access to certain objects to the Mercury IT Governance schema
- Should run as the APPS database user

For example, run:

```
spool fnd.apps.grants.log
@@fnd_apps_grants.sql itg_schema oraapps_schema
oraapps_password
```

9. If any values have changed, such as the user or password that is used to retrieve log files from the primary Object Migrator host, change the Mercury IT Governance Server configuration to reflect the new values.
10. Start the Mercury IT Governance Server.

Table 2-5. Variables for scripts to convert to local integration

Variable	Description
<i>itg_schema</i>	Name of the schema where Mercury IT Governance Center is installed
<i>oraapps_schema</i>	Name of the Oracle E-Business Suite APPS schema in this database
<i>oraapps_password</i>	Password for the APPS schema

Changing the Integration Method to Linked

To change the integration method to linked:

1. Be sure that all package line executions related to Object Migrator and GL Migrator in Mercury IT Governance Center have completed.
2. Stop the Mercury IT Governance Server.



Never change the integration method while Mercury IT Governance Server is running, since this may lead to inconsistent and unexpected results or errors.

3. While logged into the database as the database user who owns Mercury IT Governance Center, run the `fnd_drop_dummy.sql` script.

This script drops any existing local tables and database packages that need to be refreshed.

You must run this script from `ITG_Home/deploy/600/OracleApps/phases/db/fnd/`

For example, run:

```
spool drop_dummy_log.log
@@fnd_drop_dummy.sql
```

4. Review the log for errors. Correct any problems found and run the script again.
5. Make sure that a database link to the APPS account of the Oracle E-Business Suite instance is defined in the Mercury IT Governance schema.
6. Test the link as shown in the following example:

```
select count(*) from fnd_application@db_link_name;
```

The `db_link_name` variable represents the name (for example, `PROD_APPS_LINK`) of the database link defined in the Mercury IT Governance schema that connects to the APPS account of the primary Object Migrator host.

7. From the `ITG_Home/deploy/600/OracleApps/phases/db/fnd/` directory, and while logged into the database as the database user who owns Mercury IT Governance Center, run the `fnd_linked_synonyms.sql` script.

For example, run:

```
spool linked_synonyms.log
@@fnd_linked_synonyms.sql db_link_name
```

8. Review the log for errors. Correct any problems found and run the script again.
9. If any values have changed, such as the user or password that is used to retrieve log files from the primary Object Migrator host, change the Mercury IT Governance Server configuration to reflect the new values.
10. Start the Mercury IT Governance Server.

Managing Linked Integrations During Oracle E-Business Suite Downtime

When the primary Object Migrator host uses a linked integration with Mercury IT Governance Center, if the Oracle E-Business Suite database becomes unavailable, there may be some loss of functionality in Mercury IT Governance Center and you cannot do any of the following:

- Submit Object Migrator or GL Migrator concurrent requests to the primary Object Migrator host
- Track the status of concurrent requests submitted to the primary Object Migrator host
- Create new package lines for AOL objects (because values will be unavailable for certain fields)
- Add or modify the OM DB value defined on the Environment window, **Extension Data** tab, **Oracle Applications** subtab, **Database** section
- Validate the OM DB value using the Environment Checker

Impact on WatchDog Functionality

The Concurrent Request WatchDog is a part of the Extension that:

- Tracks the status of concurrent requests submitted to Oracle E-Business Suite
- Updates the related package lines with the outcome
- Links the log and output files of a concurrent request to the package line execution log where Object Migrator is running

These activities cannot take place when the Oracle E-Business Suite database is unavailable. If the Concurrent Request WatchDog cannot locate a submitted concurrent request after three attempts, it stops watching the request and terminates the package line execution with a command execution error.

Normally, this does not pose a problem. If linked database access goes down while some concurrent requests are still pending, however, the requests may produce a Mercury IT Governance Center error even though they will eventually run in Oracle E-Business Suite. To help manage this situation, you can add the following parameter in the Mercury IT Governance Server configuration file (`server.conf`) to specify a value for the number of times you want Mercury IT Governance Center to check for the request before forcing an error status:

```
com.kintana.core.server.ORACLE_APPS_MAX_CHECKS=value
```

For example, to set this parameter to 60 (that is, to check 60 times before forcing an error status), add:

```
com.kintana.core.server.ORACLE_APPS_MAX_CHECKS=60
```

If the Oracle E-Business Suite database is scheduled to be down for an extended period while the Mercury IT Governance Center instance remains available, you must make sure that all concurrent requests are completed before the Oracle E-Business Suite database is taken down.

Impact Depending on Oracle E-Business Suite Release

When Mercury IT Governance Server starts up, it detects the release of Oracle E-Business Suite being used in the primary Object Migrator host instance. If the database link to the Oracle E-Business Suite instance is down at the time Mercury IT Governance Server starts up (for example, if the Oracle E-Business Suite database is down for backup or patching), Mercury IT Governance Center cannot determine the Oracle E-Business Suite release.

In this case, Mercury IT Governance Center determines the release as required during processing. In high-volume situations, this can lead to performance degradation, because redundant queries are issued. If you restart the Mercury IT Governance Server, it attempts to detect the Oracle E-Business Suite release again.

If this situation is likely to occur frequently, and restarting Mercury IT Governance Server is not a viable option, you can provide default release (version) information for Mercury IT Governance Center to use during processing. Depending on the Oracle E-Business Suite release, add one of the following parameters to your server configuration file (`server.conf`).

For Release 10.x, add the following parameter:

```
com.kintana.core.server.ORACLE_APPS_VERSION=R10
```

For Release 11 or 11i, add the following parameter:

```
com.kintana.core.server.ORACLE_APPS_VERSION=R11
```

If you change the primary Object Migrator host, and if it was manually defined, you need to update this parameter.

Upgrading the Extension

In This Chapter:

- *Overview of Upgrade*
- *Product Features Introduced in Release 5.5.0*
 - *Product Renaming*
 - *Support for Oracle E-Business Suite Release 11.5.9*
 - *Support for Oracle Patch Unified Drivers*
 - *Improvements to Reports*
 - *Additions to AOL Object Type Parameters*
 - *Miscellaneous Changes*
- *Product Features Introduced in Release 6.0.0*
- *General Upgrade Impacts and Guidelines*
- *Impacts of Upgrading from Release 5.0.0*
 - *Support for Oracle Patch Unified Drivers*
 - *Improvements to Reports*
 - *Additions to AOL Object Type Parameters*
- *Impacts of Upgrading from Release 5.5.0*
 - *Object Type Compatibility with Oracle ADADMIN Changes*
 - *Dashboard Menus and Tab Templates (Pre-configured Dashboard Pages)*
 - *Improved Control of Oracle ADPATCH Utility*
 - *Improved Patch Capture*
 - *Enhanced OraApps Release 11i Upgrade Project Template*
 - *Improved OraApps Critical Requests Summary Report*
 - *Other New Functionality*
- *Upgrade Process*
- *Deleting Upgrade Files*

Overview of Upgrade

You can upgrade to Mercury Change Management Extension for Oracle E-Business Suite release 6.0.0 from either of the following releases:

- Mercury Change Management Extension for Oracle E-Business Suite release 5.5.0
- Kintana™ Accelerator for Oracle E-Business Suite release 5.0.0

This chapter provides information about performing either upgrade.

Before you can upgrade the Extension to release 6.0.0, you must upgrade Mercury IT Governance Center to release 6.0.

If you are upgrading the Extension from any release earlier than Kintana Accelerator for Oracle E-Business Suite release 5.0.0, you must first upgrade it to release 5.0.0, then upgrade it to release 6.0.0. The Mercury IT Governance Center upgrade process enforces this rule. For information about upgrades to (and features of) releases earlier than release 5.0.0, see the documentation for all of the releases from your current product release through release 5.0.0 as necessary.

Release 6.0.0 of the Extension can integrate with earlier releases of Object Migrator and GL Migrator (formerly called Kintana Object*Migrator and Kintana GL*Migrator). However, only parameters supported in the earlier release of Object Migrator and GL Migrator will be valid. For full integration with the latest Oracle E-Business Suite releases and minipacks, install and use the most recent releases and patch levels of Object Migrator and GL Migrator.

Product Features Introduced in Release 5.5.0

Whether you are upgrading to release 6.0.0 from release 5.0.0 or from release 5.5.0, for information about Extension features that are introduced in release 6.0.0, see [What's New in Release 6.0.0 on page 29](#).

As described in the following sections, the key features introduced in release 5.5.0 of the Extension were:

- Product renaming
- Support for Oracle E-Business Suite Release 11.5.9
- Support for Oracle Patch unified drivers
- Improvements to reports
- Additions to AOL object type parameters
- Miscellaneous changes

Product Renaming

In release 5.5.0, some product names changed as shown in [Table 3-1](#).

Table 3-1. Product name changes in release 5.5.0

Product Name in Release 5.0.0	Product Name in Release 5.5.0
Kintana Accelerator	Mercury Change Management Extension
Kintana Deliver	Mercury Change Management
Kintana Drive	Mercury Project Management
Kintana Create	Mercury Demand Management™

Support for Oracle E-Business Suite Release 11.5.9

Release 5.5.0 of the Extension introduced support for the 11.5.9 point release of Oracle E-Business Suite and the AD.H minipack.

Support for Oracle Patch Unified Drivers

In Oracle E-Business Suite Release 11.5.9 and later, some Oracle E-Business Suite patches contain a single driver file called a unified driver, rather than separate copy, database, and generate drivers. Oracle's patch utility (ADPATCH) and patch merge utility (ADMRGPCH) support both patches with split drivers and patches with unified drivers.

Release 5.5.0 of the Extension introduced support for unified drivers. Patches using unified drivers can be merged, applied, captured, and analyzed using the Extension. When patching, users can control which portions of the unified driver are executed using a parameter (new for release 5.5.0) called Driver Execution Options.

Oracle continues to deliver both unified and split driver patches, and the Extension supports both kinds of drivers.

For information about the impact of this feature, see [Support for Oracle Patch Unified Drivers on page 95](#).

Improvements to Reports

Release 5.5.0 added functionality to improve handling and performance of the following four reports used in the Extension:

- OraApps Apps Issues Detail Report
- OraApps Apps Issues Summary Report
- OraApps IT Demand Summary Report

The enhancements accomplished the following:

- Making use of Java Server Pages (JSPs)
- Allowing cancellation of running reports
- Allowing reports to be printed, saved, and sent by email

For information about the impact of this feature, see [Improvements to Reports on page 96](#).

Additions to AOL Object Type Parameters

New parameters were added to the AOL:Resp and AOL>User object types to govern the synchronization of workflow tables in Oracle E-Business Suite Release 11.5.9 instances when migrating data to those instances. The new parameters have an effect only if the Object Migrator in use is at release level 5.1 or later.

For information about the purpose and use of this parameter, see the *Mercury Object Migrator Guide*.

For information about the impact of this feature, see [Additions to AOL Object Type Parameters on page 97](#).

Miscellaneous Changes

Several object types have improved logic for detecting Windows-based platforms during processing.

Product Features Introduced in Release 6.0.0

For information about the key features introduced in Extension release 6.0.0, see [What's New in Release 6.0.0 on page 29](#):

General Upgrade Impacts and Guidelines

The Extension upgrade process replaces code entities installed in earlier releases, including all reference (REFERENCE) versions of the Extension entities, such as object types and workflows.

During the upgrade process, existing non-reference entities, including those that you have copied, renamed, or customized for your specific environment, are not modified in any way. In other words, these non-reference entities retain all your customizations and they do not reflect any changes made in release 6.0.0, so they might not be compatible with Mercury IT Governance Center release 6.0.

Therefore, after upgrading to release 6.0.0, you should review the new reference entities. If necessary, copy and rename them, then apply the same customizations that were previously made to the corresponding non-reference entities for the earlier release.

Menu XML files are backed up during the upgrade.

Impacts of Upgrading from Release 5.0.0



Note

If you are upgrading to release 6.0.0 from release 5.5.0, go to the section [Impacts of Upgrading from Release 5.5.0 on page 98](#).

If you are upgrading from release 5.0.0 directly to release 6.0.0, without ever installing release 5.5.0, you need to review both of the following sections:

- This section, which discusses the impacts of an upgrade from release 5.0.0 to release 5.5.0.
- The section [Impacts of Upgrading from Release 5.5.0 on page 98](#), which discusses the impacts of an upgrade from release 5.5.0 to release 6.0.0.

The following sections discuss the following impacts of the upgrade from release 5.0.0:

- Support for Oracle Patch Unified Drivers
- Improvements to reports
- Additions to AOL object-type parameters

Support for Oracle Patch Unified Drivers

The changes made in release 5.5.0 to support unified driver files affect the following entities:

- `ksc_oa_capture_patch` special command.
- **OraApps Patch Detail Report.** The OraApps Patch Detail Report was modified to change the prompt for the patch driver field. Use the reference copy of the report as a guide, and make the changes to your existing version. For more information about this report, see [OraApps Patch Detail Report on page 403](#).
- **OA - Capture Patch workflow step source.** If you are using the standard OA - Capture Patch workflow step source to capture patch data and you have not modified it, it will be automatically updated to support unified drivers. If you are using a different step source (for example, a copy), or you have modified the step source, you must manually apply the changes using the reference copy as a guide. Changes were generally in command conditions, messages, and command names. If you do not make these updates, details of patches using unified drivers will not be captured.
- **R11 Oracle Patch object type.** Changes were made to this object type. If you are already using a version of this object type, you need to copy the new parameter Driver Execution Options (token `P_DRIVER_OPTIONS`) and the new command Execute Patch with Execution Options from the reference copy of the R11 Oracle Patch object type into your existing version. You should also modify the Execute Patch command condition using the reference version as a model. If you do not make these changes, you will not be able to control which portions of a unified driver are executed.

Oracle's product documentation includes specific instructions for applying patches using unified drivers to multi-tier Oracle E-Business Suite instances. If the workflows you are using to apply patches to your Oracle E-Business Suite instances conflict with Oracle's guidelines, you should modify the workflows prior to applying patches with unified drivers.

- **OraApps R11i Merge Patch object type.** This object type was modified to check for cases where patches with unified drivers were merged by a version of `ADMRGPCH` that does not support unified drivers. You should review the "Run the `ADMRGPCH` Command" command in the reference copy of the object type and add the same check to the version of the object type in use. This check is important, since patches can become corrupted if patches with unified drivers are merged by a version of `ADMRGPCH` that does not support unified drivers.

Improvements to Reports

Improvements to the following reports in release 5.5.0 made previous versions of the reports obsolete and unusable:

- OraApps Apps Issues Detail Report
- OraApps Apps Issues Summary Report
- OraApps IT Demand Summary Report

The following upgrade scenarios rename these reports by prepending “(***Deprecated 5.5***)” and then they disable the reports:

- Mercury IT Governance Center was previously upgraded to release 5.5 and is now being upgraded to release 6.0
- Mercury IT Governance Center is being upgraded from release 5.0 directly to release 6.0

To use these reports in these cases, you must copy their new reference versions.

If Mercury IT Governance Center was first installed at release 5.5, these reports are not renamed or disabled.

For More Information

For more information about these reports, see:

- [OraApps Apps Issues Detail Report on page 383](#)
- [OraApps Apps Issues Summary Report on page 387](#)
- [OraApps IT Demand Summary Report on page 393](#)

Additions to AOL Object Type Parameters

In release 5.5.0, new parameters were added to the AOL:Resp and AOL>User object types to govern whether synchronization of workflow tables takes place when migrating data to Oracle E-Business Suite Release 11.5.9 or later. The new parameters take effect only when using Object Migrator release 5.1 or later. By default, workflow tables become synchronized.

To use the new functionality:

1. Make backup copies of the object types you use to migrate responsibilities and users (by default, AOL:Resp and AOL>User).
2. To the AOL:Resp responsibility, add the Synchronize Workflow Tables parameter (token P_SYNCH) by copying the parameter from the reference copy of the AOL:Resp object type.
3. To the AOL>User responsibility, add the Synchronize Workflow Tables parameter (token P_SYNCH) by copying the parameter from the reference copy of the AOL>User object type.
4. If you want, adjust the parameter display for each object type.

If you do not add the parameter and you are using Object Migrator 5.1 or later, then workflow tables will always be synchronized.

Impacts of Upgrading from Release 5.5.0

The following sections discuss the following impacts of the upgrade from release 5.5.0 to release 6.0.0:

- Object type compatibility with Oracle ADADMIN changes
- Dashboard menus and tab templates (pre-configured Dashboard pages)
- Improved control of Oracle ADPATCH utility
- Improved patch capture
- Enhanced OraApps Release 11i Upgrade project template
- Improved OraApps Critical Requests Summary Report
- Other new functionality

Object Type Compatibility with Oracle ADADMIN Changes

To support the menu structure changes that Oracle made to the ADADMIN utility after Release 11i minipack AD.H (specifically, in the AD Post-H Consolidated 12/01 patch), the following object types have been revised in release 6.0.0:

- OraApps ADADMIN Compile APPS Schema object type
- OraApps ADADMIN Compile Flexfields object type

Old menu structures in the ADADMIN utility are still supported. The revised object types determine which menu structure is in use and pass an indicator to the `ksc_respond` command, which has also been updated for release 6.0.0.

In the ReadMe file for IT Governance Center release 5.5 PL2, Mercury provided details of how to revise the command logic of these object types so that they operate in the same way as the new release 6.0.0 reference versions in the Extension.

Because that command logic changed significantly, Mercury recommends that you copy the new reference versions of the object types and modify the copies with any previous customizations you made, other than those described in PL2. If you use existing OraApps ADADMIN Compile APPS Schema and OraApps ADADMIN Compile Flexfields object types to which the changes described in PL2 have not been applied, they will not work with the ADADMIN utility in Oracle environments at the AD Post-H Consolidated 12/01 patch level or later.

Dashboard Menus and Tab Templates (Pre-configured Dashboard Pages)



Note

Tab templates, as they were called prior to release 6.0.0, are called pre-configured Dashboard pages or modules in release 6.0.0. See the *Configuring the Standard Interface* document for information about how to configure them.

The Extension installation procedure makes backup copies of your existing menu and tab XML files in the format `file_name.OracleApps`, for example `OracleApps_menu.xml.OracleApps`.

The provided files are:

- `ITG_Home/conf/menus/oraapps_menu.xml`
- `ITG_Home/conf/menus/oraapps_menu_roles.xml`

If you do not currently use the Extension menus and tab templates in a release earlier than release 6.0.0 and you do not plan to use them immediately, move the files to `ITG_Home/deploy/600/OracleApps` for storage until you need them.

If you have been using the Extension menus in release 5.0.0 or release 5.5.0, you need to make some changes for your existing menus to function correctly with release 6.0.0. The Access Type and License Type security entries in the menus should be modified to use the release 6.0.0 values. The menus delivered with the 6.0.0 Extension can be used as a model. For more information, see the *Security Model Guide and Reference*.

Installing the release 6.0.0 pre-configured Dashboard pages requires running a script after upgrading the Extension. See [Installing the Extension's Pre-configured Dashboard Pages on page 53](#). When you perform this procedure, the following new pages are added with the suffix "1":

- Oracle Apps Project Manager1
- Oracle Apps Project Sponsor1
- Oracle Apps Reports Track Owner1
- Oracle Apps DBA1

These pre-configured Dashboard pages do not have new functionality as compared to the tab templates of earlier releases.

Whether or not you install these release 6.0.0 pre-configured Dashboard pages, the tab templates from earlier releases are converted to equivalent pre-configured Dashboard pages, while preserving any customizations you may have made.

For More Information

For more information about customizing menus and pre-configured Dashboard pages, see the *Customizing the Standard Interface* document.

Improved Control of Oracle ADPATCH Utility

As it runs, the Oracle ADPATCH utility compares the Applications System Name stored in the database with the name in the APPL_TOP file system. Normally, these names should be the same, and a conflict (difference) is reported as an error, reflecting a misconfiguration that should be corrected before running ADPATCH.

In release 6.0.0, the Tolerate Applications System Name conflict parameter has been added to the R11 Oracle Patch object type. When the parameter is set to the default value of **No**, the ADPATCH utility terminates with an error, as it has in the past, if the Applications System Name stored in the database and the name in the APPL_TOP file system are different. When the parameter is set to **Yes**, the utility proceeds despite the difference in names. Do not change the default setting of **No** unless there is a specific need to allow different names.

Mercury recommends that you copy this parameter from the new reference R11 Oracle Patch object type to all object types that are used to apply Release 11i patches (because doing so is probably easier than reapplying your previous customizations to each object type). You can search for the parameter token P_IGNORE_APPL_SYS_NAME.

Improved Patch Capture

Oracle patch data captured between the installation of Extension release 5.5.0 and any subsequent patch level or upgrade of the Extension could be incomplete, affecting reporting and analysis.

To address this, perform the following procedure once:

1. From the Oracle schema that owns IT Governance Center, run the script

```
kacc_oa_check_patch_capture_550_PL1_1.sql
```

located at *ITG_Home/scripts*.

This script lists all patches that may have incomplete details, and it separates non-applied patches from applied patches.

2. For each non-applied patch:
 - a. Run the OraApps Patch Detail Report, reloading the data and setting the Remove Data after Run field to **Yes**. For efficiency, turn off display parameters (bugs, files, and environments) by leaving the Includes Bug and Includes File fields blank and setting the Show SubPatch Info and Show Env Display Info fields to **No**.

The patch archive file must be available in the patch repository (by default, Patch Stage).
 - b. Run the OraApps Patch Detail Report again to reload the patch. This load will include complete data. Display parameters (bugs, files, environments) can remain off if you do not need to see the details of the patch.
3. For each applied patch that was found:
 - a. Copy the OraApps Patch Detail Report as “OraApps 6.0 Patch ReCapture” and set the Append Data field to be displayed.
 - b. Run the report for each patch that needs to be recaptured, with the Load Data and Append Data fields set to **Yes**. Display parameters (bugs, files, environments) can be turned off if you do not need to see the details of the patch. The patch archive file must be available in the patch repository (by default, Patch Stage).
 - c. Disable the report type. It will not be needed again.

Enhanced OraApps Release 11i Upgrade Project Template

The OraApps Release 11i Upgrade project template supports multiple currencies and multiple calendars in release 6.0.0. The upgrade process installs a new reference (REFERENCE) version of this project template.

To use the updated project template, copy the new reference version.

Improved OraApps Critical Requests Summary Report

The OraApps Critical Requests Summary Report has minor improvements in release 6.0.0. The upgrade process installs a new reference (REFERENCE) version of this report.

To use the updated report, copy the new reference version.

Other New Functionality

See the following sections for information about other new Extension functionality for release 6.0.0:

- [Reports Accessible Only from Standard Interface on page 30](#)
- [Licensing on page 30](#)

Upgrade Process

To upgrade from release 5.0.0 to 5.5.0 of the Extension, follow the instructions in the section [Downloading and Installing the Extension on page 50](#) (in [Chapter 2, Installing and Configuring the Extension, on page 41](#)). These instructions are equally relevant for first-time installations or upgrades.

In addition, if you need to install the pre-configured Dashboard pages (modules) provided for the Extension, see [Installing the Extension's Pre-configured Dashboard Pages on page 53](#).

Deleting Upgrade Files

After the Extension has been running successfully for a substantial period of time, you can optionally delete all of the installation or upgrade files. However, Mercury recommends that you retain (or copy) the log files.

Chapter

4

Extension Interface

In This Chapter:

- *Menus*
 - *Oracle Applications Menu*
 - *Oracle Applications Roles Menu*
 - *Pre-configured Dashboard Pages*
 - *Overview of Pages and Portlets*
 - *Adding Pre-configured Dashboard Pages*
 - *Oracle Apps Project Manager Page and Its Portlets*
 - *Oracle Apps Project Sponsor Page and Its Portlets*
 - *Oracle Apps Reports Track Owner Page and Its Portlets*
 - *Oracle Apps DBA Page and Its Portlets*
-

Menus

The menu bar in the standard interface has two menus that are specifically associated with the Extension:

- **Oracle Applications**, which is organized by Extension functionality, allowing quick access to tasks in functional areas such as change management.
- **Oracle Applications Roles**, which is organized by role, providing direct access to tasks associated with specific roles.

Both menus provide access to key Extension functionality. Each is structured to support the following predefined business roles:

- Project manager
- Project sponsor
- Track owner
- DBA

Items relating to a given business role are defined to be accessed by that role. For more information about these business roles, see [Business Concepts Used in the Oracle Environment on page 36](#).

The Extension menus are seamlessly integrated with other menus in the menu bar. The Oracle-related menus include security definitions that restrict display of menu items to only those users authorized to see them.



Note

The menus described here represent the default configuration installed with the product. Your customized configuration may differ.

Oracle Applications Menu

The **Oracle Applications** menu has a submenu for each major area of Extension functionality. Users see only those menu items for which they have licenses and permissions. The submenus are:

- Issue Management
- Project Management
- Change Management
- Instance Management
- Patch Management

Depending on the submenu, users can:

- Create requests
- Run reports
- View or maintain key business processes
- Access the Mercury IT Governance Workbench for additional tasks

Issue Management Submenu

Items in the Issue Management submenu are described in [Table 4-1](#).

Table 4-1. Issue Management items in Oracle Applications menu

Menu Item or Submenu	Description
New Apps Issue	Creates a new OraApps Application Issue request.
Reports	
OraApps Issues Detail	Runs the OraApps Apps Issues Detail Report.
OraApps Issues Summary	Runs the OraApps Apps Issues Summary Report.
Administration	
Apps Issue Process	Opens the OraApps Apps Issue Process in the Workbench.

Project Management Submenu

Items in the Project Management submenu are described in [Table 4-2](#).

Table 4-2. Project Management items in Oracle Applications menu

Menu Item or Submenu	Description
New GAP Analysis request	Creates a new OraApps GAP Analysis request.
New Status Update request	Creates a new OraApps Status Update request.
Open Project Workbench	Opens the Workbench to the Projects screen.
Reports	
Oracle Apps IT Demand Summary	Runs the OraApps IT Demand Summary Report.
Oracle Apps Critical Request Summary	Runs the OraApps Critical Open Request Summary Report.
Ready Oracle Apps Tasks List	Runs the OraApps Ready Tasks List Report.
Unassigned/Ready Oracle Apps Tasks List	Runs the OraApps Unassigned/Ready Tasks List.
Administration	
GAP Analysis Process	Opens the OraApps GAP Analysis Process in the Workbench.
Status Update Process	Opens the OraApps Status Update Process in the Workbench.

Change Management Submenu

Items in the Change Management submenu are described in [Table 4-3](#).

Table 4-3. Change Management items in Oracle Applications menu

Menu Item or Submenu	Description
New Interface request	Creates a new OraApps Interface request.
New Conversion request	Creates a new OraApps Conversion request.
New Setup Change request	Creates a new OraApps Setup Change request.
New Report request	Creates a new OraApps Report request.
New Enhancement request	Creates a new OraApps Enhancement request.
Open Package Workbench	Opens the Workbench to the Packages screen.

Table 4-3. Change Management items in Oracle Applications menu [continued]

Menu Item or Submenu	Description
Administration	
Interfaces Process	Opens the OraApps Interfaces Process in the Workbench.
Conversion Process	Opens the OraApps Conversion Process in the Workbench.
Setup Change Process	Opens the OraApps Setup Change Process in the Workbench.
Reports Process	Opens the OraApps Reports Process in the Workbench.
Enhancements Process	Opens the OraApps Enhancement Process in the Workbench.
Deployment Process	Opens the OraApps Deployment Process in the Workbench.

Instance Management Submenu

Items in the Instance Management submenu are described in [Table 4-4](#).

Table 4-4. Instance Management items in Oracle Applications menu

Menu Item or Submenu	Description
New Cloning request	Creates a new OraApps Cloning request.
Open Package Workbench	Opens the Workbench to the Packages screen.
Reports	
Compare Custom Database Setup	Runs the Compare Custom Database Setup Report.
Compare Oracle Environments	Runs the Compare Oracle Environments Report.
Administration	
Env. Cloning Request Process	Opens the OraApps 11i Cloning Request process in the Workbench.
Env. Cloning Process	Opens the OraApps 11i Cloning process in the Workbench.

Patch Management Submenu

Items in the Patch Management submenu are described in [Table 4-5](#).

Table 4-5. Patch Management items in Oracle Applications menu

Menu Item or Submenu	Description
Open Package Workbench	Opens the Workbench to the Packages screen.
Reports	
Compare Custom Database Setup	Runs the Compare Custom Database Setup Report.
Compare Oracle Environments	Runs the Compare Oracle Environments Report.
Patch Application Comparison	Runs the Patch Application Comparison Report.
Patches Applied to an Env	Runs the Patches Applied to an Env Report.
Pending Patches	Runs the Pending Patches Report.
OraApps Patch Detail	Runs the OraApps Patch Detail Report.
OraApps Patch Analysis	Runs the OraApps Patch Analysis Report.
Administration	
Patch Deployment Process	Opens the OraApps Patch Deployment process in the Workbench.

Oracle Applications Roles Menu

The **Oracle Applications Roles** menu has a submenu for each of the following business roles:

- Project sponsor
- Project manager
- Track owner
- Apps DBA

The menu items under the submenus include common tasks for each role, for example, creating requests or running reports. Users see only those menu items for which they have licenses and permissions.

Project Sponsor Submenu

Items in the Project Sponsor submenu are described in [Table 4-6](#).

Table 4-6. Project Sponsor items in Oracle Applications Roles menu

Menu Item or Submenu	Description
New GAP Analysis request	Creates a new OraApps GAP Analysis request.
New Interface request	Creates a new OraApps Interface request.
New Conversion request	Creates a new OraApps Conversion request.
New Setup Change request	Creates a new OraApps Setup Change request.
New Report request	Creates a new OraApps Report request.
New Enhancement request	Creates a new OraApps Enhancement request.
New Status Update request	Creates a new OraApps Status Update request.
New Apps Issue	Creates a new OraApps Application Issue request.
Reports	
Oracle Apps IT Demand Summary	Runs the OraApps IT Demand Summary Report.
Oracle Apps Critical Request Summary	Runs the OraApps Critical Open Request Summary Report.
OraApps Issues Detail	Runs the OraApps Apps Issues Detail Report.
OraApps Issues Summary	Runs the OraApps Apps Issues Summary Report.
Ready Oracle Apps Tasks List	Runs the OraApps Ready Tasks List Report.
Unassigned/Ready Oracle Apps Tasks List	Runs the OraApps Unassigned/Ready Tasks List.

Project Manager Submenu

Items in the Project Manager submenu are described in [Table 4-7](#).

Table 4-7. Project Manager items in Oracle Applications Roles menu

Menu Item or Submenu	Description
New GAP Analysis request	Creates a new OraApps GAP Analysis request.
New Interface request	Creates a new OraApps Interface request.
New Conversion request	Creates a new OraApps Conversion request.
New Setup Change request	Creates a new OraApps Setup Change request.
New Report request	Creates a new OraApps Report request.
New Enhancement request	Creates a new OraApps Enhancement request.
New Status Update request	Creates a new OraApps Status Update request.
New Apps Issue	Creates a new OraApps Application Issue request.
New Cloning request	Creates a new OraApps Cloning request.
Open Project Workbench	Opens the Workbench to the Projects screen.
Reports	
Oracle Apps IT Demand Summary	Runs the OraApps IT Demand Summary Report.
Oracle Apps Critical Request Summary	Runs the OraApps Critical Open Request Summary Report.
OraApps Issues Detail	Runs the OraApps Apps Issues Detail Report.
OraApps Issues Summary	Runs the OraApps Apps Issues Summary Report.
Ready Oracle Apps Tasks List	Runs the OraApps Ready Tasks List Report.
Unassigned/Ready Oracle Apps Tasks List	Runs the OraApps Unassigned/Ready Tasks List.
Administration	
GAP Analysis Process	Opens the OraApps GAP Analysis Process in the Workbench.
Interfaces Process	Opens the OraApps Interfaces Process in the Workbench.
Conversion Process	Opens the OraApps Conversion Process in the Workbench.
Setup Change Process	Opens the OraApps Setup Change Process in the Workbench.
Reports Process	Opens the OraApps Reports Process in the Workbench.
Enhancements Process	Opens the OraApps Enhancement Process in the Workbench.

Table 4-7. Project Manager items in Oracle Applications Roles menu [continued]

Menu Item or Submenu	Description
Status Update Process	Opens the OraApps Status Update Process in the Workbench.
Apps Issue Process	Opens the OraApps Apps Issue Process in the Workbench.
Env. Cloning Request Process	Opens the OraApps 11i Cloning Request process in the Workbench.
Deployment Process	Opens the OraApps Deployment Process in the Workbench.

Track Owner Submenu

Items in the Track Owner submenu are described in [Table 4-8](#).

Table 4-8. Track Owner items in Oracle Applications Roles menu

Menu Item or Submenu	Description
New GAP Analysis request	Creates a new OraApps GAP Analysis request.
New Interface request	Creates a new OraApps Interface request.
New Conversion request	Creates a new OraApps Conversion request.
New Setup Change request	Creates a new OraApps Setup Change request.
New Report request	Creates a new OraApps Report request.
New Enhancement request	Creates a new OraApps Enhancement request.
New Status Update request	Creates a new OraApps Status Update request.
New Apps Issue	Creates a new OraApps Application Issue request.
New Cloning request	Creates a new OraApps Cloning request.
Open Project Workbench	Opens the Workbench to the Projects screen.
Open Package Workbench	Opens the Workbench to the Packages screen.
Reports	
Oracle Apps IT Demand Summary	Runs the OraApps IT Demand Summary Report.
Oracle Apps Critical Request Summary	Runs the OraApps Critical Open Request Summary Report.
OraApps Issues Detail	Runs the OraApps Apps Issues Detail Report.
OraApps Issues Summary	Runs the OraApps Apps Issues Summary Report.
Ready Oracle Apps Tasks List	Runs the OraApps Ready Tasks List Report.
Unassigned/Ready Oracle Apps Tasks List	Runs the OraApps Unassigned/Ready Tasks List.
Administration	
GAP Analysis Process	Opens the OraApps GAP Analysis Process in the Workbench.
Interfaces Process	Opens the OraApps Interfaces Process in the Workbench.
Conversion Process	Opens the OraApps Conversion Process in the Workbench.
Setup Change Process	Opens the OraApps Setup Change Process in the Workbench.
Reports Process	Opens the OraApps Reports Process in the Workbench.

Table 4-8. Track Owner items in Oracle Applications Roles menu [continued]

Menu Item or Submenu	Description
Enhancements Process	Opens the OraApps Enhancement Process in the Workbench.
Status Update Process	Opens the OraApps Status Update Process in the Workbench.
Deployment Process	Opens the OraApps Deployment Process in the Workbench.

Apps DBA Submenu

Items in the Apps DBA submenu are described in [Table 4-9](#).

Table 4-9. Apps DBA items in Oracle Applications Roles menu

Menu Item or Submenu	Description
New Apps Issue	Creates a new OraApps Application Issue request.
New Cloning request	Creates a new OraApps Cloning request.
Open Package Workbench	Opens the Workbench to the Package screen.
Reports	
Compare Custom Database Setup	Runs the Compare Custom Database Setup Report.
Compare Oracle Environments	Runs the Compare Oracle Environments Report.
Patch Application Comparison	Runs the Patch Application Comparison Report.
Patches Applied to an Env	Runs the Patches Applied to an Env Report.
Pending Patches	Runs the Pending Patches Report.
OraApps Patch Detail	Runs the OraApps Patch Detail Report.
OraApps Patch Analysis	Runs the OraApps Patch Analysis Report.
Administration	
Env. Cloning Process	Opens the OraApps 11i Cloning process in the Workbench.
Patch Deployment Process	Opens the OraApps Patch Deployment process in the Workbench.

Pre-configured Dashboard Pages

This section discusses pre-configured Mercury IT Governance Dashboard pages (referred to hereafter as pages, but also called modules and previously known as tab templates) and their portlets, in general and as they relate to particular Oracle applications roles.

Overview of Pages and Portlets

Users can add pre-configured pages to their Dashboards. Then, each page is accessed by clicking the tab for that page. The added pages consist of a set of portlets related to a specific business role or function, and are generally pre-configured for common usage. Once added, users can personalize the pages to suit their specific business needs.

The pre-configured pages for the Extension are available from the **Dashboard > Personalize Dashboard** menu item in the Mercury IT Governance standard interface. Users can access only those pages for which they have been granted access by their Mercury IT Governance Center administrator. Adding a page also adds the portlets configured for that page.

The pages shown in this document are samples. The content and configuration of pages may vary depending on the Mercury products you have purchased or how the Extension is used by your business.

Many of the portlets have drill-down capability. For example, on the Oracle Apps Project Manager page, in the OraApps: Critical Activities portlet shown in *Figure 4-1*, users can drill down in the # of Reqs column to see detailed information about the three OraApps Applications Issue requests.



Request Type	Application	# of Reqs
OraApps Application Issue	<Blank>	4
OraApps Cloning Request	CSM App	1
OraApps Report Request	ERP Application	1
OraApps Report Request	HR Application	1

Showing 1 to 4 of 4 [Prev](#) [Next](#) [Maximize](#)

Figure 4-1. OraApps: Critical Activities portlet

Adding Pre-configured Dashboard Pages

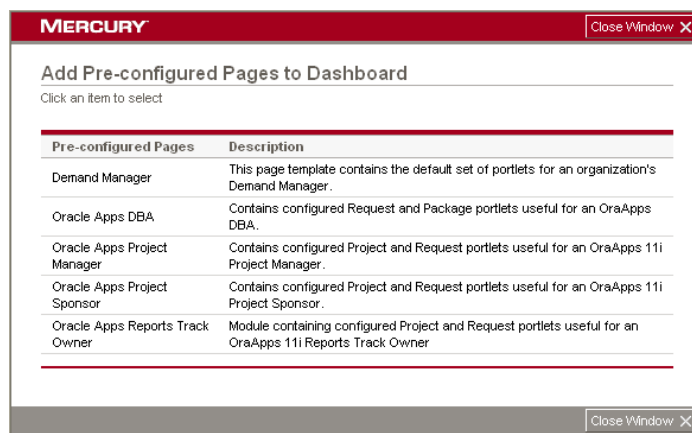
To add pre-configured Dashboard pages:

1. Log on to the Mercury IT Governance Center.
2. In the menu bar on the left, click **Dashboard > Personalize Dashboard**.

The Personalize Dashboard page appears.

3. Click **Add Pre-configured Pages**.

The Add Pre-configured Pages to Dashboard window opens.



Note

Depending on your security configuration, you might see different page names than the ones in the example.

4. From the list of page names, select the appropriate pre-configured page for your business role.

The selected page is added to your Dashboard and displayed on a new tab.

- You can now personalize the portlets as you would any portlets in the Dashboard.

To save the added page to your Dashboard, click **Save**.

To save the added page to your Dashboard and populate the portlets with data, click **Done**. Each of the following sections describes one of these pages and its portlets, with figures showing portlets with data.

Oracle Apps Project Manager Page and Its Portlets

The Oracle Apps Project Manager page (shown in [Figure 4-2](#)) allows project managers to view and access information needed to fulfill their business role. (For information about the project manager business role, see [Project Manager on page 36](#).)

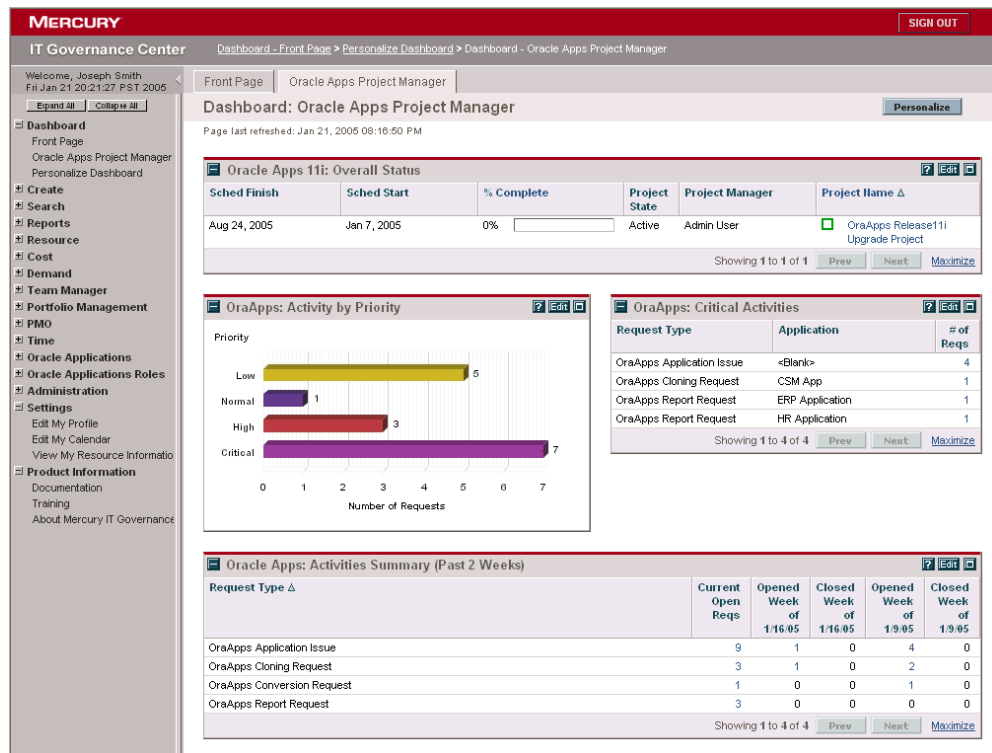


Figure 4-2. Oracle Apps Project Manager page

As described in the following sections, the Oracle Apps Project Manager page includes the following portlets by default:

- Oracle Apps 11i: Overall Status portlet
- OraApps: Critical Activities portlet
- OraApps: Activity by Priority portlet
- Apps: Activities Summary (Past 2 Weeks) portlet

Oracle Apps 11i: Overall Status Portlet

The Oracle Apps 11i: Overall Status portlet includes information about the overall status of the Oracle 11i project, such as how close it is to completion (by percentage), project state, scheduled start date, and scheduled finish date.

When you click **Edit** for this portlet, a page appears with, among other portlet filter fields, the default fields having the default values shown in [Table 4-10](#).

Table 4-10. Oracle Apps 11i: Overall Status portlet filter fields

Edit Field Name	Value
Project Name	Selected
Master Project and Subproject	OraApps Release 11i Upgrade Project
Summary Condition (green, yellow, red, and No Summary Condition checkboxes)	Green, yellow, and red selected; No Summary Condition deselected

OraApps: Critical Activities Portlet

The OraApps: Critical Activities portlet summarizes open critical Oracle-related requests, displaying information about the type and the total number of requests for each category. This information can help identify areas of the project requiring additional resources or attention. Users can drill down on each entry to view specific requests.

When you click **Edit** for this portlet, a page appears with, among other portlet filter fields, the default fields having the default values shown in [Table 4-11](#).

Table 4-11. OraApps: Critical Activities portlet filter fields

Edit Field Name	Value
Request Type	All OraApps request types
Priority	Critical
Include Closed	No

OraApps: Activity by Priority Portlet

The OraApps: Activity by Priority portlet uses a bar chart to summarize all open Oracle-related requests. The requests are grouped by priority, providing real-time data to help allocate resources to the most important tasks. Users can drill down on each entry to view specific requests.

When you click **Edit** for this portlet, a page appears with, among other portlet filter fields, the default fields having the default values shown in [Table 4-12](#).

Table 4-12. OraApps: Activity by Priority portlet filter field

Edit Field Name	Value
Request Type	All OraApps request types.

Apps: Activities Summary (Past 2 Weeks) Portlet

The Apps: Activities Summary (Past 2 Weeks) portlet displays Oracle-related request activity for the past two weeks. This includes information about the number of requests opened and closed during the last two weeks, as well as the number of open requests. Users can drill down on each total to view specific requests.

When you click **Edit** for this portlet, a page appears with, among other portlet filter fields, the default fields having the default values shown in [Table 4-13](#).

Table 4-13. Apps: Activities Summary (Past 2 Weeks) portlet filter fields

Edit Field Name	Value
Request Type	All OraApps request types.
Group By	Request Type.

Oracle Apps Project Sponsor Page and Its Portlets

The Oracle Apps Project Sponsor page (shown in *Figure 4-3*) allows project sponsors to view and access information needed to fulfill their business role. (For information about the project sponsor business role, see *Project Sponsor* on page 37.)

The screenshot displays the Oracle Apps Project Sponsor page with the following data:

Oracle Apps 11i: Overall Status

Sched Finish	Sched Start	% Complete	Project State	Project Manager	Project Name Δ
Sep 9, 2005	Jan 25, 2005	0%	Active	S K	OraApps Release11i Upgrade Project

Showing 1 to 1 of 1 | [Prev](#) | [Next](#) | [Maximize](#)

Oracle Apps 11i: All Phases

Sched Finish	% Complete	Project State	Project Name Δ
Feb 23, 2005	0%	Active	Phase I: Project Startup
Apr 13, 2005	0%	Active	Phase II: Business Operations Analysis
Apr 22, 2005	0%	Active	Phase III: Prototype, GAP Analysis, High Level Design
May 20, 2005	0%	Active	Phase IV: Development and System Test
Jul 4, 2005	0%	Active	Phase V: Integration and Performance Testing

Showing 1 to 5 of 5 | [Prev](#) | [Next](#) | [Maximize](#)

OraApps: Critical Activities

Request Type	Application	# of Reqs
OraApps Application Issue	<Blank>	5
OraApps Cloning Request	CSM App	1
OraApps Report Request	ERP Application	1
OraApps Report Request	HR Application	1

Showing 1 to 4 of 4 | [Prev](#) | [Next](#) | [Maximize](#)

Figure 4-3. Oracle Apps Project Sponsor page

As described in the following sections, the Oracle Apps Project Sponsor page includes the following portlets by default:

- Oracle Apps 11i: Overall Status portlet
- OraApps: Critical Activities portlet
- Oracle Apps 11i: All Phases portlet

Oracle Apps 11i: Overall Status Portlet

The Oracle Apps 11i: Overall Status portlet displays information about the overall status of the Oracle 11i project, such as how close it is to completion (by percentage), project state, scheduled start date, and scheduled finish date.

When you click **Edit** for this portlet, a page appears with, among other portlet filter fields, the default fields having the default values shown in [Table 4-14](#).

Table 4-14. Oracle Apps 11i: Overall Status portlet filter fields

Edit Field Name	Value
Project Name	Selected
Master Project	OraApps Release 11i Upgrade Project
Summary Condition (green, yellow, red, and No Summary Condition checkboxes)	Green, yellow, and red selected; No Summary Condition deselected

OraApps: Critical Activities Portlet

The OraApps: Critical Activities portlet summarizes open critical Oracle-related requests, displaying information about type and the total number of requests for each category. This information can help identify areas of the project requiring additional resources or attention. Users can drill down on each entry to view specific requests.

When you click **Edit** for this portlet, a page appears with, among other portlet filter fields, the default fields having the default values shown in [Table 4-15](#).

Table 4-15. OraApps: Critical Activities portlet filter fields

Edit Field Name	Value
Request Type	All OraApps Request Types
Priority	Critical
Include Closed	No

Oracle Apps 11i: All Phases Portlet

The Oracle Apps 11i: All Phases portlet displays the status of all phases of the project, such as how close each phase is to completion (by percentage), project state, and the scheduled finish date.

When you click **Edit** for this portlet, a page appears with, among other portlet filter fields, the default fields having the default values shown in [Table 4-16](#).

Table 4-16. Oracle Apps 11i: All Phases portlet filter fields

Edit Field Name	Value
Project Name	Selected
Master Project and Subproject	Phase I: Project Startup Phase II: Business Operations Analysis Phase III: Prototype, GAP Analysis, High Level Design Phase IV: Development and System Test Phase V: Integration and Performance Testing Phase VI: Simulations Phase VII: Cut Over Phase VIII: Post Implementation Support
Summary Condition (green, yellow, red, and No Summary Condition checkboxes)	Green, yellow, and red selected; No Summary Condition deselected

Oracle Apps Reports Track Owner Page and Its Portlets

The Oracle Apps Reports Track Owner page (shown in *Figure 4-4*) allows track owners to view and access information needed to fulfill their business role. (For information about the track owner business role, see *Track Owner on page 37.*)

The Oracle Apps Reports Track Owner page is provided as a sample of a single track. Additional pages would be defined for other tracks.

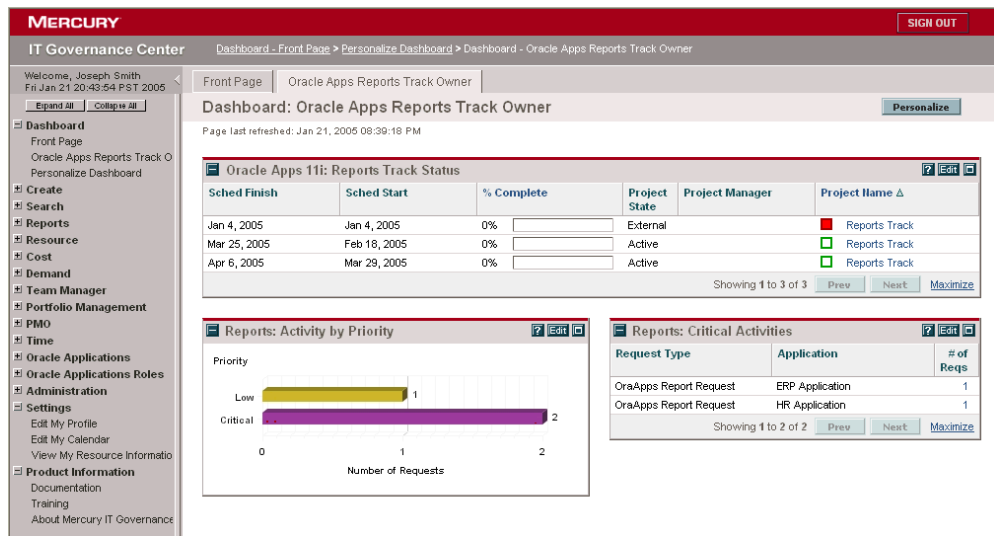


Figure 4-4. Oracle Apps Reports Track Owner page

As described in the following sections, the Oracle Apps Reports Track Owner page includes the following portlets by default:

- Oracle Apps 11i: Reports Track Status portlet
- Reports: Critical Activities portlet
- Reports: Activity by Priority portlet

Oracle Apps 11i: Reports Track Status Portlet

The Oracle Apps 11i: Reports Track Status portlet displays the status of subprojects belonging to the reports track. This information includes how close the projects are to completion (by percentage), scheduled start date, and scheduled finish date.

When you click **Edit** for this portlet, a page appears with, among other portlet filter fields, the default fields having the default values shown in [Table 4-17](#).

Table 4-17. Oracle Apps 11i: Reports Track Status portlet filter fields

Edit Field Name	Value
Project Name	Selected
Master Project and Subproject	Reports Track
Summary Condition (green, yellow, red, and No Summary Condition checkboxes)	Green, yellow, and red selected; No Summary Condition deselected

Reports: Critical Activities Portlet

The Reports: Critical Activities portlet displays critical Oracle-related requests that belong to the track, including their type, and the total number of requests for each category.

When you click **Edit** for this portlet, a page appears with, among other portlet filter fields, the default fields having the default values shown in [Table 4-18](#).

Table 4-18. Reports: Critical Activities portlet filter fields

Edit Field Name	Value
Request Type	OraApps Report Request
Priority	Critical
Include Closed	No

Reports: Activity by Priority Portlet

The Reports: Activity by Priority portlet uses a bar chart to display open Oracle-related requests that belong to the track. The requests are grouped by priority.

When you click **Edit** for this portlet, a page appears with, among other portlet filter fields, the default fields having the default values shown in [Table 4-19](#).

Table 4-19. Reporting Track: Activity by Priority portlet filter field

Edit Field Name	Value
Request Type	OraApps Report Request

Oracle Apps DBA Page and Its Portlets

The Oracle Apps DBA page (shown in [Figure 4-5](#)) allows Oracle Applications DBAs to view and access information needed to fulfill their business role. (For information about the DBA business role, see [DBA \(Database Administrator\)](#) on page 38.)

The screenshot shows the Oracle Apps DBA page with the following data:

Oracle Apps: Patch Deployment Status

Pkg #	Workflow	Priority
30085	OraApps Patch Deployment	Low
30082	OraApps Patch Deployment	Low
30056	OraApps Patch Deployment	Low
30053	OraApps Patch Deployment	Normal

Showing 1 to 4 of 4 [Prev](#) [Next](#) [Maximize](#)

Bugs reported to Oracle

No results found.

OraApps: Config. Deployments

Object Type Name	Deployments for Week of 1/16/05	Deployments for Week of 1/9/05	Deployments for Week of 1/2/05
OraApps ADADMIN	0	1	0
Generate Jar Files			
AOL:Conc Prog	0	1	0

Showing 1 to 2 of 2 [Prev](#) [Next](#) [Maximize](#)

Oracle Apps: Instance cloning queue

No results found.

Figure 4-5. Oracle Apps DBA page

As described in the following sections, the Oracle Apps Reports Track Owner page includes the following portlets by default:

- Oracle Apps: Patch Deployment Status portlet
- Bugs Reported to Oracle portlet
- OraApps Config. Deployments portlet
- Oracle Apps: Instance Cloning Queue portlet

Oracle Apps: Patch Deployment Status Portlet

The Oracle Apps: Patch Deployment Status portlet displays general information about patch deployment packages.

When you click **Edit** for this portlet, a page appears with, among other portlet filter fields, the default fields having the default values shown in [Table 4-20](#).

Table 4-20. Oracle Apps Patches Deployment Status portlet filter fields

Edit Field Name	Value
Workflow	OraApps Patch Deployment
Include Closed	No

Bugs Reported to Oracle Portlet

The Bugs Reported to Oracle portlet uses a bar chart to display all open Oracle-related issues reported to Oracle and in an OA - Pending Oracle Resolution status. The requests are grouped by priority, concisely highlighting issues requiring follow-up. Users can drill down on each entry to view specific requests.

When you click **Edit** for this portlet, a page appears with, among other portlet filter fields, the default fields having the default values shown in [Table 4-21](#).

Table 4-21. Bugs Reported to Oracle portlet filter fields

Edit Field Name	Value
Request Type	OraApps Application Issue
Status Name	OA - Pending Oracle Resolution

OraApps Config. Deployments Portlet

The OraApps Config. Deployments portlet displays a summary of the deployment activity for the past three weeks using the OraApps Customization/Configuration Deployment process. Information is grouped by object type and week of activity. Users can drill down on each entry to view specific packages.

When you click **Edit** for this portlet, a page appears with, among other portlet filter fields, the default fields having the default values shown in [Table 4-22](#).

Table 4-22. OraApps Config. Deployments portlet filter fields

Edit Field Name	Value
Workflow	OraApps Customization/Configuration Deployment
Group By	Object Type Name

Oracle Apps: Instance Cloning Queue Portlet

The Oracle Apps: Instance Cloning Queue portlet displays the list of open cloning requests. Users can drill down on each entry to view specific requests.

When you click **Edit** for this portlet, a page appears with, among other portlet filter fields, the default fields having the default values shown in [Table 4-23](#).

Table 4-23. Oracle Apps: Instance Cloning Queue portlet filter fields

Edit Field Name	Value
Request Type	OraApps Cloning Request
Eligible for My Action	No
Include Closed	No

Chapter 5 Managing Issues

In This Chapter:

- *Overview of Issues*
 - *Capturing and Tracking Issues*
 - *Processing User Requests*
 - *Reports About Issues*
 - *OraApps Critical Requests Summary Report*
 - *OraApps Apps Issues Detail Report*
 - *OraApps Apps Issues Summary Report*
 - *OraApps IT Demand Summary Report*
-

Overview of Issues

During an Oracle E-Business Suite implementation or upgrade, there is a constant influx of new problems, questions and requests. Each issue must be quickly addressed and resolved in order to prevent a growing backlog that can jeopardize the success of the project. Often these issues are captured in a variety of places: design documents, testing scripts, email discussions among team members, and help desks. Some problems, questions, and requests simply fall through the cracks and become visible only when the issue escalates.

The Extension provides the necessary tools for effective issue management. It systematizes the process of issue analysis and resolution, so that the appropriate detailed data is gathered at the right time. The Extension recognizes and intelligently routes issues based on whether they are internal, external, or vendor (that is, an Oracle bug). Through automation and notifications, the right people work on the right issues at the right time. Then, when resolution means a change to the system, the process integrates with other Extension functionality so that development and deployment maintain the same level of control and visibility.

Throughout the life of an issue, its status and progress can be monitored through the Mercury IT Governance standard interface. Reports help assess both individual issues and the aggregate status of groups of issues. Once the implementation or upgrade is complete, the issue management processes can be used on an ongoing basis for maintenance.

This chapter provides conceptual and procedural information about issue management and the entities (request types, workflows, and report types) relating to issue management provided in the Extension.

Capturing and Tracking Issues

The OraApps Application Issue request type is used to capture and track the issues at an Oracle E-Business Suite implementation site. It provides a means for entering specific information needed to resolve issues. This request type works in conjunction with the OraApps Application Issue workflow.

Someone creates a request in order to resolve a specific problem. The request originator is prompted for the information needed to successfully identify and analyze the issue. This information focuses on the Oracle environment in addition to project and business information. After entering all necessary information, the originator submits the request.

The following items are examples of the kind of information the request type uses:

- Oracle environment information:
 - Application in which the problem occurred
 - Form being used
 - Operating system being used
- Project and business information:
 - Business area affected
 - Perceived business benefit for resolving the issue
 - Source of the issue
 - Estimated technical impact on the organization
 - Attachments (for example, relevant documents and image files)

When all the relevant information for an issue has been entered into the OraApps Application Issue request type, it can be analyzed and processed through to resolution.

For More Information

For more information, see [OraApps Application Issue Request Type on page 287](#).

Processing User Requests

Once an OraApps Application Issue request type has been submitted, it is routed through a business process of approvals, decisions, and actions as defined in the OraApps Application Issue workflow. This workflow involves a number of steps that require review and analysis by users.

You can locate and open issues through the standard interface. The Extension provides pre-configured portlets based on your business role that help you manage Oracle-related requests. A project manager, for example, might use the OraApps: Critical Activities portlet to view open issues. You can also configure additional portlets or pages in the Dashboard.

Figure 5-1 shows the OraApps Application Issue workflow.

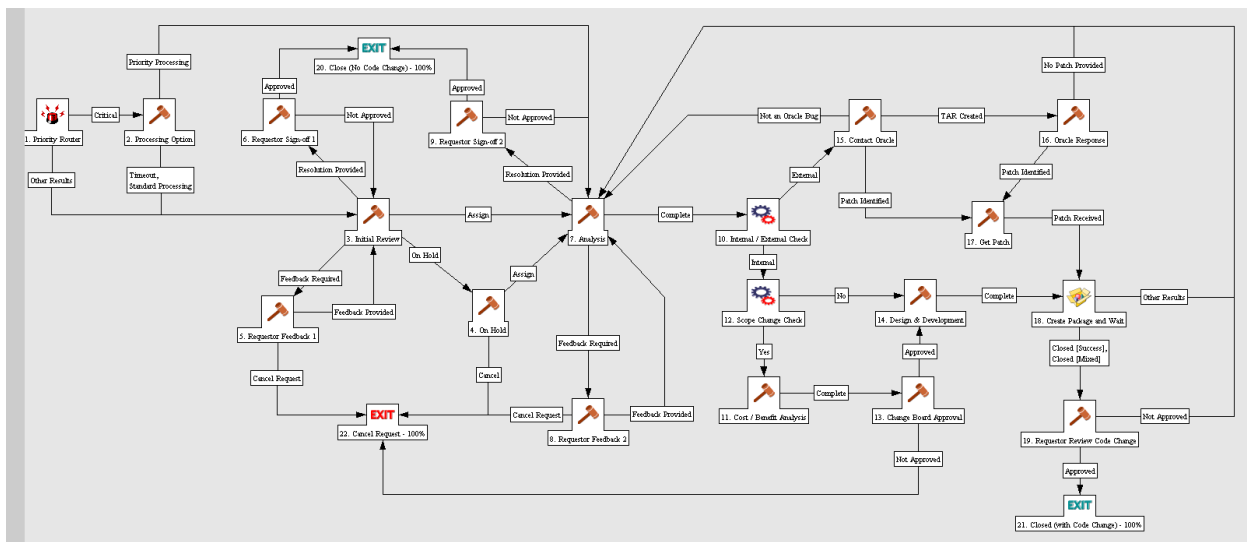


Figure 5-1. OraApps Application Issue workflow

The OraApps Application Issue workflow automatically evaluates issues based on the priority level assigned to them by their requestor. Issues designated as critical are routed to project managers to decide the correct processing option, priority or standard. All other issues not having a priority level of critical go to the Initial Review step.

Track owners can request more information or provide a resolution to the issue at the Initial Review step. If not resolved, the issue goes through several steps where the requestor is asked for more feedback. Requestors can sign off if they are satisfied with the resolution created by the track owner or team lead. A

business analyst determines whether the resolution requires a code or configuration change, and if this change is internal or external.

If the code or configuration change is internal, a cost/benefit analysis is performed and the scope change board decides whether to go forward with it or not. If approved, the code or configuration change goes to the design and development organization.

If the code or configuration change is external, then Oracle is contacted for the appropriate patch.

Once the patch is received or the development is complete, the change goes through the standard deployment cycle. The original requestor is notified and can close the issue as successful, or send it back to the business analyst for further review.

For More Information

For more information, see [OraApps Application Issue Workflow on page 324](#).

Reports About Issues

Using the Extension, you can run the following reports about issues:

- OraApps Critical Requests Summary Report
- OraApps Issues Detail Report
- OraApps Issues Summary Report
- OraApps IT Demand Summary Report

OraApps Critical Requests Summary Report

The OraApps Critical Requests Summary Report displays a summary of all the Extension-related requests that have a priority level of critical. The report breaks down this information by:

- Total number of open critical requests
- Number of critical requests opened in the current week
- Number of critical requests closed in the current week
- Total number of closed critical requests

The report also lists detailed information for each issue.

For More Information

For more information, see [OraApps Critical Requests Summary Report on page 391](#).

OraApps Apps Issues Detail Report

The OraApps Apps Issues Detail Report provides details of requests created using OraApps Application Issue request type. It lets you filter for requests by values in the standard header fields and specific detail fields such as TAR #, Business Benefit, and Environment.

For More Information

For more information, see [OraApps Apps Issues Detail Report on page 383](#).

OraApps Apps Issues Summary Report

The OraApps Apps Issues Summary Report provides summary information on requests for the OraApps Application Issue request type. It lets you filter for requests by values in the standard header fields and specific detail fields such as TAR #, Business Benefit, and Environment. The Group By field lets you summarize requests using many of these detail fields.

For More Information

For more information, see [OraApps Apps Issues Summary Report on page 387](#).

OraApps IT Demand Summary Report

The OraApps IT Demand Summary Report displays a summary of all Extension-related requests. This report gives the total counts for groups of issues that match the selection criteria. Users can group selected issues into as many as five categories and get the subtotals for each group.

Project Managers for Oracle-related projects can use this report to determine priority, assigned group, and total number of project-related request types.

For More Information

For more information, see [OraApps IT Demand Summary Report on page 393](#).

Chapter 6 Managing Projects

In This Chapter:

- *Overview of Projects*
 - *Planning an Upgrade Strategy*
 - *Including Activities in a Project*
 - *Integrating Mercury Project Management and Demand Management*
 - *Performing a Gap Analysis*
 - *Using the OraApps Release 11i Upgrade Project Template*
 - *Overview of Project Template*
 - *Project Template Structure*
 - *Customizing Project Plans*
 - *Reporting Status*
 - *OraApps Status Update Request Request Type*
 - *OraApps Status Update Request Workflow*
 - *Reports About Project-Related Status*
 - *Configuring Project Management*
 - *Configuring Request Types*
 - *Configuring Workflows*
 - *Configuring Report Types*
-

Overview of Projects

Upgrading to Oracle E-Business Suite Release 11i is a significant effort. The upgrade usually involves a project team crossing a variety of departments and functional and technical backgrounds, often including external system integrators. Complicating the matter are the many architectural, functional, and process changes in Release 11i. Existing customizations and Extensions need to be assessed and removed or amended. In addition, the upgrade to Release 11i might include implementation of its new modules (for example, Oracle CRM modules).

All this complexity and effort require central control and wide visibility of the upgrade project. Tasks and deliverables need to be closely monitored. Common processes need to be enforced across many teams. Open issues related to the project need to be tightly managed. Roles and responsibilities for each issue and task need to be clearly defined.

The Extension includes content to help you manage your Oracle E-Business Suite upgrade or implementation project. The key entity for this effort, the OraApps Release11i Upgrade project template, provides a sample outline you can modify to suit your organization's needs. The activities in the project template have been consolidated from actual Release 11i upgrade projects as well as other implementations and upgrades performed by Mercury and its partners, providing you with a significant jump-start to your project activities.

With Mercury Project Management and the OraApps Release 11i Upgrade project template, you can bring your Oracle 11i project to fruition, allowing project team members to directly update the project as they go through their tasks, reviews, and approvals. With the Mercury project entity as a central information repository, team members can spend more time working on tasks and less time providing detailed status information on their own tasks and looking for status information about their dependent tasks. Project deliverables are attached to related tasks, allowing team members to quickly access current information.

Mercury Project Management can also serve as the integration point to tie together all activities relating to the project:

- Packages and requests that require their own detailed processes can be associated with the project entity.
- Issues raised from any source can be associated with appropriate project tasks. When a task requires its own detailed process of analysis and approvals, it can be integrated with a Mercury request entity such that there

is a dependency between the two. For example, a gap analysis request can be linked to the project.

- Team members can use the Mercury IT Governance standard interface to access their current task list, reducing the potential for lost productivity. Track owners and project managers can quickly assess the status of their areas of control, allowing them to focus on areas requiring attention.
- Reports tailored for the Oracle E-Business Suite environment assist you in identifying situations requiring action, such as unassigned tasks that need to begin or unresolved critical issues.
- The Extension includes the OraApps Status Update Request request type, which can quickly request project status. Automated executions and notifications reduce the need for manual tracking down of status, allowing more time to be spent on tasks.

Planning an Upgrade Strategy

The OraApps Release 11i Upgrade project template, provided with the Extension, is modeled after activities that have been used in real-world upgrade projects in Oracle E-Business Suite systems. When planning your upgrade, open the OraApps Release 11i Upgrade project template and decide how these activities apply to your particular situation before beginning the upgrade process. The project template is a model or blueprint for upgrading; it does not cover every possible configuration or situation.

Planning an upgrade strategy involves including activities in a project and integrating Mercury Project Management and Mercury Demand Management, as discussed in the following sections:

Including Activities in a Project

As you plan your project, include the following kinds of activities:

- Determine the roles for the project and the responsibilities for each role.
- Determine your installation requirements (both hardware and software).
- Identify which business functions will be used in the new Oracle E-Business Suite release.

- Identify training and documentation needed for your organization to learn the new Oracle E-Business Suite release.
- Understand how your business requirements will be supported by the new Oracle E-Business Suite release.
- Identify major project dependencies.

After deciding on your strategy for these aspects of the upgrade, modify the project plan accordingly.

Integrating Mercury Project Management and Demand Management

Your upgrade strategy also includes integrating the Mercury Project Management capabilities with the issue tracking capabilities in Mercury Demand Management. Demand Management requests and Project Management projects and tasks can be linked to each other through a variety of dependent and informational relationships. This allows requests to be part of project initiatives and gives instant visibility to the detailed activities that support the upgrade project:

- You can link requests to projects in the **Reference** tab in the Project Information window in the Workbench or from the Reference section of the Project Detail page in the Dashboard.
- You can create tasks from requests in the Workbench. A dependency relationship is established between the task and the request.

The Dashboard lets you view references related to tasks and projects using the Project References portlet. Add the portlet to your Dashboard and then personalize it to show references that are relevant to upgrade activities.

Integrating Mercury Project Management and Mercury Demand Management, and then personalizing the Dashboard to include the proper portlet gives you quick access to clear information about the following aspects of your upgrade project:

- Current percentage complete of tasks and requests
- Reference types and their relationships
- When the references were added
- Whether the references are preventing actions on tasks

Performing a Gap Analysis

An Oracle E-Business Suite upgrade project can result in significant changes to current business processes and in additions or customizations to the Oracle environment to accommodate the changed business processes.

Assessment of the differences between existing business processes and functionality or capability provided by a packaged application is referred to as “gap analysis.” Each identified difference, or “gap,” is analyzed to determine how it will be handled; for example, Oracle functionality might be extended or an existing business process might be changed.

Members of the upgrade project team who have the business role of project manager, track owner, or project sponsor can request a gap analysis using the OraApps GAP Analysis Request request type. This request type captures business information needed to analyze a gap and uses the OraApps GAP Analysis Process workflow to enforce a process that includes approval, assignment, and analysis steps for the gap analysis.

The project team uses information gathered from the gap analysis to help plan their upgrade tasks. The analysis gives the team a better picture of the resources, time, and effort needed to successfully complete the upgrade.

Steps in the OraApps GAP Analysis Process workflow include the following:

- A project manager assigns a priority and a business lead to the request.
- The business lead can either reject the request (which sends it back to the project manager) or accept it. If the request is accepted, the business lead assigns a team to the gap analysis.
- The gap analysis team determines where the gap is in the Oracle functionality. They also determine whether the gap can be closed by using a code, setup, or process change.
- The gap analysis team creates either an enhancement request or a setup change.
- The gap analysis request is closed.

For information on configuring this workflow process, see [OraApps GAP Analysis Request Request Type on page 162](#) and [OraApps GAP Analysis Process Workflow on page 164](#).

Using the OraApps Release 11i Upgrade Project Template

This section provides an overview of the OraApps Release 11i Upgrade project template and describes its structure and how to customize project plans.

Overview of Project Template

The OraApps Release 11i Upgrade project template helps maintain visibility and control of your Oracle E-Business Suite Release 11i upgrade. The project template includes activities typically performed during an upgrade. These activities have been consolidated from actual Release 11i upgrade projects, as well other implementations and upgrades performed by Mercury and its partners.

The OraApps Release 11i Upgrade project template, described in this section, is a sample outline. You can modify the project template to suit your organization's needs.



The project template and reports described in this section are delivered with the Extension as reference (REFERENCE) entities only. Make copies and name them as documented in this section.

Project Template Structure

The OraApps Release 11i Upgrade project template contains eight phases corresponding to a group of related activities and milestones in a typical Oracle E-Business Suite upgrade. Each phase is a subproject, and each of the subprojects contains sample tasks for common functional areas such as demand planning, order management, and manufacturing. The project template concentrates on activities around the actual upgrade, but can be used as a subproject for projects with a bigger scope.

You can configure a copy of the project template to reflect your organization's needs. For example, if you have not implemented Oracle Manufacturing, you can delete the manufacturing sections.

The OraApps Release11i Upgrade project template is shown in *Figure 6-1*. Columns in the project template are described in *Table 6-1*.

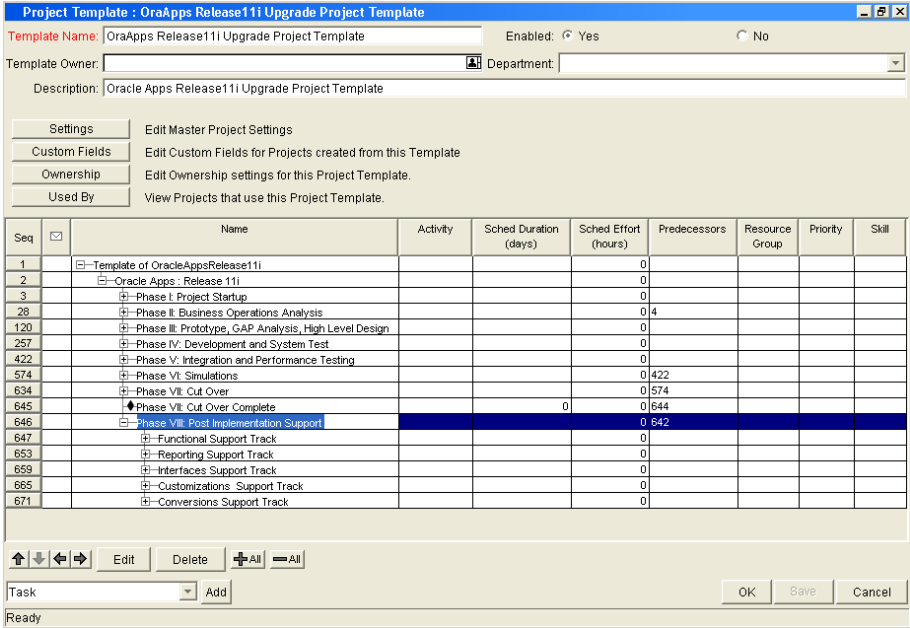


Figure 6-1. OraApps Release11i Upgrade project template

Table 6-1. OraApps Release11i Upgrade project template fields

Project Template Field	Description
Seq	Sequence numbers of the tasks. There are 676 tasks in the project template as delivered.
Name	Name of the parent or child task.
Activity	Project activity.
Sched Duration (days)	Typical amount of time in days required to complete the task. When a project is created from the template, this value allows Mercury Project Management to schedule the task.
Sched Effort (hours)	Typical effort in hours required to complete the task in an average upgrade project. This value is used to show the roll-up of effort required to complete the project.
Predecessors	Tasks that need to be completed before the start of the target task. Predecessors can be individual tasks or subprojects. Critical predecessors are included in the project template to illustrate the relationship among tasks.
Resource Group	Name of the security groups associated with the tasks or subprojects.

Table 6-1. *OraApps Release11i Upgrade project template fields [continued]*

Project Template Field	Description
Priority	Relative importance of the task. This project template does not make any assumption about the relative importance of a particular task.
Skill	Type of skill required to accomplish the project task.



Note

The rolled-up durations shown on the project template change once the project is created and resourced, as the scheduling engine takes into consideration dependencies and information unavailable to the project template.

The project template has the following phases, as described in the following sections:

- Phase I: Project Startup
- Phase II: Business Operations Analysis
- Phase III: Prototype, Gap Analysis, High Level Design
- Phase IV: Development and System Test
- Phase V: Integration and Performance Testing
- Phase VI: Simulations
- Phase VII: Cutover
- Phase VIII: Post-Implementation Support

Phase I: Project Startup

This phase consists of activities pertaining to the overall structure of the upgrade. These activities are done in the weeks leading to kickoff, and include the formal kickoff meeting with the project team. This phase lays the groundwork for the project so the team members understand the tasks involved and can efficiently complete the tasks that are assigned to them.

This phase includes milestones for the completion of each major grouping of activities. You may want to add or modify these milestones based on your tracking needs.

Key deliverables for this phase are a project readiness document, a system architecture, a project plan, and a kickoff presentation.

Phase II: Business Operations Analysis

This phase clarifies and consolidates information about the current installation, identifies key business requirements for the upgraded system, and assesses the potential to meet those requirements in the upgraded Oracle E-Business Suite instance. Current installation information includes a catalog of customizations in place in the Oracle E-Business Suite instance, and processes and interface points for business functions to be newly incorporated into the Oracle E-Business Suite instance. The Oracle functionality is analyzed to determine how well it meets the business requirements, including allowing for the elimination of existing customizations or Extensions. In a typical upgrade, most of the time spent in this phase is focused on learning the new Oracle E-Business Suite release and identifying functional gaps.

Also included are milestones marking the completion of activities for each track or sample functional area. You may want to add or modify these milestones for signoff on your most critical or interdependent Oracle Applications modules.

The key deliverables for this phase are a gap list and as-is process flows.

Phase III: Prototype, GAP Analysis, High Level Design

This phase begins with a gap analysis for various tracks from Phase II. As gaps are identified, they are analyzed in greater detail in order to determine solutions. High-level designs are created for each solution, and any required process changes are incorporated in the to-be process documents. This includes customizations and Extensions, reports, interfaces, and conversions.

Typically a representative from each business area formally accepts designs. At this point, all remaining technical effort should be known. Generally these activities are conducted per business functional area per gap or per group of interdependent gaps. Often reviewers include representatives from all dependent functional areas, as well as a technical reviewer to ensure technical feasibility and consistency.

The project template includes:

- Tasks for gap analysis
- Solution design and setup
- Profiles options for each track and sample functional area

Also included are milestones marking the completion of activities for each track or sample functional area. You may want to add or modify these

milestones based on your tracking needs. For example, you may want to include milestones for signoffs of to-be process flows, prioritization of interfaces, or reports readiness.

Key deliverables for this phase are final to-be process flows, solution designs for all gaps, setup and profile options, and signoff on the unit (business) test scripts.



The OraApps GAP Analysis Request request type is designed to manage the process each gap must pass through to complete successfully. This can be done by attaching requests using this request type to the appropriate tasks in the plan.

Phase IV: Development and System Test

At this phase the detailed designs are finalized for conversion, interfaces, reports, and customizations. Coding of conversion, interfaces, reports and customizations is done, and unit tests are run.

The project template includes milestones marking the completion of activities for each track or sample functional area. You may want to add to or modify these milestones based on your tracking needs. For example, you may want to include milestones for signoffs of exit criteria, successful completion of initial testing, or signoffs of training plans.

The key deliverables for this phase are unit-tested interfaces, reports, conversions, customizations, enhancements, and signoffs of exit criteria.



You can use the specialized development request types that are part of the Extension to facilitate development and increase visibility to development efforts on the project. These request types are:

- OraApps Conversion request type
- OraApps Enhancement request type
- OraApps Interfaces request type
- OraApps Reports request type
- OraApps Setup Change request type

You can use the OraApps Customization/Configuration Deployment workflow to manage the code migrations for each gap by attaching packages using this workflow to the appropriate tasks in the plan.

Phase V: Integration and Performance Testing

In this phase all required test scripts for integration and performance testing are created. This may involve consolidation or adaptation of existing test scripts, generation of new test scripts to bridge between different processes, and design of sample test data. For performance test scripts, identification of representative transactions and loads should be completed. Baseline performance should also be defined, if possible.

Integration testing is performed to ensure that business processes, developed code, interfaces, and conversions function as expected. This is usually the first time these elements are tested end-to-end in the same instance. Performance testing is performed to ensure that system performance is acceptable for users and to perform final tuning of programs. Disaster recovery processes should be defined.

The project template provides a single combined phase for integration and performance testing. Additional rounds of testing (such as acceptance testing) should be planned and may be added to the project template. Also included are milestones marking the completion of:

- Build
- Data population using conversions
- Functional testing
- Exit criteria signoff for each round of testing
- Project management and infrastructure tasks associated with the phase

You may want to add to or modify these milestones based on your tracking needs.

The key deliverables for this phase are a fully operational system that is at least 90% complete, and a document identifying production readiness.

Phase VI: Simulations

This phase is a dry run of the cutover event in order to streamline the implementation time needed to implement Oracle Release 11i and minimize the impact to your business. This event is usually conducted around the clock, and is used to set a baseline so that management knows throughout the cutover event how the project team's status compares to agreed upon schedules. The simulation is also used to ensure that the workload is balanced and all tasks can be accomplished by those assigned.

The project template includes milestones marking the completion of activities for each track or sample functional area. You may want to add to or modify these milestones based on your tracking needs. For example, you may want to include milestones to mark completion of each simulated iteration, or to identify a formal “go-live” signoff milestone after simulations are complete.

The key deliverable for this phase is the final cutover plan, including an estimate of the time required for the implementation of the Release 11i upgrade.

Phase VII: Cut Over

This phase addresses the actual upgrade and cutover to the new production system. The plans refined in Phase VI (Simulations) are executed against the production environments.

The project template includes milestones to mark the completion of cutover activities and formal go-live system acceptance. You may want to add to or modify these milestones based on your tracking needs.

The key deliverables from this phase are a committed go-live decision and signed acceptance of the production system.

Phase VIII: Post Implementation Support

Once the upgrade has been completed from a technical standpoint, it is common to go through a period of additional training and heightened support as users become accustomed to the new system.

The support activities for functional, reporting, interface, customization support track, and conversion support tracks include:

- Issue resolution
- Bug fixes
- Knowledge transfer

Customizing Project Plans

Every project plan in Mercury Project Management can be customized. The custom settings in the plan obtain their values from various custom settings in the project template, making creation of a plan more efficient.

The values defined in the OraApps Release 11i Upgrade project template represent typical values. You can change these values by creating a copy of the project template and then changing the values in the copy.

For more information about setting up and using Mercury project templates, see the *Mercury Project Management User's Guide*.

The following sections discuss the following subjects:

- Selecting information to track and display
- Configuring exception behavior
- Configuring indicators
- Configuring scheduling options
- Configuring security restrictions

Selecting Information to Track and Display

Project managers can use the Workbench to choose the information they want to track over the course of the upgrade. Eligible project team members can then enter information in the enabled fields. As seen by clicking the **Settings** button on the OraApps Release 11i Upgrade project template, the Extension includes planning fields and activity fields, as described in the following sections. In [Table 6-2](#) and [Table 6-3](#), the Workbench Only column indicates whether the fields apply to the Workbench only, not the standard interface.

Planning Fields

Table 6-2 describes the available planning fields in the OraApps Release11i Upgrade project template.

Table 6-2. OraApps Release11i Upgrade project template planning fields

Planning Field	Workbench Only?	Default Value
Department	No	Disabled
Description	No	Disabled
Last Update Date	No	Disabled
Priority	No	Disabled
Project/Task #	No	Disabled
Resource Group	No	Disabled
Scheduled Effort	No	Disabled
Skill	No	Disabled
Activity	No	Disabled

Activity Fields

Table 6-3 shows the available activity fields in the OraApps Release11i Upgrade project template.

Table 6-3. OraApps Release11i Upgrade project template activity fields

Activity Field	Workbench Only?	Default Value
%Complete	No	Enabled
Actual Duration	No	Disabled
Actual Effort	No	Disabled
Actual Finish Date	No	Disabled
Actual Start Date	No	Disabled
Confidence	No	Disabled
Estimated Finish Date	No	Disabled
Estimated Remaining Duration	No	Disabled
Estimate Remaining Effort	No	Disabled

Configuring Exception Behavior

A project manager can specify which exception rules are to be enabled, as well as how many days it takes to trigger them. A project manager can also configure the summary condition, which is a color-coded indicator of the health of a project. The summary condition watches for exceptions that have been triggered, and changes color according to rules that can be set by the project manager.

The OraApps Release11i Upgrade project template presets the exception behavior color codes to the values show in [Table 6-4](#).

Table 6-4. OraApps Release11i Upgrade project template exception behavior

Color Code in Project Summary View	Percentage of Tasks with the Exception Condition
Red	60%
Yellow	30%

As seen by clicking the **Settings** button on the OraApps Release 11i Upgrade project template, the project template is equipped with independently configurable exception rules on the **Exception Rules** tab.

[Table 6-5](#) lists and defines the exception rules in the OraApps Release11i Upgrade project template.

Table 6-5. OraApps Release11i Upgrade project template exception rules

Exception Rule	Description
Unassigned Tasks	The task has no resource assigned and the start date is in <i>x</i> days (you specify <i>x</i>). The task is included in the summary condition.
Late Tasks or Milestones	The task or milestone is not yet complete and the current date is <i>x</i> days (you specify <i>x</i>) past the scheduled finish date.
Tasks Starting Late	The scheduled start date is <i>x</i> days prior to the current date (you specify <i>x</i>) and the task is not yet in progress. The task is included in the summary condition.
Late Estimated Finish	(By default, this rule is disabled in the project template.) The estimated finish date is more than <i>x</i> days (you specify <i>x</i>) past the scheduled finish date.
Not Enough Time Left	(By default, this rule is disabled in the project template.) The estimated remaining duration exceeds the time left between the current date and the scheduled finish date by <i>x</i> days (you specify <i>x</i>).
Predecessor Has Exception	At least one predecessor currently has an exception.
Critical Path Running Late	The critical path to this milestone is running late.

Configuring Project Plan Indicators

Mercury Project Management provides project plan indicators to give users a better “at-a-glance” overview of projects in progress. The indicators, listed and described in [Table 6-6](#), can be configured to be displayed at the master project level.

Table 6-6. OraApps Release11i Upgrade project template project plan indicators

Indicator	Value	Description
Constraint	Disabled	Displays any constraints on scheduling for a particular item.
Custom Fields Indicator	Disabled	Displays any projects for which custom fields have been defined.
Exception Indicator	Enabled	Displays any tasks that have triggered exceptions, and also displays the summary condition for projects.
Notification Indicator	Disabled	Displays tasks for which notifications have been defined.
Scheduling Indicator	Disabled	Displays any updated tasks or projects that might require rescheduling.

Configuring Scheduling Options

Project managers can set different scheduling configurations, depending on the user or project's level of resource dependency. The scheduling options also allow for the creation of schedules based on actual start and finish dates. The default configuration does not use actual dates.

In the OraApps Release11i Upgrade project template, scheduling options are set up as shown in [Table 6-7](#).

Table 6-7. OraApps Release11i Upgrade project template scheduling options

Scheduling Option	Description
Default Date from which to Schedule	Blank.
Schedule Project Using	Scheduled duration, predecessors, and calendars.
Only Include Days in the Schedule When	All resources are available.

Configuring Security Restrictions

Project managers can set security restrictions on a master project such that only those who are involved in the project can access its information. Someone who is not a manager or participant in a project will not be able to open it or edit it. The project will not appear in any searches run from the Project Workbench or Project Search page.

The default setting for the OraApps Release11i Upgrade project template is unrestricted (that is, all users can view the projects and tasks).

Reporting Status

This section contains guidance about various entities in the Extension that are used in reporting status. For more information on the entities discussed in this section, see the appendices in this document.

The following sections discuss the following subjects:

- OraApps Status Update Request request type
- OraApps Status Update Request workflow
- Report types for project-related status

OraApps Status Update Request Request Type

The OraApps Status Update Request request type automates the process of gathering and reviewing status reports. It lets managers request information, automatically notifies group members of requirements for information and monitors them to make sure that tasks are completed. When the tasks for completing a report have been done, reviewers are automatically notified. All involved parties have access to the latest version of the report.

This request type allows users to enter basic requestor information and determine which project teams need to provide status updates. The requestor can indicate the required date for the information and which individual is directly responsible for the update from each team (this responsibility information can also be defaulted). Once the request is submitted, it goes through an update request process that routes the request to the appropriate teams and individuals. The status updates are attached to the request, so all individuals with appropriate access can use the request as the repository for the project status information for the given time period.

The OraApps Status Update Request request type is designed to work with the OraApps Status Update Request workflow, described in [OraApps Status Update Request Workflow](#) on page 159.

For More Information

For more information, see [OraApps Status Update Request](#) on page 315.

OraApps Status Update Request Workflow

The OraApps Status Update Request workflow provides a process to automatically notify group members, follow up with them until information is received, contact the appropriate reviewers, and distribute the latest version of the reports.

Once a status update request is made, the OraApps Status Update Request workflow splits into parallel branches for each project subteam. If a status update is requested for a specific subteam, the designated team lead is notified of the request. The team lead is then continually reminded of this request until the information is provided to the request.

Once all the requested status updates are provided, the requestor is notified and can review the individual information directly from the request. The entire project management team or the entire project team can review the information, as well.

For More Information

For more information, see [OraApps Status Update Request Workflow on page 368](#).

Reports About Project-Related Status

This section discusses the following reports on project-related status:

- OraApps Critical Requests Summary Report
- OraApps IT Demand Summary Report
- OraApps Ready Task List Report
- OraApps Unassigned Ready Tasks Report

OraApps Critical Requests Summary Report

The OraApps Critical Requests Summary Report displays a summary of Extension-related requests that have a priority level of critical. Project managers for Oracle projects need to identify critical issues quickly so they can either add more resources or adjust the schedule.

The report breaks down the information by:

- Total number of open critical requests
- Total number of critical requests opened in the current week
- Total number of critical requests closed in the current week
- Total number of closed critical issues

The report also lists detailed information for each issue.

For More Information

For more information, see [OraApps Critical Requests Summary Report on page 391](#).

OraApps IT Demand Summary Report

The OraApps IT Demand Summary Report displays a summary of all Extension-related requests. This report gives the total counts for requests that match the selection criteria. Users can categorize selected requests into as many as five categories and get subtotals for each group.

Project managers for Oracle projects can use this report to determine the priority, assigned group, and total number of project-related request types.

For More Information

For more information, see [OraApps IT Demand Summary Report on page 393](#).

OraApps Ready Task List Report

The OraApps Ready Task List Report displays Oracle project tasks that are in the Ready state. It also shows which users or groups have been assigned to tasks for a given project.

Project managers for Oracle projects can use this report to compare a task's scheduled start date, scheduled finish date, and scheduled duration to actual progress in the project.

For More Information

For more information, see [OraApps Ready Task List Report on page 396](#).

OraApps Unassigned Ready Tasks Report

The OraApps Unassigned Ready Tasks Report displays Oracle project tasks currently in the Ready state but not yet assigned to a group or user.

Project managers for Oracle projects can use this report to assign groups or individual users to unassigned tasks, preventing them from falling behind schedule.

For More Information

For more information, see [OraApps Unassigned Ready Tasks Report on page 398](#).

Configuring Project Management

The following sections discuss configuring request types, workflows, and report types for Oracle Applications project management.

Configuring Request Types

This section provides conceptual and procedural information about the following request types:

- OraApps GAP Analysis Request request type
- OraApps Status Update Request request type

OraApps GAP Analysis Request Request Type

The OraApps GAP Analysis Request request type initiates the process of requesting a gap analysis for an Oracle system. Project managers, track owners, and project sponsors can use this request type to create a gap analysis. Once a request for a gap analysis has been created, it needs to be attached to relevant Release 11i project tasks.

The OraApps GAP Analysis Request request type captures business details and works with the OraApps GAP Analysis Process workflow, which consists of approval, assignment, and analysis steps.

To configure the OraApps GAP Analysis Request request type:

1. Review the security information on the workflow and modify it to meet your organization's structure and needs.
2. Review the security information on the request type (user access and field security) and modify it to meet your organization's structure and needs. For example, you may want to limit the ability to create requests to specific groups of users.
3. In the **Request Type Rules** tab of the Request Type window, set the default workflow for the request type to the OraApps GAP Analysis Process workflow.
4. In the **Workflow** tab of the Request Type window, restrict the request type to use only this workflow.
5. Enable the request type.

For More Information

For more information about this request type, see [OraApps GAP Analysis Request Request Type](#) on page 303.

OraApps Status Update Request Request Type

To configure the OraApps Status Update Request request type:

1. Decide which of your subteams will do status reports, and for each subteam:
 - a. Create Status and Lead fields for subteams not already defined.
 - b. Delete any Status and Lead fields not relevant to your project.
2. Define the defaults for each Status field (that is, Requested or Not Requested).
3. Identify team leads for the project and define them as default values in the Team Lead fields.
4. Review the security information on the request type (User Access and Field Security) and modify it to meet your organization's structure and needs. For example, you may want to limit the ability to create status update requests to specific groups of users.

For More Information

For more information about this request type, see [OraApps Status Update Request](#) on page 315.

Configuring Workflows

This section provides conceptual and procedural information about the following workflows:

- OraApps GAP Analysis Process workflow
- OraApps Status Update Request workflow

OraApps GAP Analysis Process Workflow

As described in [Performing a Gap Analysis on page 145](#), members of an upgrade project team can request a gap analysis using the OraAppsGAP Analysis Request request type. This request type uses the OraApps GAP Analysis Process workflow to carry out the gap analysis.

Using information in the gap analysis results, the gap analysis team determines the best action:

- If they can find matching functionality in Oracle, they send this information back to the business lead, who decides whether or not to accept the functionality match.
- If they determine that a code, setup, or process change is required for bridging the gap, they manually create an enhancement request or a setup change request.

Following one of these two actions, the gap analysis is closed.

To configure the OraApps GAP Analysis Process workflow:

1. Review the security information in the workflow and modify it to meet your organization's structure and needs.
2. In the **Request Type Rules** tab of the Request Type window for the OraApps GAP Analysis Request, set the default workflow to the OraApps GAP Analysis Process workflow.
3. In the **Workflow** tab of the Request Type window, restrict the request type to use only this workflow.
4. Modify the workflow to define any additional processing notifications required.
5. Modify the workflow to reflect localized business practices.

For More Information

For more information about this workflow, see [OraApps GAP Analysis Process Workflow](#) on page 348.

OraApps Status Update Request Workflow

To configure the OraApps Status Update Request workflow:

1. Verify the security groups for each step and modify them, if necessary, to fit your organization's needs.
2. Remove workflow branches for any status or lead fields that have been removed from the request type.
3. Add branches, if required:
 - a. Copy existing branches.
 - b. Update the step names.

Be sure the name of the resolution step token matches the name of the request detail field token for the status field.
 - c. Update the notifications.

For More Information

For more information about this workflow, see [OraApps Status Update Request Workflow](#) on page 368.

Configuring Report Types

This section contains information about configuring the following report types included with the Extension:

- OraApps IT Demand Summary Report
- OraApps Critical Requests Summary Report
- OraApps Ready Task List Report
- OraApps Unassigned Ready Tasks Report

To configure these report types:

1. Copy the reference versions of the reports and name them using the names listed above.
2. Enable them for use.
3. Configure any security restrictions required.

By default, these reports are available for all users.

4. Use hidden fields to restrict the data provided by the reports.

For More Information

For more information about these report types, see:

- [OraApps IT Demand Summary Report on page 393](#)
- [OraApps Critical Requests Summary Report on page 391](#)
- [OraApps Ready Task List Report on page 396](#)
- [OraApps Unassigned Ready Tasks Report on page 398](#)

Chapter 7 Managing Changes

In This Chapter:

- *Overview of Changes*
 - *Requesting Changes*
 - *Requesting New Reports*
 - *Requesting New Interfaces*
 - *Requesting Conversions*
 - *Requesting Enhancements*
 - *Requesting Setup Changes*
 - *Deploying Oracle Changes*
 - *Overview of AOL and GL Migration*
 - *AOL or GL Migration Execution*
 - *AOL or GL Concurrent Request Log*
-

Overview of Changes

Oracle E-Business Suite has a flexible architecture that allows companies to shape the system to their business needs through both configuration and customization. In a typical implementation or upgrade, a significant amount of work goes into designing, developing, and rolling out new or updated configurations and changes. Even companies primarily trying to use only standard functionality find themselves with high volumes of configurations (for example, security setup) and customizations (for example, custom reports). After implementation, business processes continue to evolve, and Oracle systems needs to evolve, as well.

The success of an Oracle implementation, upgrade, or ongoing production support depends on a stable, efficient process for managing these configurations and customizations. Without such a process, determining which changes have been applied to which instances can seem to be a nearly impossible task. Further, the potential for inaccurate application of changes to a given instance is high. Limiting who can apply changes reduces inaccuracies, but requires considerable time from valuable people.

The Extension provides processes and automation to help you maintain visibility to and control of the changes to your system. This packaged content has been derived from first-hand, in-depth experience by Mercury and its partners in implementing, upgrading, and supporting Oracle E-Business Suite, from Release 10 character-mode through Release 11i. These components allow you to quickly get control of your changes, from initial requirements to final deployment to your production system.

Maintaining control over how changes are developed is important. The Extension includes request types and workflows that model various types of development for Oracle, such as interfaces and reports. This helps with efficient use of resources and reduces the likelihood that changes will need rework late in the development cycle. When changes need to be deployed, the deployment workflow and object types in the Extension enforce best practices for deployments, testing, and approvals.

The Extension provides special handling for AOL and GL objects, integrating with the Object Migrator and GL Migrator products to migrate AOL and GL setup data between Oracle E-Business Suite instances.

For More Information

For more information about Mercury Change Management, see the *Mercury Change Management User's Guide* document and the *Mercury Change Management: Configuring a Deployment System* document.

Requesting Changes

Request types and workflows provided by the Extension take users through the process of requesting, validating, and developing changes to an Oracle system. Users can request enhancements, new interfaces, and new reports, and can set up changes and conversions. The processes are initiated by creating a request of the appropriate request type and are moved through the necessary activities by the corresponding workflows.

The following sections discuss the following kinds of changes users can request:

- Requesting new reports
- Requesting new interfaces
- Requesting conversions
- Requesting enhancements
- Requesting setup changes

Requesting New Reports

Oracle Applications includes standard reports that provide users with information about the system's setup and transaction activity. These reports do not always fully meet the needs of an organization. For example, you may need to include additional information not provided in standard reports, such as descriptive flexfield data. Or corporate standards may dictate a different report format than Oracle provides. Regardless of the reason, in these cases new or modified reports need to be developed.

In the Extension, users can request creation of a new report with the OraApps Report Request request type. This request type works in conjunction with the OraApps Reports Process workflow to enforce proven practices for report development, and to help manage IT workload effectively. Once the request is initiated, it captures the following information at the appropriate point in the process:

- Report name and description
- Whether a new report is needed
- The date by which the report must be completed
- Technical analysis
- Technical impact of the report

The workflow sends the request through approval, analysis, and assignment steps. The requestor creates the high-level requirements for the report, which is then signed off. After being prioritized by the business lead, the request goes to a team lead or track owner, who assigns it to a developer. If the developer cannot find any existing reports that match the technical requirements, a new report is created. The developer prepares the detailed technical and functional designs for the report. When these are verified against the original design for the report, the report is developed. To deploy the report through the testing and migration phases, a package is created that uses the OraApps Customization/Configuration Deployment workflow.

For More Information

For more information about this request type, see [OraApps Report Request Request Type on page 309](#).

Requesting New Interfaces

In an Oracle E-Business Suite system, interfaces are used to extract data from and provide data to other systems.

The OraApps Interface Request request type is used to request new interfaces for the import and export of data from Oracle. This request type works in conjunction with the OraApps Interfaces Process workflow. Once the request is initiated, it captures business information such as:

- Interface name and description
- Date by which the interface must be completed
- Interface type
- Estimated effort
- Technical analysis
- Technical impact of the new interface

The workflow sends the request through approval, analysis, and assignment steps. The requestor creates the high-level requirements for the interface, which is then signed off. After being prioritized by the business lead, the request goes to a team lead or track owner, who assigns it to a developer. If the developer cannot find any existing interfaces that match the technical requirements, a new interface is created. The developer prepares detailed technical and functional designs for the interface. When these are verified against the original design for the interface, the interface is developed. When

development is completed, a package is created (using OraApps Customization/Configuration Deployment workflow) to deploy the new code through the Oracle E-Business Suite instances for testing and validation.

For More Information

For more information about this request type, see [OraApps Interface Request Request Type](#) on page 306.

Requesting Conversions

Organizations implementing Oracle often use other legacy applications and third-party software programs. Similarly, additional business functions may be moved to Oracle after initial implementation. These other applications and software programs contain data that needs to be brought into Oracle. However, the data must go through a one-time conversion process so that it can be imported and used by Oracle. Whether third-party conversion tools or internally developed programs are used, development effort is required to accomplish the data conversions. In the Extension, these activities can be managed using a conversion request for each set of entities to be unpacked into Oracle.

The process of requesting new conversions requires the OraApps Conversion Request request type, which works in conjunction with the OraApps Conversion Process workflow. Once the request is initiated, it captures business information such as:

- Conversion name and description
- The date by which the conversion must be completed
- Estimated effort in hours
- Technical analysis
- Technical effort required to complete the conversion

After being prioritized by the business lead, the request goes to a team lead or track owner, who assigns it to a developer. If the developer cannot find any similar data that is being converted that matches the technical requirements, a new conversion is created. The developer prepares the detailed technical and functional designs for the conversion. When these are verified against the original design for the conversion, the conversion programs are developed. A package is created to deploy the conversion programs to the appropriate Oracle E-Business Suite instances for testing.

For More Information

For more information about this request type, see [OraApps Conversion Request Request Type](#) on page 294.

Requesting Enhancements

Although Oracle Applications include extensive functionality across a broad spectrum of business areas, it is possible that the standard functionality may not fully meet your organization's needs. For example, corporate policy may require that extensive additional data be captured at certain points in the manufacturing cycle, requiring an additional form for data capture. Or you may need specialized logic incorporated in your purchase order approval process. When situations such as these arise, enhancements need to be developed and implemented in your Oracle E-Business Suite instances.

The following sections discuss the following request types that are provided in the Extension for requesting enhancements:

- OraApps Enhancement request type
- OraApps Design & Development request type

OraApps Enhancement Request Type

The OraApps Enhancement Request request type provides a means of requesting new functionality or modifying existing functionality in an Oracle system. The request could be the result of developing new products or technology, market changes, or complying with legal changes. In the course of a typical Oracle E-Business Suite upgrade, significant changes are made to existing customizations.

The technical requirements behind the change need to be reviewed, approved, and prioritized so that resources are not wasted on activities that do not have significant payback for the organization. The OraApps Enhancement Request request type gathers the following information to help you make these kinds of decisions:

- Enhancement name
- Business area affected by the change
- Application requesting the enhancement
- Date by which the enhancement is needed

- Detailed description of the enhancement
- Analysis information needed for review and approval

The request becomes the central repository for information about the change process involved, including all notes and attachments. Everyone involved with the change process has access to the request so they can view requirements and justifications.

After all approvals have been gathered and the resulting change matches the initial technical requirements, the change is deployed to the appropriate instances.

For More Information

For more information about this request type, see [OraApps Enhancement Request Request Type](#) on page 300.

OraApps Design & Development Request Type

The OraApps Design & Development request type provides a simplified development process for cases where a more extensive process does not make sense, for example, a simple customization that requires porting only. The OraApps Design & Development request type gathers the following information for a simplified development process:

- Summary description
- Specific business areas affected
- Actual business benefit that results from the request
- Analysis of the request from the designated personnel

The request becomes the central repository for information about the change process involved, including all notes and attachments. Everyone involved with the change process has access to the request so they can view requirements and justifications.

The OraApps Design & Development request type works in conjunction with the OraApps Design & Development workflow. Once the change process has been initiated, the workflow routes it through the following steps:

- The request goes through functional analysis, review, and signoff.
- The request goes through technical design, review, and signoff.
- The approved request goes into development.

- A package is created and released into deployment.
- Configurations and customizations are deployed through the standard deployment cycle.

For More Information

For more information about this request type, see [OraApps Design & Development Request Type on page 297](#).

Requesting Setup Changes

An Oracle E-Business Suite implementation can involve many setup changes, depending on the module being implemented. These include application setups, such as setups for sets of books, payment terms, payable options, and accounting periods. These setup changes vary depending on the environment and operating units. Any setup changes approved by business leads need to be made in Oracle before they can be used by application users.

However, an organization may not have the process in place to define and execute these setup changes. Any setup change process that currently exists may not be able to capture all the necessary information to successfully perform the setup change.

The OraApps Setup Change Request request type lets users capture important setup information. The OraApps Setup Change Request request type works in conjunction with the OraApps Setup Change Process workflow. This workflow puts the setup change request through a cycle of analysis, approval, and assignment. Once the setup change process has been initiated, the request captures business information such as:

- Setup name and description
- Application
- Requestor name
- Operating unit
- Environment name

The workflow contains the following steps:

- The functional lead analyzes the change and determines the estimated effort and dependencies to complete the change, if any. The technical lead also determines the estimated completion date for the setup change.

- The technical lead approves the setup change for the environment and assigns a resource.
- The assigned developer determines if the setup change is a manual setup or a Mercury-migrated setup.

A package can be created for migrations handled by Object Migrator or GL Migrator. Other migrations are handled manually.

- The assigned developer sets up the environment.
- For Mercury-migrated setups, the developer creates the package.
- The developer documents the setup changes.
- The request closes successfully.
- If the technical lead does not approve the request, the requestor provides more information and resubmits it.

For More Information

For reference about this request type, see [OraApps Setup Change Request Request Type](#) on page 312.

Deploying Oracle Changes

Over time, and especially when multiple development efforts are underway, it can be challenging to make sure you have visibility to and control over the changes made to a given Oracle E-Business Suite instance. The problem grows larger if you are supporting multiple testing instances. But it is critical to your business to maintain visibility to and integrity of your production mission-critical Oracle E-Business Suite instances.

Once development of new or modified functionality is complete, the changes need to be deployed to the appropriate testing instances, validated, and then deployed to production. Often multiple changes must occur together to implement the new functionality. The Extension includes the OraApps Customization/Configuration Deployment workflow, providing proven practice expertise and standards to help you control your deployment processes. This workflow emphasizes the reviews required at the end of the deployment cycle (for example, a performance review) to make sure that the modification is fully tested and complies with your organization's standards.

A Mercury Change Management package is created that specifies the customizations and configurations ready for deployment. The package then follows the deployment process steps below:

- Modifications are applied to the DEV environment to allow unit testing. AOL or GL objects to bypass this step, since these online configurations would be entered manually in the DEV environment.
- The package moves through the deployment stages as a coordinated entity rather than an individual package. This avoids partial deployment of the overall change.
- Unit testing is performed by the assigned user. A successful test moves the package forward.
- The package goes through a technical review where actual code and scripts are examined. If the package passes the review, it moves to the next step in the workflow.
- The package is deployed from the DEV instance to the TEST instance. If the deployment is successful, the customization or configuration goes through QA testing. If QA testing is successful, the package moves to the next step.
- If the object type for the package is database-related, it is reviewed by a DBA. If it is not, the package skips the DBA review and goes to the Performance Review step.
- The performance architecture team reviews the package before it goes to the production instance. If the package review is successful, it moves forward through the workflow.
- The package is deployed from the TEST instance to the PROD instance.
- When all the package lines have been successfully deployed to the PROD instance, the package is closed. If work needs to be done as part of the Rework Package step, it is performed, and the package starts again at the beginning of the deployment cycle.

The following sections discuss the following deployment subjects:

- Overview of AOL and GL Migration
- AOL or GL Migration execution
- AOL or GL Concurrent Request Log

Overview of AOL and GL Migration

This section describes how to execute Extension-enabled Object Migrator and GL Migrator entities from within Mercury Change Management. The Migrate Concurrent Program (AOL:Conc Prog object type) program is used as an example. The program fields are shown in [Figure 7-1](#).

The screenshot shows a dialog box titled "Add Line" with the following fields and options:

- Object Type Information:**
 - Object Type: AOL:Conc Prog
 - Sequence: 1
 - Application Code: None
- Parameters / User Data:**
 - Source Application: [Text Field]
 - Selection Type: Specific Object
 - Conc Program: [Text Field]
 - New User Program Name: [Text Field]
 - Conc Program From: [Text Field]
 - Conc Program To: [Text Field]
 - Version Label: [Text Field]
 - Version Desc: [Text Field]
 - Overwrite if Exists: Yes No
 - Partials Allowed: Yes No
- Buttons:** Clear, OK, Add, Cancel
- Status Bar:** 'AOL:Conc Prog' parameters loaded.

Figure 7-1. Migrate Concurrent Program (AOL:Conc Prog object type) fields

The Extension includes an object type for each specific migrator in Object Migrator and GL Migrator. The fields are similar to the fields used when running Object Migrator or GL Migrator directly from Oracle. Both the application name and the object name are validated against the initial environment in the workflow. The Version Label and Version Desc (description) fields are used only if the workflow is configured to use the Object Archive. You do not enter the source and destination database for each object. As the package line travels through the workflow, Mercury Workflow Engine determines the source and destination database at each step. [Figure 7-2](#) shows a package using an AOL migration workflow.

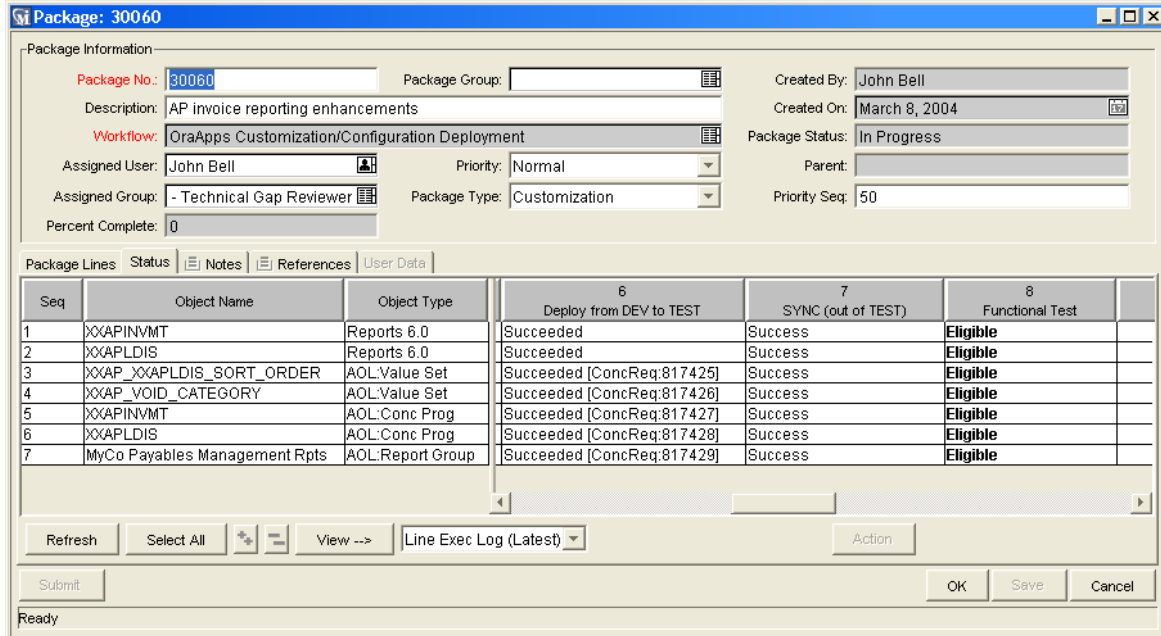


Figure 7-2. Package including AOL objects

Some fields that are visible when Object Migrator runs in Oracle may be hidden by default in Mercury IT Governance Center. This reflects the most common usage in an automated deployment environment. If your organization requires it, these fields can be displayed.

The Extension includes special handling of Object Migrator and GL Migrator migrations. When an action step in a workflow is executed, Mercury Change Management submits a concurrent request in Oracle to run Object Migrator or GL Migrator. It prints the request ID for the concurrent request and displays the status of the concurrent request once it completes. The Oracle log and output files are attached to the Mercury execution log.

AOL or GL Migration Execution

Every time you execute a migration step for a package line containing an AOL or GL entity, Mercury Change Management uses the associated workflow to determine the source and destination databases. It then maps the values entered on the package line to the fields from the Concurrent Program definition of the specific Object Migrator or GL Migrator installation. After the mapping is complete, Mercury Change Management submits a concurrent request to Oracle to run the Object Migrator or GL Migrator installation.

If a package includes more than one AOL or GL object, Mercury Change Management automatically launches the AOL or GL migrations in order (as set in the object type). For example, all the Value Set migrations would have to complete before the Concurrent Program migration requests would start. If there are any migration failures, Mercury Change Management Server fails all dependent concurrent requests and updates the Change Management package appropriately.

Mercury Change Management monitors the status of the concurrent request. When the request is complete, Change Management updates the package line with the appropriate status and attaches the concurrent request output and log files to the Mercury Execution Log.



Note

If you are migrating multiple AOL or GL object package lines, Mercury Change Management automatically runs the concurrent requests serially, based on object dependencies.



Note

If you use environment groups in your workflows, they must be configured for serial execution in order for the AOL or GL object dependencies to be considered.

AOL or GL Concurrent Request Log

You can view the report output of the Object Migrator or GL Migrator concurrent request directly from Mercury Change Management. Change Management launches a Web browser session to display the report.

Using the Extension, Mercury Change Management is able to track and report on concurrent requests triggered by Mercury products. Users can view the AOL or GL Concurrent Request Log for package activities through the Package Status window.

To view a Concurrent Request Log for a particular package line:

1. From the shortcut bar, click **Change Mgmt** and click the **Packages** icon.
2. Query and highlight the target package.
3. Click **Open**.

The Package: ##### window for the target package number opens.

4. Click the **Status** tab.
5. Select the package line for which you want to view the Concurrent Request Log.
6. From the drop-down list at the bottom of the Package: ##### window, select **Line Exec Log (Latest)**.
7. Click **View**.

The Mercury Change Management execution log opens, as shown in the example in *Figure 7-3*. At the bottom of this log is the Mercury IT Governance Concurrent Request Output section. This section contains embedded links to the concurrent request logs and output files.

The screenshot displays the Mercury IT Governance interface for viewing a concurrent request log. At the top, it shows the title 'Mercury IT Governance ©' and a breadcrumb path: 'Execution Log For Batch 90839 - Line 3'. There are three links: 'Package Execution Log History', 'Line Execution Log History', and 'Batch Execution Log'. Below this is a table with the following data:

Object	ACL:Value Set- XXAP_XXAPLDIS_SORT_ORDER
Package No.	30060
Workflow	OraApps Customization/Configuration Deployment
Workflow Step	6 - Deploy from DEV to TEST
Started	March 8, 2004 12:24:49 PM PST

Below the table is a button labeled 'Show Debugging Details'. The main content area shows the following text:

```
KSC OM MIGRATE
Source Command: CLMRMVS1
Successfully submitted: CLMRMVS1
Concurrent Request ID: 817425
```

At the bottom, there is a section titled 'Mercury IT Governance © Concurrent Request Output' with the following details:

Completion Text	Normal completion
Parent Conc. Request ID	

Logfile: [OMSVR - d:\Ora_R11i\appl\oaomcom\admin\log\817425.req](#)
 Outfile: [OMSVR - d:\Ora_R11i\appl\oaomcom\admin\out\o817425.out](#)

Figure 7-3. Mercury Change Management execution log sample data

Chapter 8 Managing Instances

In This Chapter:

- *Overview of Instances*
 - *Cloning Requirements*
 - *Cloning Instances*
 - *Cleaning Up after Cloning*
 - *Configuring Instance Management*
 - *Configuring Environments*
 - *OraApps Cloning Request Request Type*
 - *OraApps Cloning-Related Workflows*
 - *OraApps 11i Cloning Object Type*
 - *ksc_oa_copy_clone_scripts Special Command*
-

Overview of Instances

An Oracle system consists of one or more application file systems, an application database, and associated Oracle home directories and other installed entities. In addition, there are multiple supporting services required to make the system functional. These systems must be maintained and periodically cloned without risking the stability of the system.

Cloning is complex in Oracle E-Business Suite Release 11i, primarily because of the many processes and components involved. Simply replacing all the components of an existing instance does not provide a duplicate functioning system. Several configuration files must also be modified for the clone to work. In addition, the Oracle Rapid Install installation process makes use of Oracle Universal Installer, which writes information about the installation into a binary file. Copying the binary file to a different instance invalidates its contents. As a result, patches may not be applied correctly to those components.

Oracle publishes guidelines that identify the process by which cloning an instance should be performed. The process consists of a series of manual steps that include running cloning scripts provided by Oracle, as well as deleting and copying the database files and other application files.

Typically, an instance might need to be cloned for the following reasons:

- To effectively test patches against a production system without risk, a duplicate of the production system needs to be created and the patch applied to that copy. After it is clear that the patch does not cause any problems, it can then be applied to the actual production system.
- Periodically refreshing a test instance from an active production instance can provide more realistic results for test cases.
- An existing instance may need to be moved to a different machine.
- Multiple training instances may need to be created. These training instances can then be refreshed after a series of classes.

A given development instance may have become unstable due to multiple teams working on different customizations. The development instances must be deleted and cloned with a copy from the Oracle 11i Reference instance. A typical set of activities to complete this process would include:

- Deleting and backing up the development instance.
- Extracting the configuration files from the development instance.

- Copying the application code from the Oracle 11i reference instance to the development instance.
- Copying the database files.
- Restoring the saved configurations after the copy operation.

Due to the complexity of the process used for cloning, there are several steps during which user error can cause major problems. These include steps in which database and application files need to be deleted and overwritten. Manually executing the entire process increases the possibility that some crucial item might be overlooked.

The Extension provides a model workflow for cloning Oracle E-Business Suite Release 11i instances. This workflow enforces a process patterned after Oracle's recommended cloning process.

Automating the process using Mercury IT Governance Center products reduces risks and provides a means to replicate the cloning process. Mercury IT Governance Center also provides a history of the files and databases affected. Having an audit trail of the executions, decisions, justifications, and people responsible improves current and future cloning implementations by providing accurate and complete information, as well as forcing a standard process to which everyone in the organization adheres.

Note

These product features are intended for Oracle E-Business Suite Release 11i only, and only for instances not using Autoconfig.

Note

The Extension lets you automate the cloning process across multiple instances. After cloning is complete, the Extension reports all the customizations and patches that have been lost during the cloning process and need to be applied again. The Extension can then automatically apply the customizations and patches to the newly cloned instance.

Warning

The tools provided are templates. They must be reviewed and configured for your environment before use. Mercury recommends that you utilize Mercury Services to facilitate a successful implementation.

Cloning Requirements

You must adhere to the following considerations for the cloning functionality in the Extension to work properly:

- The source and target systems must have the same type of Oracle database (for example, VISION, fresh install), base language, default territory, APPL_TOP character set, server and node configurations, and platform.
- Perl must be installed on the target system to run the cloning scripts provided by Oracle.
- Adequate disk space must exist on the target system directories.
- The cloning patch from Oracle must be applied to the target system.
- The `config.txt` file should exist for the target system and should not be located under `$APPL_TOP`, `$OA_HTML`, `$JAVA_TOP`, or the database files directory, because these directories are purged during the cloning process.
- If a live instance is to be used as the source system, its application and database services must be shut down prior to initiating the cloning process and kept down until all the file copies are completed.
- If a cold backup is to be used as the source, the files or archives of files for `$APPL_TOP`, `$OA_HTML`, `$JAVA_TOP`, and database files should reside in different directories.
- A trace file to be used as a template for recreating control files must exist in the directory where the source database files are stored. This file can be generated using the `kacc_ora_ctrlfile_template.sql` script delivered with the Extension. This script should be run against the source database. A trace file is created in the `$UDUMP` directory. Rename the trace file to `kacc_ora_ctrlfile.trc` and place it in the source database files directory.

The considerations listed above are based on the process outlined in the *Cloning Oracle Applications Release 11i* white paper published by Oracle.

Cloning Instances

The OraApps Cloning Request request type and the OraApps Cloning Process workflow work together to streamline the process of initiating and tracking a request to have an Oracle Applications instance cloned. The request type provides a means for entering specific information needed for the cloning. The definitions include the following information:

- Source environment
- Destination environment
- Whether or not to clone the database portion of the system
- Date by which the cloning process needs to be completed

The requestor is also prompted for business information for the benefit of the business or technical analyst. This information includes:

- Business area affected
- Perceived business benefit of cloning

Business and technical analysts can record in the request type the results of their analysis and estimate the technical impact of the request. Requirements and justification details are attached to the request so it serves as a central repository for information.

The OraApps 11i Cloning object type and the OraApps 11i Cloning workflow clone an Oracle instance by automating the execution of major steps in the process. Having all the environment information correctly set up through Mercury IT Governance Center causes only the appropriate directories to be cloned.

Information pertinent to the environments to be cloned is captured by the object type. This information is to be used during the execution of commands that clone the instance. Approval and analysis steps throughout the workflow cause the proper analysis and decisions to be made by the appropriate personnel before any execution is performed.



Note

Cloning your Oracle instance is a complex process, and misconfigurations can easily result in unexpected consequences. This object type and its associated workflow provide a template for cloning your instances, but they require tailoring for your environment. Mercury recommends that you utilize Mercury IT Governance Services to facilitate a successful implementation.

Cloning an instance through Mercury IT Governance Center allows the entire process to be monitored from initiation to completion. Information can be shared through references that can be attached to either the request or the package.

When a clone is needed, a request is created using the OraApps Cloning Request request type and the OraApps Cloning Process workflow. The required fields are entered by the requestor to provide the necessary information for the proper groups to be able to decide whether or not the request should be approved. Once the request is submitted, the business analysts, team leads, and DBAs are also prompted to enter appropriate fields to provide information for the next group of decision makers. The fields may or may not be required, depending on the group eligible for that particular step in the workflow.

Once approvals from the various groups are received, a package is created to execute the clone. The package should be defined to use the OraApps 11i Cloning workflow and the OraApps 11i Cloning object type. Once the package line is submitted, the user is requested to verify whether or not the destination system exists and to install it if it does not.

The requirements for cloning are verified, and if any requirements are not met, the user is prompted to make required changes. Once all the requirements have been met, the user is advised to create a backup of the destination instance. The user can decide whether or not a backup of the destination is needed.

The cloning scripts are then run to shut down the application services and to save the destination configuration information to be used later in the process. If the database is to be cloned along with the application code, steps for deleting and copying application files and database files are executed. There are appropriate analysis and rerun steps to address errors that may occur during the course of these tasks.

Once the preceding steps have been successfully executed, the cloning scripts are run to start up the application services and restore the configuration files with the appropriate information specific to the target instance. The user is then asked to perform any final tasks that are required and to test the destination instance before closing the package line.

Once the package is closed, the request also closes with the appropriate status.

For More Information

For more information about configuring this process, see [OraApps Cloning Request Request Type on page 193](#) and [OraApps Cloning Process Workflow on page 194](#).

For more information about doing clean-up tasks, see [Cleaning Up after Cloning on page 187](#).

For more information about the entities mentioned in this section, see:

- [OraApps Cloning Request Request Type on page 291](#)
- [OraApps Cloning Process Workflow on page 329](#)
- [OraApps 11i Cloning Object Type on page 276](#)
- [OraApps 11i Cloning Workflow on page 373](#)

Cleaning Up after Cloning

The OraApps 11i Cloning workflow contains a number of steps that take place after the cloning of an instance has been completed:

- The user is prompted to do additional tasks that may be required.

These tasks could include modifying the Web configuration file, updating the self-service fields, and relinking the Oracle executables.

For more information about these tasks, see the *Cloning Oracle Applications Release 11i* white paper published by Oracle.

- A test of the resulting destination instance is requested.

If there are no problems, the package line is closed as successful.

After cloning an instance and responding to the prompts listed above, an Environment Refresh of that instance needs to be performed to synchronize the Mercury data and determine what entities need to be reapplied. After using the Refresh function, Mercury Change Management has the ability to update its audit history and its in-process packages. This update is performed using Mercury IT Governance Center's Environment Refresh Workbench functionality.

An Environment Refresh:

- Identifies open package lines that have migrated through the environment and are now inaccurate due to the software changes.

For example, if you refresh the DEV and QA environments with the current PROD environment, the package lines become inaccurate.

- Updates the list of affected package lines, as necessary, including adding or deleting lines from the list.
- Updates the Mercury Change Management internal object inventory tables to reflect the physical refresh.

This operation copies the audit history from the source environment to the audit history of the refreshed environment. Since they are now physical matches of each other, their audit history should be the same.

- Updates the open package lines in the refresh list and resets them so they are eligible for migration into the refreshed environment again.

For More Information

For more information about refreshing environments, see the *System Administration Guide and Reference*.

Configuring Instance Management

The following sections discuss the following subjects:

- Configuring environments
- OraApps Cloning Request request type
- OraApps cloning-related workflows
- OraApps 11i Cloning object type
- Special command `ksc_oa_copy_clone_scripts`

Configuring Environments

To configure environments, you need to enter server, client, and database environment information in the Workbench Environment window's **Host** tab and its **Extension Data** tab, **Oracle Applications** subtab.



The tools provided are templates. They must be reviewed and configured for your environment before use. Mercury recommends that you utilize Mercury Services to facilitate a successful implementation.

Sample data in the **Host** tab of the Environment window is shown in *Figure 8-1*.

The screenshot shows the 'Environment : OA - OA9U 11i local' window with the 'Host' tab selected. The window contains three main sections: Server, Client, and Database, each with its own set of configuration fields and an 'Enable' checkbox.

Section	Field	Value
Server	Name	server.company.com
	Username	applmgr
	NT Domain	
	Type	Sun Solaris
	Password	*****
	Base Path	/export/home/applmgr
Client	Name	server.company.com
	Username	applmgr
	NT Domain	
	Type	Sun Solaris
Database	Server Type	Oracle Server
	Host Name	server.company.com
	Username	apps
	Oracle SID	OA9U
	DB Link	
	Connect String	OA9U
	Port Number	1528

Buttons at the bottom: Check..., OK, Save, Cancel. Status: Ready.

Figure 8-1. Environment window, Host tab, sample data

Sample data in the **Oracle Applications** subtab on the **Extension Data** tab of the Environment window is shown in *Figure 8-2*.

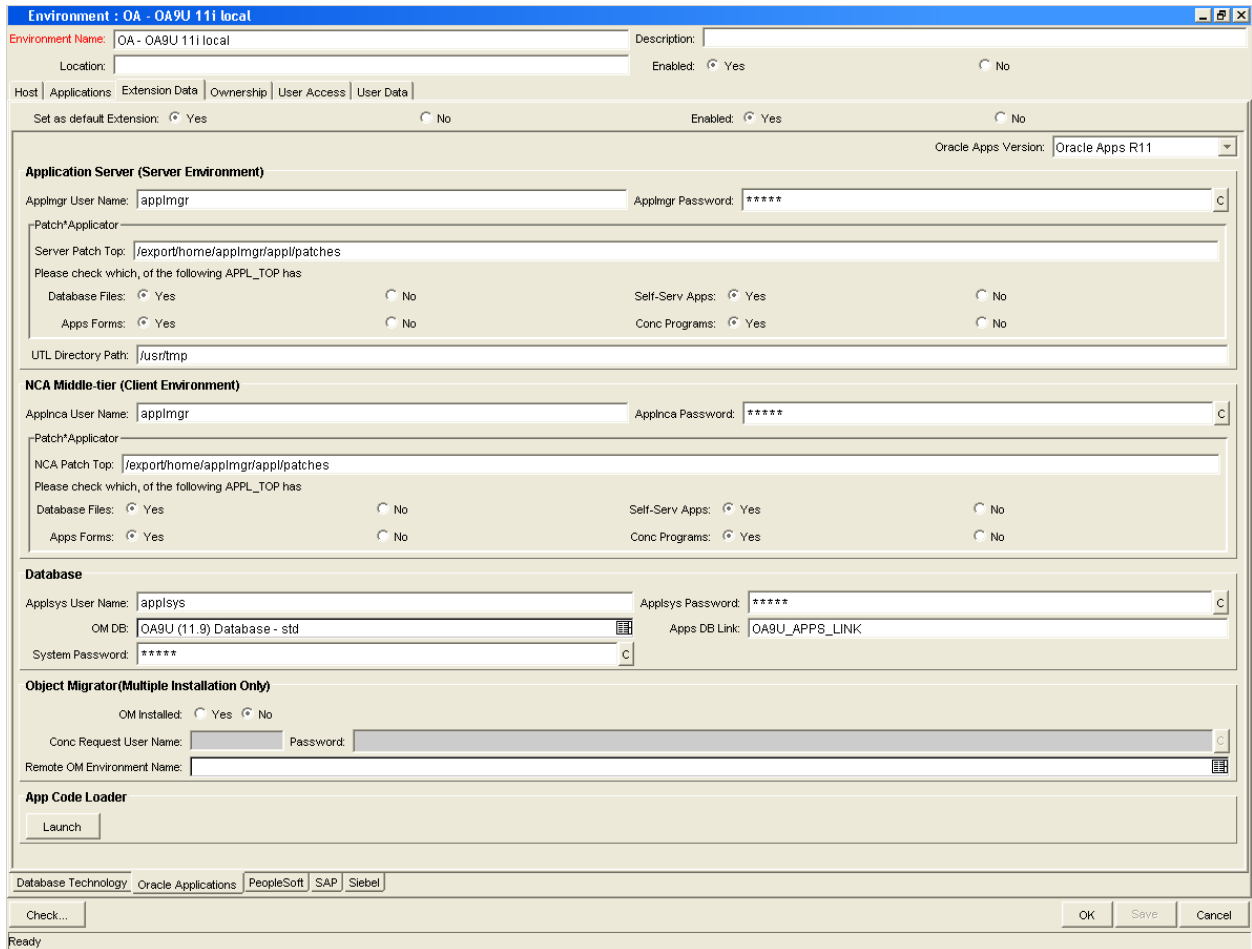


Figure 8-2. Environment window, Oracle Applications subtab, sample data

Required Field for the Source Environment

In the **Host** tab, the field in *Table 8-1* is required for the source environment.

Table 8-1. Host tab required field for source environment

Field	Value
Database section	
Oracle SID	SID for database.

Required Fields for the Destination Environment

Table 8-2 shows the required fields for the destination environment on the **Host** tab and on the **Extension Data** tab, **Oracle Applications** subtab.

Table 8-2. Host tab and Oracle Applications subtab required fields for destination environment

Field Name	Value
Host tab, Server section	
Name	Host name of the server to be used as the target.
Type	Type of server.
Username	User name.
Password	User password.
Base Path	Base path.
Connection Protocol	Connection protocol to use.
Transfer Protocol	Transfer protocol to use.
Host tab, Database section	
Server Type	Oracle Server
Host Name	Host name of the database server to be used as the target.
Connect String	Connect string for the database.
Username	APPS Username.
Password	APPS password.
Oracle SID	SID for the database.
Extension Data tab, Oracle Applications subtab, Application Server (Server Environment) section	
Applmgr User Name	Applmgr user name.
Applmgr Password	Applmgr password.
Extension Data tab, Oracle Applications subtab, Database section	
System Password	System password
Applications tab	
Application Code	<i>ORACLE</i>
Application Name	<i>Oracle_User</i>
Server Username	Oracle Username.

Table 8-2. Host tab and Oracle Applications subtab required fields for destination environment [continued]

Field Name	Value
Server Password	Oracle password.
Server Base Path	Base path for the Oracle User.

OraApps Cloning Request Request Type

The OraApps Cloning Request request type lets you streamline the process of initiating and tracking a request to have an Oracle E-Business Suite instance cloned. The request type captures the business details that justify the cloning process and works in conjunction with the Oracle Cloning Process workflow to go through approval and analysis steps. If cloning the instance is approved, a Mercury Change Management package is created that uses the OraApps 11i Cloning object type and the OraApps 11i Cloning workflow.

To configure the OraApps Cloning Request request type:

1. Use the Default Request Header Type.
2. Restrict the request type to allow only the OraApps Cloning Process workflow.
3. Set the OraApps Cloning Process workflow as the default workflow for the request type.
4. If you need tighter security for specific fields on the request type, configure field security setups for these fields. For example, you might want to configure the request type to allow only the DBA team to update the Analysis field.
5. If you want to limit who can create cloning requests, add user access restrictions.



Note

The request type is defined to use participant security, so only users directly involved with the request can access it.

6. When you are finished configuring, enable the request type and the workflows.

For More Information

For more information about the request type, see [OraApps Cloning Request Request Type](#) on page 291.

OraApps Cloning-Related Workflows

This section contains information about the following workflows:

- OraApps Cloning Process workflow
- OraApps 11i Cloning workflow

OraApps Cloning Process Workflow

The OraApps Cloning Process workflow lets you streamline the process of cloning an Oracle E-Business Suite instance. This workflow works in conjunction with the OraApps Cloning Request request type and consists of approval and analysis steps for the appropriate resources. Once approved, the workflow creates a package in Mercury Change Management that completes the cloning process, using the OraApps 11i Cloning workflow, which is described in [OraApps 11i Cloning Workflow on page 373](#).

To configure this OraApps Cloning Process workflow:

1. Assign a security group to each workflow step so the appropriate user can act on them.

The requestor (for example, developer, system administrator, or QA tester), business group managers, IT lead, and DBA team need authorizations for creating a cloning request.

2. If you want, add notifications to the workflow to help streamline the process.
3. Restrict the workflow so it can create packages using only the OraApps 11i Cloning workflow.
4. Enable the workflow.

For More Information

For more information about the OraApps Cloning Process workflow, see [OraApps Cloning Process Workflow on page 329](#).

OraApps 11i Cloning Workflow

The OraApps 11i Cloning workflow works in conjunction with the OraApps 11i Cloning object type. When the OraApps Cloning Process workflow reaches the step where it must create the package and wait, the user creates a package that uses the OraApps 11i Cloning workflow and the OraApps 11i Cloning object type in its package lines.

The OraApps 11i Cloning workflow follows this process:

- The workflow verifies the cloning requirements and begins copying Mercury IT Governance Center cloning scripts.
- The cloning scripts run in pre-clone mode to preserve the instance's configuration information.
- The database is shut down, any database files that need to be deleted are deleted, and the files are copied from source to destination.
- Control files for the destination database are created.
- The database is started.
- Any concurrent requests copied from the source are cleaned up.
- Various code files are copied from the source backup directory to the destination.
- The cloning scripts are executed in post-cloning mode.

If invoked from a request, the request would get the final status and continue.

To configure the OraApps 11i Cloning workflow:

1. Assign the proper security groups to each step of the workflow.
2. Limit the workflow to use only the OraApps 11i Cloning object type.
3. Review the workflow and add steps unique to your business.

This may also entail changes to the OraApps 11i Cloning object type.

4. Enable the workflow.

For More Information

For more information about the OraApps 11i Cloning workflow, see [OraApps 11i Cloning Workflow on page 373](#).

OraApps 11i Cloning Object Type

The OraApps 11i Cloning object type provides a means to clone an instance of Oracle E-Business Suite by automating the execution of major steps in the process. Information pertinent to the environments to be refreshed is captured in the object type. This information is to be used during the execution of commands that clone the instance. Only the databases and directories that are supposed to be cloned are affected. This object type works in conjunction with the OraApps Cloning Process workflow, described in [OraApps Cloning Process Workflow](#) on page 194.

If you alter any of the default workflow step names in the OraApps 11i Cloning workflow, apply the proper modifications to the OraApps 11i Cloning object type command conditions, because the various commands execute based on the workflow step name.

For More Information

For more information, see [OraApps 11i Cloning Object Type](#) on page 276.

ksc_oa_copy_clone_scripts Special Command

The `ksc_oa_copy_clone_scripts` command copies custom Mercury IT Governance Center versions of the Oracle cloning scripts to the destination server. It is used by the OraApps 11i Cloning object type.

Chapter 9 Managing Patches

In This Chapter:

- *Overview of Patches*
 - *Analyzing Patch Information*
 - *OraApps Patch Detail Report*
 - *OraApps Patch Analysis Report*
 - *OraApps Patch Data Capture Subworkflow*
 - *Applying Patches*
 - *Overview of Applying Patches*
 - *Automating ADPATCH and ADADMIN*
 - *Merging Patches*
 - *Reports About Patches*
 - *Oracle Applications Implementation and Patching Scenarios*
 - *Configuring Patch Management*
 - *System Requirements*
 - *Configuring the OA - Oracle Patch Admin Security Group*
 - *Configuring Environments*
 - *Configuring Object Types*
 - *Configuring Workflows*
 - *Configuring Report Types*
 - *Special Commands*
 - *Maintaining a Mercury IT Governance System*
 - *Maintaining a Mercury IT Governance Server*
 - *Maintaining a Mercury IT Governance Database*
-

Overview of Patches

A significant number of patches are distributed by Oracle for each Oracle release, and the number per release continues to grow. For system stability, these patches need to be applied to and verified in each Oracle E-Business Suite instance in use. Research is often required to determine prerequisite patching requirements or the impact of a given patch on custom-developed extensions and system functionality. Patches may be requested by different parts of the organization, and multiple patches may be in progress at a given time. Analyzing, applying, and tracking of patches across multiple Oracle E-Business Suite instances (and tiers) can pose a major management challenge. In addition, patches can require system downtime, so timely processing is important.

The Extension provides you with the tools you need for effective patch management. With minimal configuration, the workflows, object types, request types, and report types included in the Extension function together to manage common patching tasks. The Extension allows you to take control of your patch management functions by providing critical functionality such as:

- Automated management of ADPATCH and ADADMIN interactive sessions using a rule-based engine and notification-on-fail features.
- Ability to analyze patch content details before or after patching, without reliance on scarce DBA resources.
- Ability to merge patches for more efficient application.
- Ability to patch single or multi-tier instances in serial or parallel.
- Automated execution of post-patch activities, such as required ADADMIN functions, in conjunction with patch application.
- Identification of bugs reported to Oracle and pending resolution.
- Ability to track the patches received and the instances to which they are applied using processes that have been verified by Oracle customers.

Analyzing Patch Information

Before applying a patch, it is common to analyze the patch contents to assess impact and potential risks. This analysis helps to determine when to apply the patch, where to apply it, whether functional analysis is required (for example, if the patch impacts functionality relied on by custom Extensions), what tests should be run to verify it, and which functional areas need to approve the patch application.

The file contents of a patch are listed in the patch's C-driver or U-driver file. For patches that contain a large number of changes or large number of subpatches, deciphering the contents of the patch can take a considerable amount of time and manual effort, exacerbated by the fact that patch files may be controlled by a DBA team and not available directly to the person doing the analysis.

When bugs are found in a given Oracle instance, it may be necessary to identify whether specific existing patches have already been applied to the instance. The patches in question may be subpatches delivered in a master patch set. Quickly assessing whether a particular file or subpatch has been applied can save analysis time and help speed resolution of an Oracle TAR. Similarly, if a patch is a prerequisite for a bug fix, identifying that the patch has already been applied can save the effort of reapplying the patch.

The patch analysis feature of the Extension provides functionality that allows you to capture and analyze Release 11 or 11i patch contents before or after patch application. It provides an automated mechanism to record patch details during the patch application process, allowing this information to be leveraged during patch verification activities. Analyst users without configuration licenses for Mercury Change Management can load patches and run the analysis with a standard Change Management license through the Dashboard.

A flexible mechanism is provided to allow users to report on patch data in a way that is meaningful to them. Patch analysis using the Extension consists of using the following entities, as described in the following sections:

- OraApps Patch Detail Report
- OraApps Patch Analysis Report
- OraApps Patch Data Capture Subworkflow

OraApps Patch Detail Report

The OraApps Patch Detail Report allows users to view patch contents, including files and subpatches, both before and after patches are applied to an environment. The format of the report is similar to the Mercury standard detail reports for entities such as object types. Users can locate patches based on their contents (for example, files, bugs, and environments) and can determine which data is displayed in the report. Users can load patch data prior to reporting or delete it prior to application. The C-driver or U-driver file of patches to be loaded are taken out of the standard patch repository, removing the need for users to physically access the patch archive, thereby reducing the load on DBAs to provide patch information to non-DBA users.

For More Information

For more information about this report, see [OraApps Patch Detail Report on page 403](#).

OraApps Patch Analysis Report

The OraApps Patch Analysis Report allows users to view patch contents in a tabular format, which facilitates integrated data analysis. Users can locate patches based on their contents or application status, and can specify the types of data displayed on the report and the display order. Using this report, users can quickly determine whether a given subpatch or file has been applied, and in which environments.

For More Information

For more information about this report, see [OraApps Patch Analysis Report on page 400](#).

OraApps Patch Data Capture Subworkflow

The OraApps Patch Data Capture Subworkflow automates the capture of patch data during patch application. The subworkflow is invoked during each patch application, so that patch detail data is available when patches are validated for a given environment. The subworkflow enforces rules regarding when a patch is eligible for capture, captures patch details, and associates patch details with the patched environment. The C-driver or U-driver file is taken from the patched OraApps instance and used to determine patch details. For efficiency, once the patch has been successfully loaded for a given package line, the full patch is not analyzed if the patch already exists in the data tables. A notification is issued if the capture of patch data fails for any reason. Logs are created to indicate whether the capture took place and to show the treatment of the bugs found in the patch.

When the Mercury IT Governance Center Environment Refresh functionality is used, patch detail data is automatically kept in sync with environment contents information.

Patch details are captured for both standard and merged patches. In order to meaningfully report on merged patches, they should have been merged using the “merged_name” syntax, otherwise all merged patches are reported as a single patch.

For More Information

For more information about this subworkflow, see [OraApps Patch Deployment Workflow on page 356](#).

For more information about merged patches, see [OraApps R11i Merge Patch Object Type on page 224](#).

For descriptions of the various configuration processes involved in patch analysis, see [Configuring Patch Management on page 212](#).

Applying Patches

The following sections discuss the following subjects:

- Overview of applying patches
- Automating ADPATCH and ADADMIN
- Merging patches
- Reporting on patches

Overview of Applying Patches

The Extension automates Oracle’s ADPATCH, ADADMIN, and ADMRGPCH utilities through the standard Mercury IT Governance Workflow and Execution engines. This allows you to apply and track patches across multiple Oracle instances using Mercury workflows and Mercury object types.



Note

The Extension is designed to apply only Oracle patches, and cannot be used to apply Oracle database or 10SC patches.

Once an Oracle patch has been identified and received from Oracle, it should be taken through a structured process of review and deployment to maintain visibility to the patch and reduce risk of applying a bad patch to a critical environment. Review each patch for impact, pre- and post-patch application activities, and general relevance to your specific environment.

After the review process is complete, apply it to a “vanilla” Oracle instance that does not include any in-house customization. This is a standard practice to isolate the Oracle patch and verify its correctness. If you find that the patch works in this environment and not in the other instances, then the problem might be due to customization or configuration. After deploying the patch successfully to the vanilla instance, apply it to your development, testing, and production environments, in turn, verifying success in each environment.

If the patch fails testing and verification in any of these instances, review it again. The purpose of re-review is to determine the problem with the patch, and, if can be corrected, move the patch through the full deployment cycle to all specified instances. If the patch cannot be fixed, cancel deployment.


The Extension provides the tools you need for effective patch management. With minimal configuration, Mercury workflows, object types, and report

types function together to automate common patching activities and processes as follows:

- Interactive ADPATCH and ADADMIN sessions are automatically initiated and managed using rule-based intelligence.
- Post-patch activities (such as performing ADADMIN functions) are included.
- Interested users are notified of patch failures, availability for verification, and rejection.
- Analysis of patch content is performed.

Note

If you do not have a central repository of patches, each patch must be manually copied to each of the target environments where it will be deployed. You can do this in a controlled manner using a Mercury workflow or by modifying the (R10 or R11) Oracle Patch object type provided by Mercury to reference a source environment instead of a patch stage. The patch repository can be NFS mounted to avoid having duplicate copies of the patch files in each instance.

Warning

There are two different types of entities for applying patches: a Release 10 patch (R10 Oracle Patch) and a Release 11 or 11i patch (R11 Oracle Patch). Attempting to apply a Release 10 patch to a Release 11 environment, or vice versa, will result in a failed migration.

Automating ADPATCH and ADADMIN

To apply patches in an Oracle environment using the ADPATCH and ADADMIN utilities:

1. Using Mercury Change Management, create a package.
2. Select the OraApps Patch Deployment workflow.



Mercury recommends, but does not require, using the OraApps Patch Deployment workflow. Object types related to patching can function with workflows you have created or configured at your site. However, capturing patch detail data requires the OraApps Patch Capture Subworkflow.

3. Add package lines and select the R11 Oracle Patch object type.

The ADPATCH utility is invoked automatically by the object type.

The Oracle patch is copied from a central repository, and then unpacked in the Oracle instance where the patch will be applied. The ADPATCH prompts need to be answered interactively, based on current field values and the setup of the environment being patched.

4. If your patch requires administration tasks, you can make these tasks occur automatically. Add a new package line and select one of the following object types to work in conjunction with the Oracle ADADMIN utility:
 - OraApps ADADMIN Compile APPS Schema object type
 - OraApps ADADMIN Compile Flexfields object type
 - OraApps ADADMIN Generate Jar Files object type
 - OraApps ADADMIN Generate Messages object type

For descriptions and configuration information about these object types, see [Configuring Object Types on page 222](#).

5. Submit the package.

When the workflow executes, the Oracle ADADMIN utility is automatically invoked by the object type.

Limitations

The Extension is designed to run the ADPATCH and ADADMIN utilities, which utilize a standard prompt and response structure. Most of the time this is sufficient to successfully run a complete Oracle patch. Unfortunately, Oracle does not have a guaranteed standard for the manner in which patches are generated and applied. Some patches require additional interaction other than simple questions and answers. The Extension does not support these more complex and unpredictable interactions.

For this reason, Mercury strongly recommends that the first time you apply a patch, you apply it manually in your “sandbox” environment, after reading the `README.html` file provided with the patch. During this process, pay particular attention to any manual actions the patch requires beyond answering questions when prompted. If such manual actions are required for a patch, the Extension cannot be used to apply the patch. However, you can still generate a package line to track the history of the patch, and execution steps can be bypassed while the patch is manually applied.



While manually applying a patch to your sandbox environment, note the names of the driver files required to apply it. These names will be required later when you create a package line in Mercury Change Management for the patch to be automatically applied to other environments.

Execution Logs

For each environment to which you apply a patch, a log file of the ADPATCH output is produced. These log files typically reside local to the Oracle instance to which the patch was applied. This makes it very difficult to audit results across environments. Additionally, there are no records of pre- or post-application steps that may be associated with the patch.

The Extension solves this by combining all ADPATCH interactions into a single color-coded HTML log file that is accessible from a Web browser through the standard package line execution logs. The Mercury Change Management execution log includes pre- and post-application step records, and links to associated information (including the `README.txt` file for that particular patch).

For More Information

For more information about the “OraApps ADADMIN” object types, see the following sections:

- [OraApps ADADMIN Compile APPS Schema Object Type on page 266](#)
- [OraApps ADADMIN Compile Flexfields Object Type on page 269](#)
- [OraApps ADADMIN Generate Jar Files Object Type on page 271](#)
- [OraApps ADADMIN Generate Messages Object Type on page 273](#)

Merging Patches

The OraApps R11i Merge Patch object type automates the Oracle ADMRGPCH utility, allowing users to merge multiple compatible Oracle E-Business Suite Release 11i patches into a single patch. This decreases the number of individual patches to be tracked and applied, and can decrease the total time dedicated to tracking and applying patches.

The OraApps R11i Merge Patch object type takes specified patch archive files from the Patch Stage repository, merges them into a single patch in an Oracle E-Business Suite Release 11i instance using ADMRGPCH, and then returns the merged patch archive to the Patch Stage repository.

The OraApps R11i Merge Patch object type produces a new patch archive with a user-specified name that can be used with Oracle patch-related object types to apply the patch. Details of the patch are captured by the patch analysis functionality in the Extension.

To merge patches in an Oracle environment that uses ADMRGPCH:

1. Place the patch archives to be merged in the Patch Stage area.
2. In Mercury Change Management, create a package.
3. Add a package line to the patch that uses the OraApps R11i Merge Patch object type.
4. Select the workflow defined to merge patches.
5. Submit the package.

This procedure merges the individual patches into a single patch. You can then process the merged patch using the OraApps Patch Deployment workflow.



Warning

ADMRGPCH does not merge patches of different releases, different platforms, or different parallel modes.

ADMRGPCH does not consider any dependencies in the order in which patches should be applied. You need to be extremely careful in merging patches if there are application-order dependencies.



Warning

In Release 11i, Oracle added support for a new type of patch driver, a U-driver. Earlier versions of ADMRGPCH merge these patches but do not merge the U-driver information, which can result in a potentially destructive merged patch. If you need to merge patches that include U-drivers, you must use an instance that is at least at the 11.5.9 or AD.H patch level.



Note

The name you choose for the patch archive is also the name used for the patch itself. The name can contain no special characters except for the underscore (_).

For More Information

For more information about the object type discussed in this section, see [OraApps R11i Merge Patch Object Type on page 263](#).

For more information about the workflow discussed in this section, see [OraApps Patch Deployment Workflow on page 356](#).

Reports About Patches

After applying all the patches to your Oracle environment, you can run the reports (available through the Extension) that give you information about the impact these patches have had on your Oracle system. The following reports, described in the following sections, provide specialized patch information that can help you in your post-patch analysis:

- OraApps Patch Detail Report
- OraApps Patch Analysis Report
- Patch Application Comparison Report
- Patches Applied to an Environment Report
- Pending Patches Report

OraApps Patch Detail Report

The OraApps Patch Detail Report provides details about selected patches. This report is also used to load patch data for unapplied patches. Users have control over the type of information displayed and the patches selected. If fields are selected to load patch data, the report type calls a special command to load the data prior to executing the report. If required, the report calls another special command to remove the patch data after report completion.

For More Information

For more information about this report, see [OraApps Patch Detail Report on page 403](#).

For information about configuring this report, see [OraApps Patch Detail Report on page 229](#).

OraApps Patch Analysis Report

The OraApps Patch Analysis Report is similar to the OraApps Patch Detail Report, except that it does not allow users to load in patch data. Also, rather than providing information in a master/detail format, it consolidates all the information into a single table. This allows you to bring the information into spreadsheet programs such as Microsoft Excel for additional reporting. You can select a variety of filter criteria, and you can configure which information to show in the table and in what order. For example, you might want to see only patch headers and included bug information, or just bug and file information.

For More Information

For more information about this report, see [OraApps Patch Analysis Report on page 400](#).

For information about configuring this report, see [OraApps Patch Analysis Report on page 229](#).

Patch Application Comparison Report

The Patch Application Comparison Report is specialized for Oracle E-Business Suite patches. It can be used to obtain a list of patches migrated to or applied to up to five different environments. The report lists each patch number and the last migration date for the patch into each environment, allowing you to see if a patch was applied out of order or was missed in one of the environments.

For More Information

For more information about this report, see [Patch Application Comparison Report on page 406](#).

Patches Applied to an Environment Report

The Patches Applied to an Environment Report lists Oracle E-Business Suite patches migrated to or applied to a specific environment. You can use this report to see the history of your Mercury Change Management executions for these patches. You can see if an Oracle Application patch has been applied multiple times to the same environment. You can also use this report to obtain a list of all the patches applied in a specific date range or all the patches run after a specific patch was applied.

For More Information

For more information about this report, see [Patches Applied to an Environment Report on page 408](#).

Pending Patches Report

The Pending Patches Report is specialized for Oracle E-Business Suite patches. It lists all package lines for Oracle patches that are waiting at a migration step (that is, the line has been approved for a given environment and now the patch needs to be applied to it).

For More Information

For more information about this report, see [Pending Patches Report on page 410](#).

Oracle Applications Implementation and Patching Scenarios

A number of implementation architectures are supported by Oracle. Each is discussed briefly in this section to provide context for the patch application process in the Extension. The architectures considered here are:

- Release 10.7 Character Mode Only

All Oracle patches are applied to the server using ADPATCH, and can be automated using the Extension. Patches applied should precisely follow the instructions sent with the patch.

- Release 10.7 Smart Client

This scenario is the client-server model for Oracle. It consists of a strong Oracle Server (such as the character-mode server of the Release 10.7 Character Mode Only architecture) and a GUI client installed on Windows platforms. Server-side patches are applied using ADPATCH and can be automated using the Extension. Client-side patches do not use ADPATCH and the Extension does not support their application.

- Release 10.7 Network Computing Architecture

This scenario is a three-tier architecture with a lean client (that is, front-end client) in the form of an Internet browser, a middle tier to handle Internet or network traffic, GUI forms for user interaction, and a powerful server that performs core functions (for example, programs and reports). With this architecture, Oracle ships separate patches for the middle tier and the server. The Extension supports server-side patches on UNIX and Windows, and client-side patches on Sun Solaris.

- Release 11

The Extension supports both server-side and middle-tier standard patches on UNIX and Windows platforms.

Normally all three drivers (C-, D-, and G-) are to be applied at the server. Only *Cbugnumber.drv* and *Gbugnumber.drv* should be applied to the middle tier. Do not apply *Dbugnumber.drv* or any database driver file to the middle tier.

If the server and middle tier share the same code tree, then do not apply any driver files to the middle tier.

- Release 11i

With Release 11i, for the first time on Windows, Oracle implements relinking of executables during patching. To allow this relinking, Oracle requires the installation of additional third-party software tools for C++ compilation and UNIX shell emulation. In order for Mercury IT Governance Center to function correctly for Windows, you must make sure that your Oracle environment variables (for example, APPL_TOP) are visible within a Bourne-style shell. If all settings are not immediately available, you may need to customize the patch-related object type to set the necessary values. This includes PATH settings, both for Oracle directories and third-party tools such as MKS Toolkit. If the host uses multiple UNIX emulators, the MKS Toolkit shell must be used for patch application.

The Extension supports both server-side and middle-tier standard patches on UNIX and Windows platforms.

Using a patchset to Release 11i, Oracle added support for a new type of patch driver—the U- (Unified) driver, which combines the actions of the C- (Copy), D- (Database), and G- (Generate) drivers. The Extension supports the use of U-drivers when applied in accordance with the guidelines in the “*11.5.9 Maintenance Procedures Guide*” document from Oracle. Generally this requires applying the U-driver first to the admin server node, then to any other nodes defined for the instance.

Oracle’s ADPATCH utility determines whether a given driver can be applied to a given physical server node.

Configuring Patch Management

This section discusses configuring the following for Oracle Applications patch management:

- System requirements
- OA - Oracle Patch Admin security group
- Environments
- Object types
- Workflows
- Report types
- Special commands

System Requirements

You should establish a patch repository where patch archives for all patches will reside, regardless of Oracle release. This repository is represented by the Patch Stage environment in Mercury IT Governance Center. A patch repository is used as the source of patch archives for pre-patching detail loads and for patch application activities.

Considerations for creating the patch repository include the following:

- The patch repository should be accessible by a telnet/ssh/bash client and must allow file transfer and directory creation.
- The repository should have adequate available space to extract the C-driver for a given patch. This occurs during pre-patching load of patch detail data.
- The server on which the repository is located must have an unzip utility installed that supports the format of the Oracle patch archives located on the server.

Each Oracle instance that will be accessed for patch-related activities must allow the following:

- A location where patches are unpacked on the Oracle instance must be accessible by a telnet/ssh/bash client and must allow file transfer.
- The patch must be unpacked in the patching area before the ADPATCH utility is called and the patch detail data is captured during patching. By default, the patch-related object type moves the archive from the patch repository and unpacks it in this directory (as specified by the PATCH_TOP value in the environment definition for the instance).
- Operating system sessions must be able to run a configuration file to set an environment context (for example, \$APPL_TOP) for the Oracle instance. By default it is expected that the configuration file has the same name as the database SID for the instance. On Windows, this file may need to be constructed, because the format of the files delivered by Oracle may not be executable from a bash shell.

Configuring the OA - Oracle Patch Admin Security Group

The OA - Oracle Patch Admin security group is used by default in the OraApps Patch Deployment workflow and the OraApps Patch Data Capture Subworkflow. Normally, users in this group would be DBAs responsible for patching and instance administration.

If this group is appropriate to your organization, you should add access grants for it. Usually these include standard Mercury Demand Management and Mercury Change Management functions, plus environments, workflows, environment refreshes, and releases. You should also add view grants for environments, workflows, environment refreshes, and releases.

Configuring Environments

This section discusses configuring the following environments:

- Patch Stage environment
- Patch destination environment
- SERVER_ENV_NAME environment

Patch Stage Environment

The patch management functionality in the Extension assumes that all patches reside in a patch repository represented by the Patch Stage environment, and this environment is distinct from all Oracle environments that require patching. The environment exists solely as a repository from which patches are copied and later applied to target environments, or where C-drivers or U-drivers are extracted from existing patch archives to allow capture of patch details prior to patching. The patch repository is not release specific.

The patch repository is used during patching, loading of patch details prior to patching, and when patches are merged.

To create a Patch Stage environment:

1. In the Workbench Environment window, create an environment called Patch Stage.



Note

It is important that this environment be called Patch Stage, because this is the exact environment name that the object type looks for when copying the patch to the target environment. Exact spelling and case is important. You can change the standard object types if you want to follow a different naming convention.

2. Complete the remaining **Host** tab fields as shown in [Table 9-1](#).
3. Click the **Extension Data** tab and then the **Oracle Applications** subtab at the bottom of the window, and complete the field as shown in [Table 9-1](#).

No other configuration of the Patch Stage environment is required.

An example **Host** tab for the Patch Stage environment is shown in [Figure 9-1](#).

An example **Extension Data** tab for the Patch Stage environment is shown in [Figure 9-2](#).

Table 9-1. Host tab and Oracle Applications subtab required fields for Patch Stage environment

Field Name	Value
Host tab, Server section	
Name	Host name of the server to be used as a server-side patch repository.
Username	Valid Username that has access to the patch repository. This user must have access to an unzip utility that supports the Oracle patch archives located on the server, and must also have access to create directories and files within the repository directory.
Password	Password for the Username.
Base Path	Path to the directory where Oracle Server patches are located.
Host tab, Client section	
Name	Host name of the client (or middle tier) to be used as a middle-tier patch repository.
Username	A valid Username that has access to the patch repository.
Password	Password for the Username.
Base Path	Path to the directory where Oracle middle-tier patches are located.
Extension Data tab, Oracle Applications subtab, Application Server (Server Environment) section	
Server Patch Top	Path to the directory where Oracle patches are located. This is used during patch merge.



Note

The directory entered in the Base Path field for the server and client should be the directory where all the patch archive files are placed. Typically these are compressed files named *filename/number.Z* or *filename/number.zip* from which the patch files are extracted.

Windows users should configure a virtual FTP server directory for the base path, if one does not already exist.

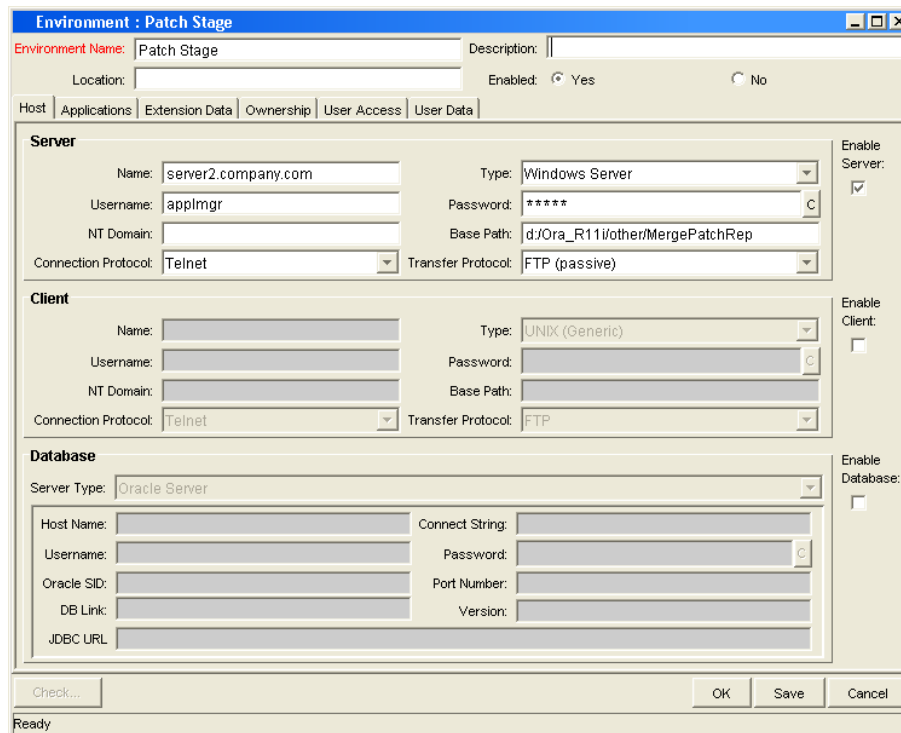


Figure 9-1. Patch Stage Environment window, Host tab, sample data

Figure 9-2. Patch Stage Environment window, Oracle Applications subtab, sample data



Note

If you do not plan to maintain a central repository of patches, you must manually copy each patch to each of the target environments in which you plan to have Mercury Change Management apply it. You can do this in a controlled manner using a Mercury IT Governance workflow, or by modifying the patch object type to reference a source environment instead of the patch stage. The patch repository can also be NFS mounted to avoid having duplicate copies of the patch files in each instance.

Patch Destination Environment

Before applying any patches using the Extension, you need to configure each destination environment. To configure an environment for patch application, enter fields in the Environment window's **Host** tab and **Extension Data** tab, **Oracle Applications** subtab.

To configure a destination environment:

1. In the **Host** tab, enter the names of the server and middle tier (if applicable) for the environment. [Table 9-2](#) lists and describes the fields that are required.
2. Click the **Extension Data** tab and then the **Oracle Applications** subtab at the bottom of the window, and complete the fields as shown in [Table 9-2](#). An example of a configured **Oracle Applications** subtab for an Oracle E-Business Suite Release 10 environment is shown in [Figure 9-3](#) on page 221.



Note

The server and client user must have read/write permissions for the PATCH_TOP directory.

Table 9-2. Host tab and Oracle Applications subtab required fields for patch destination environment

Field Name	Value
Host tab, Server section	
Name	Host name of the server to which server-side patches will be applied.
Username	Required during initial copy of the patch file for backend patches.
Password	Required during initial copy of the patch file.
Base Path	Required during initial copy of the patch file.
Host tab, Client section	
Name	Host name of the client (or middle tier) to which middle-tier patches will be applied. This information is used when forms-type patches are applied.
Username	Required during initial copy of the patch file.
Password	Required during initial copy of the patch file for middle-tier patches.
Base Path	Required during initial copy of the patch file.

Table 9-2. Host tab and Oracle Applications subtab required fields for patch destination environment [continued]

Field Name	Value
Extension Data tab, Oracle Applications subtab	
Oracle Apps Version	Oracle Apps R10 or Oracle Apps R11. Oracle Apps R11 applies to both R11 and R11i Oracle instances.
Extension Data tab, Oracle Applications subtab, Application Server (Server Environment) section	
Applmgr User Name	Owner of the Oracle file system objects on the server computer, and the user that executes the AD utilities. This is normally APPLMGR.
Applmgr Password	Password for APPLMGR.
Server Patch Top	Location where all server-side patches being applied to the environment are copied and extracted from their archive formats. For example, <code>APPL_TOP/Patches</code> or <code>/u1/apps/prod</code> .
Database Files	Whether or not the files for installing or upgrading the database are installed in the <code>APPL_TOP</code> area of the server host. For Oracle Apps R11, this value is used during execution of the ADPATCH and ADADMIN utilities, and to determine whether patch detail information will be captured for a given R11 or R11i patch. Patch detail information is captured only for database tiers.
Self-Serv Apps (Required only for R11)	Whether or not the Java and HTML files for self-service applications will be installed in the <code>APPL_TOP</code> area of the server host.
Apps Forms (Required only for R11)	Whether or not the Oracle Forms files are installed in the <code>APPL_TOP</code> area of the server host.
Conc Programs (Required only for R11)	Whether or not the concurrent programs files are installed in the <code>APPL_TOP</code> area of the server host.
Extension Data tab, Oracle Applications subtab, NCA Middle-tier (Client Environment) section	
Applnca User Name	Owner of the Oracle Apps filesystem objects on the middle tier computer. This is normally APPLNCA.
Applnca Password	Password for APPLNCA.
NCA Patch Top	Location where all middle-tier patches being applied to the environment will be copied and extracted from their archive formats. For example, <code>APPL_TOP/Patches</code> or <code>/u1/apps/prod</code> .
Database Files (Required only for R11)	Whether or not the files for installing or upgrading the database are installed in the <code>APPL_TOP</code> area of the middle-tier host.
Self-Serv Apps (Required only for R11)	Whether or not the Java and HTML files for self-service applications are installed in the <code>APPL_TOP</code> area of the middle-tier host.

Table 9-2. Host tab and Oracle Applications subtab required fields for patch destination environment [continued]

Field Name	Value
Apps Forms (Required only for R11)	Whether or not the Oracle Forms files are installed in the APPL_TOP area of the middle-tier host.
Conc Programs (Required only for R11)	Whether or not the concurrent programs files are installed in the APPL_TOP area of the middle-tier host.
Extension Data tab, Oracle Applications subtab, Database section	
Applsys User Name	Database owner of the Oracle Applications Object Library (AOL). This is normally applsys.
Applsys Password	Password for applsys.
System Password	Password for the database system user.
Apps DB Link	The database link defined in the Mercury IT Governance schema that connects to the APPS account of the Oracle instance. This is used in rare cases during patch detail data capture after patch application. This may already be defined if you are using AOL object types to integrate with Object Migrator.

The screenshot shows the 'Environment : OA - OA9U 11i local' window with the following configuration details:

- Environment Name:** OA - OA9U 11i local
- Description:** (empty)
- Location:** (empty)
- Enabled:** Yes
- Host:** Applications | Extension Data | Ownership | User Access | User Data
- Set as default Extension:** Yes
- Enabled:** Yes
- Oracle Apps Version:** Oracle Apps R11
- Application Server (Server Environment):**
 - Applmgr User Name:** applmgr
 - Applmgr Password:** *****
 - Server Patch Top:** /export/home/applmgr/appl/patches
 - Database Files:** Yes
 - Apps Forms:** Yes
 - Self-Serv Apps:** Yes
 - Conc Programs:** Yes
 - UTL Directory Path:** /usr/tmp
- NCA Middle-tier (Client Environment):**
 - Applmca User Name:** applmgr
 - Applmca Password:** *****
 - NCA Patch Top:** /export/home/applmgr/appl/patches
 - Database Files:** Yes
 - Apps Forms:** Yes
 - Self-Serv Apps:** Yes
 - Conc Programs:** Yes
- Database:**
 - Applsyst User Name:** applsys
 - Applsyst Password:** *****
 - OM DB:** OA9U (11.9) Database - std
 - Apps DB Link:** OA9U_APPS_LINK
 - System Password:** *****
- Object Migrator (Multiple Installation Only):**
 - OM Installed:** Yes
 - Conc Request User Name:** (empty)
 - Password:** (empty)
 - Remote OM Environment Name:** (empty)
- App Code Loader:** Launch button
- Database Technology:** Oracle Applications | PeopleSoft | SAP | Siebel
- Buttons:** Check..., OK, Save, Cancel
- Status:** Ready

Figure 9-3. Patch destination environment window, Oracle Applications subtab, sample data

SERVER_ENV_NAME Environment

You need to configure an environment whose name matches the `SERVER_ENV_NAME` parameter in the `server.conf` file of the Mercury IT Governance Center instance. This environment represents the Mercury instance itself and is used during certain FTP and connection functions.

On the **Host** tab, the Username should be the user who owns the Mercury IT Governance instance. The Base Path should be the location of the Mercury IT Governance instance (`ITG_Home`).

Configuring Object Types

This section discusses configuring the following object types:

- R10 Oracle Patch object type
- R11 Oracle Patch object type
- OraApps R11i Merge Patch object type
- Object types that automate the use of the ADADMIN utility



In Release 11i, Oracle added support for a new type of patch driver, the U-driver. Earlier versions of ADMRGPCH will merge these patches but will not merge the U-driver information, which can result in a potentially destructive merged patch. If you need to merge patches that include U-drivers, make sure that you use an instance that is at least at the 11.5.9 or AD.H patch level.

R10 Oracle Patch Object Type

The R10 Oracle Patch object type is used for applying and tracking patches across multiple Oracle instances in a Release 10 environment. This object type obtains the patch file from a central repository and then copies and unpacks it in the Oracle environment that needs patching. The R10 Oracle Patch object type invokes the Oracle ADPATCH utility to run the patch. The object type responds to ADPATCH prompts dynamically, eliminating the need for the user to answer the prompts interactively. A log of the interactive session is captured and remains available in Mercury IT Governance Center.

To configure the R10 Oracle Patch object type:

1. Configure the object type for any special processing required at your site. For example, you might want to execute a site-specific script that gathers database statistics prior to patching.
2. Review the field default values.
3. Test the object type before making it generally available.

For More Information

For more information about this object type, see [R10 Oracle Patch and R11 Oracle Patch Object Types on page 259](#).

R11 Oracle Patch Object Type

The R11 Oracle Patch object type is used for applying and tracking patches across multiple Oracle instances in Release 11 or 11i environments. This object type obtains the patch file from a central repository and then copies and unpacks it in the Oracle environment that needs to be patched. The R11 Oracle Patch object type invokes the Oracle ADPATCH utility to run the patch. The object type responds to ADPATCH prompts dynamically, eliminating the need for the user to answer the prompts interactively. A log of the interactive session is captured and remains available in Mercury IT Governance Center.

To configure the R11 Oracle Patch object type:

1. Configure the object type for any special processing required at your site. For example, you might want to execute a site-specific script that gathers database statistics prior to patching.
2. Review the field default values and display settings, and adjust them to meet your needs. For example, if your Oracle environments are multilingual, you may want to display the Apply Multi-Lingual Patch Now field, or change its default value.
3. If your Oracle instances reside on Windows, add logic to call an environment file to set the required environment variables at the bash level. This script needs to be created manually.
4. To capture auditing information in greater detail, consider making the Patch Drivers field the “Object Revision” for the object type.



Note

In UNIX environments, the object type expects to call an environment file to set the Oracle context. In Windows, the object type expects the environment information to already be set. You may need to modify the object type or define the appropriate environment files in your Oracle instance to make sure that the Oracle environment context is set before the object type invokes ADPATCH.



Note

Review any special processing added to the R11 Oracle Patch object type to see if it should also be applied to the OA - Capture Patch Workflow step source. See [OA - Capture Patch Workflow Step Source on page 228](#).

For More Information

For more information about this object type, see [R10 Oracle Patch and R11 Oracle Patch Object Types on page 259](#).

OraApps R11i Merge Patch Object Type

The OraApps R11i Merge Patch object type allows users to merge multiple compatible Oracle E-Business Suite Release 11i patches into a single patch using Oracle's standard ADMRGPC utility. This decreases the number of individual patches to be tracked and applied and can decrease the total time dedicated to tracking and applying patches.

The OraApps R11i Merge Patch object type executes a file to set the required Oracle environment information. By default it expects the filename to be the Oracle SID as defined for the environment. You need to amend the object type if the default mechanism is not appropriate.

For More Information

For more information about this object type, see [OraApps R11i Merge Patch Object Type on page 263](#).

Object Types That Automate the Use of the ADADMIN Utility

The Extension provides a number of object types that automate the use of the ADADMIN utility for Oracle E-Business Suite Release 11 or 11i. [Table 9-3](#) lists the OraApps ADADMIN-related object types provided with the Extension. Review the field default values and display settings for these object types and tailor them to your needs.

Table 9-3. OraApps ADADMIN-related object types

Object Type	Description
OraApps ADADMIN Compile APPS Schema	Recompiles one or more APPS schemas in an Oracle E-Business Suite R11 or R11i instance, using the ADADMIN utility. It dynamically responds to ADADMIN prompts based on environment and package line information. The object type then navigates the ADADMIN menus, selecting the correct options to compile the APPS schemas.
OraApps ADADMIN Compile Flexfields	Recompiles all flexfields defined in an Oracle E-Business Suite R11 or R11i instance, using the ADADMIN utility. This function may be required after patching or other changes to make sure the flexfield views are accurate.

Table 9-3. OraApps ADADMIN-related object types [continued]

Object Type	Description
OraApps ADADMIN Generate Jar Files	Generates either out of date or all (FORCE option) jar files in an Oracle E-Business Suite R11i instance, using the ADADMIN utility. This task can be run at any time, but is usually run after applying forms patches.
OraApps ADADMIN Generate Messages	Generates binary message files for some or all applications in an Oracle E-Business Suite R11 or R11i instance, using the ADADMIN utility. This task is usually run after applying a patch that requests message regeneration. Oracle uses the message files to display online messages to users. The option to generate messages for specific applications and languages is not available in all version of ADADMIN.



Note

In UNIX environments, these object types execute a file to set the required Oracle environment information. If the default mechanism is not appropriate, you can amend the object type.

For More Information

For more information about these object types, see:

- [OraApps ADADMIN Compile APPS Schema Object Type on page 266](#)
- [OraApps ADADMIN Compile Flexfields Object Type on page 269](#)
- [OraApps ADADMIN Generate Jar Files Object Type on page 271](#)
- [OraApps ADADMIN Generate Messages Object Type on page 273](#)

Configuring Workflows

This section discusses configuring the following workflows:

- OraApps Patch Deployment workflow
- OraApps Patch Data Capture Subworkflow
- OA - Capture Patch workflow execution step source
- Creating a workflow to merge patches

OraApps Patch Deployment Workflow

The OraApps Patch Deployment workflow takes a patch through a proven process of review, application, detail data capture, and verification. If the patch successfully completes these steps, it goes through the full deployment cycle to all specified instances.

To configure the OraApps Patch Deployment workflow:

1. Verify security settings and modify them, as required.

By default, the Assigned User and Group, Package Creator, and OA - Oracle Patch Admin security groups can act on the workflow steps.

The calls to the subworkflow called by this workflow ([OraApps Patch Data Capture Subworkflow on page 227](#)) do not include or require security.

2. Modify the workflow to reflect the actual environments in use. For each environment you add, you need to duplicate the logic of “Apply/Capture Patch/Verify.”
3. Add destination environment values to the Apply step and the Capture Patch step. These steps should occur in a pair for each targeted environment and should have exactly the same destination environment settings.
4. The workflow includes Mercury-supplied notifications that inform users when they need to take action. Customize these notifications as required. You may want to make the notifications reflect the destination environment rather than the generic DEV/TEST/PROD references, or you may want to change the recipients.
5. Restrict the object types that can be used with the workflow to only those that should be used. An example of one to deselect (assuming you are operating in the R10 environment) might be the R11 Oracle Patch OraApps ADADMIN object type.



For each migration step in the OraApps Patch Deployment workflow, the destination environment is the target environment to which the patch will be deployed. For each target environment, the patch archive is always retrieved from the patch repository (Patch Stage environment) and copied to that environment.

OraApps Patch Data Capture Subworkflow

The OraApps Patch Data Capture Subworkflow is called by the OraApps Patch Deployment workflow. This subworkflow captures the details of the patch being applied to a given instance, and associates the patch with the environment. Instances are specified in the OraApps Patch Deployment workflow, not in the subworkflow. The detailed patch data captured by the subworkflow is taken only when the R11 Oracle Patch object type is in use, by default.

To configure the OraApps Patch Data Capture Subworkflow:

1. Verify security settings.
2. Add security on the Record Patch (IMMED) step, if required. By default, the step is available to Assigned User and Group, and OA - Oracle Patch Admin Group.
3. If patching object types do not use the standard name (R11 Oracle Patch), add the names of Release 11 or Release 11i object types to the Check Object Type step. Object types listed here will have patch detail data captured if additional conditions are met, whereas other object types will not.
4. If special handling is required, copy and modify the OA - Capture Patch Workflow step source (described in [OA - Capture Patch Workflow Step Source on page 228](#)). If a copy is made, the workflow step must be removed and re-added to point to the target step source. Changes might include special logic to locate the patch, or special logic to identify when to capture patch details (for example, capturing data for every tier). Make sure that any additional setups needed for the specialized logic are completed.
5. The Data Capture step sends a notification to the OA - Oracle Patch Admin group if it fails. Modify the recipients to meet your needs. Modify the notification text or conditions as required.



Note

Do not specify environments on the subworkflow step. Environments are supplied by the parent workflow.

For More Information

For more information about the workflow that calls the OraApps Patch Data Capture Subworkflow, see [OraApps Patch Deployment Workflow on page 356](#).

OA - Capture Patch Workflow Step Source

Any special processing added to the R11 Oracle Patch object type should be evaluated and applied to the OA - Capture Patch workflow (execution) step source, as appropriate. For example, the C-driver or U-driver is expected to reside in a specific location. If the object type is customized to use a different location, the step source should be similarly changed.

The step source checks the following criteria before attempting to capture patch detail data:

- The type of patch is Backend.
- The environment is Release 11 or 11i.
- The environment is a DB tier.
- The patch is being executed.
- The package line includes a C-driver or U-driver.

Although these criteria fit most needs, the step source can be modified to reflect different criteria, if required. For example, an organization may want to capture patch data for all tiers, not just the database tier.

Creating a Workflow to Merge Patches

If you want to use the OraApps 11i Merge Patch object type to merge multiple Release 11i patches together, you need to create a new workflow.

The workflow should include a standard execution step (for example, DLV Execution with Reset). In this step, the source environment should be the Patch Stage environment, and the destination environment should be the Oracle environment where you want to run ADMRGPCH to merge the patch files.

Add appropriate security to the workflow. Mercury recommends that you limit the workflow to using only the OraApps 11i Merge Patch object type.

Configuring Report Types

This section discusses configuring the following report types:

- OraApps Patch Analysis Report
- OraApps Patch Detail Report

OraApps Patch Analysis Report

The OraApps Patch Analysis Report is similar to the OraApps Patch Detail Report, except that it does not allow users to load in patch data.

When you configure an OraApps Patch Analysis Report, consider that the timeouts on the command steps reflect average-sized patch needs. Increase the timeouts as appropriate to your site and patch analysis style. Of course, larger patches and broader reporting require more time.



Note

Take special care when selecting running fields for the OraApps Patch Analysis Report. A report that includes thousands of records (for example, all files and bugs in a maintenance pack) drains database resources and produces a report of questionable usefulness.

For More Information

For more information about this report, see [OraApps Patch Analysis Report on page 208](#) and [OraApps Patch Analysis Report on page 400](#).

OraApps Patch Detail Report

The OraApps Patch Detail Report lists details of selected patches.

When you configure an OraApps Patch Detail Report, consider the following:

1. The timeouts on the command steps reflect average-sized patch needs. Increase the timeouts, as appropriate, to the customer site and patch analysis style. For example, if MegaPatches, FamilyPacks, and MaintenancePacks are commonly analyzed prior to application, the timeout for the load and purge steps may need to be increased. The timeout for the report itself probably would need to be increased also.
2. If two distinct user groups are expected to use the Patch Detail Report, it may be worthwhile to make a second copy of the report type and hide the load-related fields from users who would never load patches for analysis.

Mercury recommends that file details of maintenance patches never be displayed in this report, because there can be tens of thousands of files in a given patch.

For More Information

For more information about this report, see [OraApps Patch Detail Report on page 208](#) and [OraApps Patch Detail Report on page 403](#).

Special Commands

This section discusses the following special commands:

- `ksc_oa_capture_patch`
- `ksc_oa_purge_temp_patch`

ksc_oa_capture_patch

The `ksc_oa_capture_patch` command extracts the C-driver or U-driver from the patch archive to the Mercury IT Governance Server, validates it, and calls a program to load detailed patch data based on the contents. It is used by the OraApps Patch Detail Report when its Load Data option is set to **Yes**.

ksc_oa_purge_temp_patch

The `ksc_oa_purge_temp_patch` command removes patch detail data for the loaded patch if the patch has been applied in any environments. It is used by the OraApps Patch Details Report when its Remove Data after Run option is set to **Yes**.

Maintaining a Mercury IT Governance System

The following sections discuss the following subjects:

- Maintaining a Mercury IT Governance Server
- Maintaining a Mercury IT Governance database

Maintaining a Mercury IT Governance Server

Some patch-related considerations for maintaining a Mercury IT Governance Server are as follows:

- Patch data can take up considerable space. For example, the C-driver file for the Release 11.5.7 Maintenance Pack is 42 megabytes. Make sure there is adequate space on the Mercury IT Governance Server to save these files during processing. In addition, with large patch files, the log file for processing and the report output for detail listings can also be quite large. Mercury IT Governance Server needs adequate space to hold these files.
- Be sure to regularly monitor the space available for temporary processing, as well as for report and log storage.
- Deleting reports that are no longer required conserves disk space, especially when a patch was loaded for reporting, since the load log file may also be large.

Maintaining a Mercury IT Governance Database

Some patch-related considerations for maintaining the Mercury IT Governance database are as follows:

- If users are commonly loading patches for analysis prior to application and then purging the data, the patch tables can easily become fragmented. Monitor regularly for fragmentation problems, and take the actions required to periodically clean up tables and indexes.
- Patch data can optionally be stored in a separate tablespace from other Mercury IT Governance objects, and Mercury strongly recommends this. Creating separate tablespaces reduces the likelihood that space constraints will affect other processing in the Mercury IT Governance instance, and it reduces system resource contention when processing patch and other data.
- If Mercury IT Governance Center is running in a database that uses cost-based optimization, make sure that the patch-related database objects are analyzed regularly, especially after large data loads or purges.

Appendix

A

Object Types

In This Appendix:

- *Overview of Object Types*
 - *Reference Object Types*
 - *List of Object Types*
 - *Application Data Migration Object Types*
 - *AOL and GL Object Types*
 - *AOL:Single Conc Mgr Entry Object Type*
 - *AOL:Single GUI Menu Entry Object Type*
 - *AOL:Single Report Group Unit Rel 10 Object Type*
 - *AOL:Single Report Group Unit Rel 11 Object Type*
 - *OraApps AKLOAD Migrate AK Setups Object Type*
 - *OraApps Oracle Workflow Object Type*
 - *Patch and Administration Object Types*
 - *R10 Oracle Patch and R11 Oracle Patch Object Types*
 - *OraApps R11i Merge Patch Object Type*
 - *OraApps ADADMIN Compile APPS Schema Object Type*
 - *OraApps ADADMIN Compile Flexfields Object Type*
 - *OraApps ADADMIN Generate Jar Files Object Type*
 - *OraApps ADADMIN Generate Messages Object Type*
 - *Instance Management Object Type*
 - *OraApps 11i Cloning Object Type*
-

Overview of Object Types

This appendix provides reference information about the Oracle-specific object types provided in the Extension. The object types are listed and defined in [Table A-1](#).

You can view or modify an object type as follows:

1. In the Workbench, click **Change Mgmt** and click the **Object Types** icon.
2. If you want, select a particular Extension from the Extension drop-down list.
3. In the Object Type Workbench window, click **List**.
4. Select the object type of interest and click **Open**.
5. Edit the object type. (On the **Fields** tab, the list of fields in the Prompts column is alphabetized.)

Subsequent figures in this appendix show screens you can use to revise object types in conjunction with adding package lines. You can access these screens as follows:

1. In the Workbench, click **Change Mgmt** and click the **Packages** icon.
2. Add a new or open an existing package, as necessary.
3. Select a workflow.
4. Add a line.
5. Select the object type of interest.

Reference Object Types

Reference (REFERENCE) object types cannot be edited, but you can copy and rename them and edit the copies to meet your needs. You can also use existing non-reference object types as is or configure them further to meet your needs.

List of Object Types

The Oracle-related object types fall into the following categories:

- **Application data migration object types.** These object types are used to migrate application setup or configuration data. They integrate Object Migrator or GL Migrator programs with Mercury IT Governance Center, or they automate standard Oracle utilities.
- **Application patching and administration object types.** These object types are used to apply patches and perform standard administration tasks.
- **Instance management object types.** These object types are used to perform instance refreshes.

Table A-1 lists and defines the object types included in the Extension. Each is described in one of the following documents:

- Subsequent sections of this appendix (if Note 1 is indicated in *Table A-1*)
- The *Mercury Object Migrator Guide* (if Note 2 is indicated in *Table A-1*)
- The *Mercury GL Migrator Guide* (if Note 3 is indicated in *Table A-1*)

Table A-1. Object types included in the Extension

Object Type Name	Definition	Note
Application data migration object types		
AOL:Conc Prog	Automates the AOL:Conc Prog migration using Object Migrator.	2
AOL:Concurrent Manager Rel 10	Automates the AOL:Concurrent Manager Rel 10 migration using Object Migrator.	2
AOL:Concurrent Manager Rel 11	Automates the AOL:Concurrent Manager Rel 11 migration using Object Migrator.	2
AOL:Desc Flex	Automates the AOL:Desc Flex migration using Object Migrator.	2
AOL:Folder	Automates the AOL:Folder migration using Object Migrator.	2
AOL:FSG Set	Automates the AOL:FSG Set migration using Object Migrator.	2
AOL:Function	Automates the AOL:Function migration using Object Migrator.	2
AOL:GUI Menu	Automates the AOL:GUI Menu migration using Object Migrator.	2
AOL:Help Text	Automates the AOL:Help Text migration using Object Migrator.	2
AOL:Menu	Automates the AOL:Menu migration using Object Migrator.	2
AOL:Message Rel 10	Automates the AOL:Message Rel 10 migration using Object Migrator.	2
AOL:Message Rel 11	Automates the AOL:Message Rel 11 migration using Object Migrator.	2
AOL:Named SQL	Automates the AOL:Named SQL migration using Object Migrator.	2
AOL:Printer Def	Automates the AOL:Printer Def migration using Object Migrator.	2
AOL:Profile	Automates the AOL:Profile migration using Object Migrator.	2
AOL:QuickCode	Automates the AOL:QuickCode migration using Object Migrator.	2

Table A-1. Object types included in the Extension [continued]

Object Type Name	Definition	Note
AOL:Report Group	Automates the AOL:Report Group migration using Object Migrator.	2
AOL:Report Set Rel 10	Automates the AOL:Report Set Rel 10 migration using Object Migrator.	2
AOL:Report Set Rel 11	Automates the AOL:Report Set Rel 11 migration using Object Migrator.	2
AOL:Resp	Automates the AOL:Resp migration using Object Migrator.	2
AOL:Single Conc Mgr Entry	Adds a single specialization rule to an existing Concurrent Manager definition.	1
AOL:Single GUI Menu Entry	Adds a single entry to an existing GUI Menu definition.	1
AOL:Single Report Group Unit Rel 10	Adds a single unit to an existing Report Group definition.	1
AOL:Single Report Group Unit Rel 11	Adds a single unit to an existing Report Group definition.	1
AOL:User	Automates the AOL:User migration using Object Migrator.	2
AOL:Value Set	Automates the AOL:Value Set migration using Object Migrator.	2
AOL:Zoom	Automates the AOL:Zoom migration using Object Migrator.	2
GL:Budget Orgs	Automates the GL:Budget Orgs migration using GL Migrator.	3
GL:Consolidations	Automates the GL:Consolidations migration using GL Migrator.	3
GL:Cross Validation Rules	Automates the GL:Cross Validation Rules migration using GL Migrator.	3
GL:JE Sources	Automates the GL:JE Sources migration using GL Migrator.	3
GL:Journal Categories	Automates the GL:Journal Categories migration using GL Migrator.	3
GL:Mass Allocation/Budgets	Automates the GL:Mass Allocation/Budgets migration using GL Migrator.	3
GL:Summary Templates	Automates the GL:Summary Templates migration using GL Migrator.	3

Table A-1. Object types included in the Extension [continued]

Object Type Name	Definition	Note
OraApps AKLOAD Migrate AK Setups	Migrates Web setup data using the Oracle AKLOAD utility.	1
OraApps Oracle Workflow	Migrates an Oracle workflow definition from one environment to another using the Oracle WFLOAD utility.	1
Application patching and administration object types		
R10 Oracle Patch	Applies a Release 10 Oracle patch.	1
R11 Oracle Patch	Applies a Release 11 or 11i Oracle patch.	1
OraApps R11i Merge Patch	Allows multiple Oracle patches to be merged into a single patch.	1
OraApps ADADMIN Compile APPS Schema	Uses ADADMIN to compile the APPS schema in the destination environment.	1
OraApps ADADMIN Compile Flexfields	Uses ADADMIN to compile all flexfields in the destination environment.	1
OraApps ADADMIN Generate Jar Files	Uses ADADMIN to generate some or all <code>jar</code> files in the destination environment.	1
OraApps ADADMIN Generate Messages	Uses ADADMIN to generate some or all message files in the destination environment.	1
Instance management object type		
OraApps 11i Cloning	Automates the cloning process for an Oracle E-Business Suite Release 11i system.	1

Note 1 – Documented in subsequent sections of this appendix.

Note 2 – Documented in the *Mercury Object Migrator Guide*.

Note 3 – Documented in the *Mercury GL Migrator Guide*.

Application Data Migration Object Types

The Extension includes an object type corresponding to each specific migrator in Object Migrator (for example, Concurrent Program, Value Set, and Menu) and each specific migrator in GL Migrator (for example, Consolidations and Summary Templates). It also includes object types for patch application, AD administration, workflow migrations and Web setup data. Some object types are specific to particular Oracle releases.

This section provides reference information on selected application data migration object types as indicated in [Table A-1](#):

AOL and GL Object Types

This section provides a description, migration order information, and field definitions for the Oracle Applications Object Library (AOL) and Oracle General Ledger (GL) object types.

Description

The Extension includes object types that integrate Object Migrator or GL Migrator programs with Mercury IT Governance Center. Object Migrator and GL Migrator handle the management and migration of AOL and GL setup entities.

Migration Order

The order of and dependencies among object types (for example, Concurrent Programs are dependent on Value Sets) are specified by the OM/GLM Sequence field on the **Ora Apps** tab in the Workbench Object Type window. *Figure A-1* shows an example.

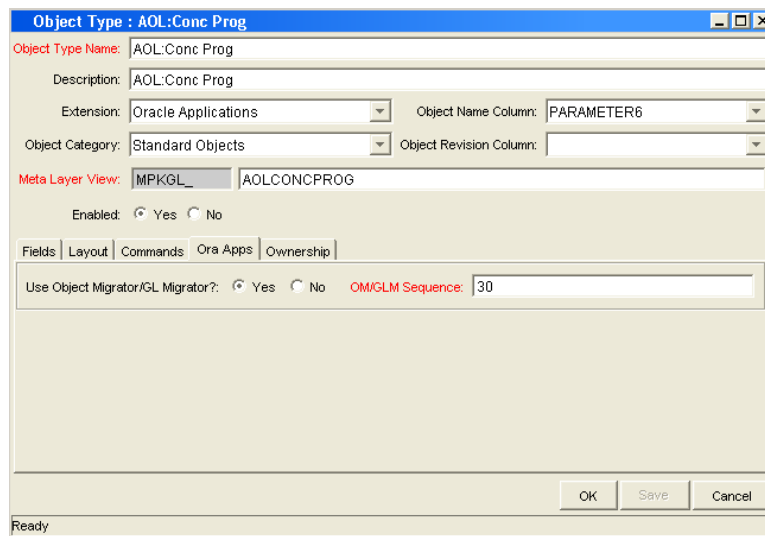


Figure A-1. Object Type window, Ora Apps tab

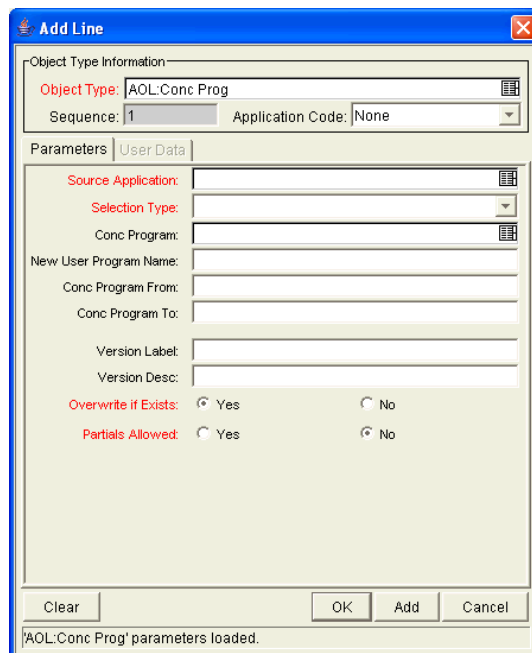
The value entered in the OM/GLM Sequence field determines the order of submission for Oracle-related requests when multiple package lines are submitted for execution as a batch. Typical dependencies are reflected in the values provided, but they can be modified, as required.

Field Definitions

The object type definition contains fields corresponding to each of the fields used by Object Migrator and GL Migrator, although the field layout differs between them.

You can view the fields for each specific migrator in one of two ways:

- By adding a package line using the Workbench as described in [Overview of Object Types on page 234](#) and as shown in the example for the AOL:Conc Prog object type in [Figure A-2](#).
- By querying a specific migrator in the Define Concurrent Programs form in Oracle, as shown in [Figure A-3](#).



The screenshot shows the 'Add Line' dialog box with the following fields and values:

- Object Type Information:**
 - Object Type: AOL:Conc Prog
 - Sequence: 1
 - Application Code: None
- Parameters | User Data:**
 - Source Application: (empty)
 - Selection Type: (empty)
 - Conc Program: (empty)
 - New User Program Name: (empty)
 - Conc Program From: (empty)
 - Conc Program To: (empty)
 - Version Label: (empty)
 - Version Desc: (empty)
 - Overwrite if Exists: Yes No
 - Partials Allowed: Yes No
- Buttons:** Clear, OK, Add, Cancel
- Status Bar:** 'AOL:Conc Prog' parameters loaded.

Figure A-2. AOL:Conc Prog object type sample data

The screenshot shows a dialog box titled "Run this Request..." with a "Copy..." button in the top right. Below the title bar, the "Name" field contains "Migrate Concurrent Programs". A "Parameters" window is open, listing various fields for configuration:

- Source Database: []
- Dest Database: []
- Source Application: []
- Dest Application: []
- Selection Type: 1 (dropdown), Specific Object (checkbox)
- Specific Conc Program: []
- New Conc Program Short Name: []
- New User Program Name: []
- Conc Program From: []
- Conc Program To: []
- Conc Program Like: []
- Save To Archive: []
- Version Label: []
- Version Desc: []
- Recover From Archive: []
- From Version Label: []
- Overwrite if Exists: 1 (dropdown), Yes (checkbox)

Buttons at the bottom of the Parameters window include OK, Cancel, Clear, and Help.

Figure A-3. Migrator fields from Define Concurrent Programs form in Oracle



Note

By default, some of the fields used by Object Migrator and GL Migrator might not be displayed by Mercury IT Governance Center. (For example, the Compare Only field is not displayed.) In most cases this represents the most common usage or most efficient way of integrating with Mercury IT Governance Center, allowing Mercury IT Governance Center's features to determine some values.

AOL:Single Conc Mgr Entry Object Type

This section provides a description, configuration consideration, and field definitions for the AOL:Single Conc Mgr Entry object type.

Description

The AOL:Single Conc Mgr Entry object type allows users to add a single specialization rule to an existing Concurrent Manager definition. The component can be Concurrent Program, User, Oracle ID, Request Type, or Combined Rule. This is useful for minor ongoing maintenance activity that might not follow a classic promotion process.

The concurrent manager to which the item is added, as well as the entry definition values, are all chosen from the source environment. The same items must exist in the target environment for the migration to be successful.

Configuration Consideration

The Apps DB Link value in the Environment window, **Extension Data** tab, **Oracle Applications** subtab must be a valid database link from a Mercury IT Governance Center instance.

Field Definitions

Figure A-4 shows sample data for the AOL:Single Conc Mgr Entry object type when adding a package line. Table A-2 provides definitions for the fields.

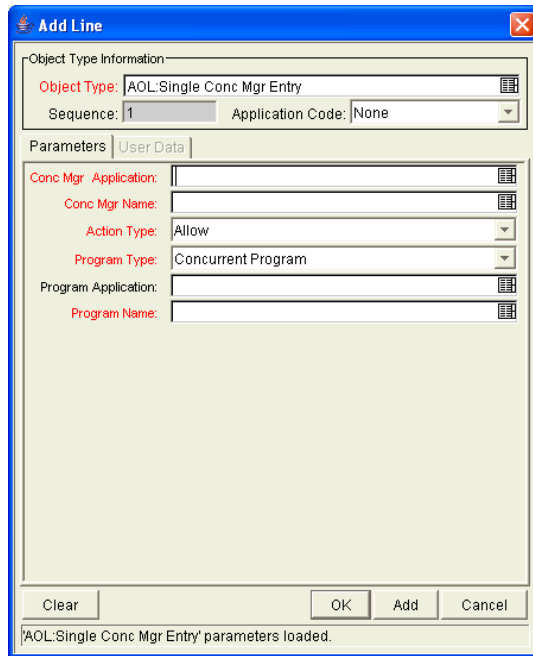


Figure A-4. AOL:Single Conc Mgr Entry object type sample data

Table A-2. AOL:Single Conc Mgr Entry object type field definitions

Field Name	Definition
Conc Mgr Application	Name of the application to which the Concurrent Manager belongs.
Conc Mgr Name	Name of the Concurrent Manager to which an entry is added.
Action Type	Action Type of the entry being added: Allow or Disallow . The default is Allow .
Program Type	Type of entry being added. The values can be Concurrent Program (the default), Request Type , Combined Rule , Oracle ID , or User .
Program Application	Short name of the application to which the entry belongs. Not applicable for Oracle ID and user entries.
Program Name	Specific name identifying the object to add.



Note

This object type was created to be used with Oracle Release 11 or 11i. If you are using it with Release 10, copy it and change the Conc Mgr Name field to use the OA - OM Concurrent Managers Rel 10 Validation.

AOL:Single GUI Menu Entry Object Type

This section provides a description, configuration consideration, and field definitions for the AOL:Single GUI Menu Entry object type.

Description

The AOL:Single GUI Menu Entry object type allows users to add a single component to an existing GUI menu. The component can be a function or a submenu.

The GUI menu to which to add the item and the item definition values are chosen from the source environment. The same menu and items must exist in the target environment for the migration to be successful.

Configuration Consideration

The Apps DB Link value in the Environment window, **Extension Data** tab, **Oracle Applications** subtab must be a valid DB link from the Mercury IT Governance Center instance.

Field Definitions

Figure A-5 shows sample data for the AOL:Single GUI Menu Entry object type when adding a package line. Table A-3 provides definitions for the fields.

Figure A-5. AOL:Single GUI Menu Entry object type sample data

Table A-3. AOL:Single GUI Menu Entry object type field definitions

Field Name	Definition
GUI Menu Name	Name of the GUI Menu to which an entry can be added.
Menu Entry Seq	Sequence to assign to the entry. This should be a number and should not be in use on the menu in the destination.
Prompt	Prompt the user sees when looking at the menu.
SubMenu name	Name of the GUI SubMenu to be added to the menu. It is blank if you are adding a function.
Function Name	Name of the function to add to the menu. It is blank if you are adding a SubMenu.

AOL:Single Report Group Unit Rel 10 Object Type

This section provides a description, configuration consideration, and field definitions for the AOL:Single Report Group Unit Rel 10 object type.

Description

The AOL:Single Report Group Unit Rel 10 object type allows users to add a single component to an existing report group. The component can be an application, concurrent program, or report set.

Use this object type when adding report group components in a Release 10 Oracle environment. There is a separate object type for migrations in Release 11 or 11i environments.

The report group to which to add the item, and the item definition values are all chosen from the source environment. The same group and items must exist in the target environment for the migration to be successful.

Configuration Consideration

The Apps DB Link value in the Environment window, **Extension Data** tab, **Oracle Applications** subtab must be a valid DB link from the Mercury IT Governance Center instance.

Field Definitions

Figure A-6 shows sample data for the AOL:Single Report Group Unit Rel 10 object type when adding a package line. Table A-4 provides definitions for the fields.

The screenshot shows a dialog box titled "Add Line" with a blue border and standard window controls. It is divided into two main sections: "Object Type Information" and "Parameters".

- Object Type Information:**
 - Object Type: AOL:Single Report Group Unit Rel 10
 - Sequence: 1
 - Application Code: None
- Parameters:**
 - Report Group Application: (empty text field)
 - Report Group: (empty text field)
 - Unit Type: Concurrent Program (dropdown menu)
 - Unit Application: (empty text field)
 - Unit Name: (empty text field)

At the bottom of the dialog, there are buttons for "Clear", "OK", "Add", and "Cancel". A status bar at the very bottom reads: "AOL:Single Report Group Unit Rel 10' parameters loaded."

Figure A-6. AOL:Single Report Group Unit Rel 10 object type sample data

Table A-4. AOL:Single Report Group Unit Rel 10 object type field definitions

Field Name	Definition
Report Group Application	Application to which the report group belongs.
Report Group	Report group to which the item will be added.
Unit Type	Type of item being added: Application , Concurrent Program (the default), or Report Set .
Unit Application	Application to use for the new item.
Unit Name	Name of the new item.

AOL:Single Report Group Unit Rel 11 Object Type

This section provides a description, configuration considerations, and field definitions for the AOL:Single Report Group Unit Rel 11 object type.

Description

The AOL:Single Report Group Unit Rel 11 object type allows users to add a single component to an existing report group. The component can be an application, concurrent program, or report set.

Use this object type when adding report group components in a Release 11 or Release 11i environment. A separate object type handles migrations in Release 10 environments.

Configuration Considerations

The Apps DB Link value in the Environment window, **Extension Data** tab, **Oracle Applications** subtab must be a valid DB link from the Mercury IT Governance Center instance.

The report group to which the item is added and the item definition values are all chosen from the source environment.

Field Definitions

Figure A-7 shows sample data for the AOL:Single Report Group Unit Rel 11 object type when adding a package line. Table A-5 provides definitions for the fields.

The screenshot shows a dialog box titled "Add Line" with the following fields and values:

- Object Type Information:
 - Object Type: AOL:Single Report Group Unit Rel 11
 - Sequence: 1
 - Application Code: None
- Parameters:
 - Report Group Application: (empty)
 - Report Group: (empty)
 - Unit Type: Concurrent Program
 - Unit Application: (empty)
 - Unit Name: (empty)

Buttons at the bottom include Clear, OK, Add, and Cancel. A status bar at the bottom reads: "AOL:Single Report Group Unit Rel 11' parameters loaded."

Figure A-7. AOL:Single Report Group Unit Rel 11 object type sample data

Table A-5. AOL:Single Report Group Unit Rel 11 object type field definitions

Field Name	Definition
Report Group Application	Application to which the report group belongs.
Report Group	Report group to which the item will be added.
Unit Type	Type of item being added: Application , Concurrent Program (the default), or Report Set .
Unit Application	Application to use for the new item.
Unit Name	Name of the new item.

OraApps AKLOAD Migrate AK Setups Object Type

This section provides a description, configuration considerations, and field definitions for the OraApps AKLOAD Migrate AK Setups object type.

Description

Oracle provides a mechanism for users to define the layout and content of a Web page that is integrated with Oracle E-Business Suite. This is accomplished by defining the relevant data elements in a series of forms in the AK (Common Modules) module. The definition process takes a long time due to the amount of input required and the performance of the forms used to input the data.

The OraApps AKLOAD Migrate AK Setups object type allows users to automate the migration of custom Web page configuration data using the Oracle AKLOAD utility, thereby saving effort, increasing auditability, and reducing data entry errors. The following types of data can be migrated:

- Attributes
- Objects
- Regions
- Flows

This object type should be used only to migrate custom data.



The AKLOAD utility uses ID values to migrate data. These IDs need to match between source and destination instances. If non-matching IDs are found, the execution fails. You can choose to override this behavior, but this is not recommended, since it can lead to unintended results.

Configuration Considerations

When it runs, the object type:

- Verifies whether all the applications that are in the source environment have the same ID values at the destination.
- If differences are found, and the Check for Application Differences field is set to **Yes**, causes the execution to fail. Otherwise, it prints a warning message in the execution log and continues.

- Connects to the source environment, and extracts the AK data. This may include creation of a temporary directory and script. The execution fails if errors are found in the AKLOAD log.
- Connects to the destination environment, verifies whether the extract directory exists, and creates it if it does not exist.
- Moves the extract file from source to destination.
- If the Clean the Extract Files field is set to **Yes**, connects to the source environment and removes the extract file.
- Connects to the destination environment and checks whether the extract file contains any non-application raw ID values.
- If the Check for Raw IDs field is set to **Yes** and raw IDs are found, causes the execution to fail. Otherwise it prints a warning message in the execution log and continues with the execution.
- Imports the AK data. This may include creation of a temporary directory and script. The execution fails if errors are found in the AKLOAD log.
- If the Clean the Extract Files field is set to **Yes**, connects to the destination environment and removes the extract file.



Note

Only custom data should be migrated with this object type. The object type does not validate or prevent migration of standard or Mercury-supplied data.



Note

By default, this object type is configured to connect using JDBC. The object type can be configured to use other protocols supported by AKLOAD.



Note

Data should be migrated between Oracle instances that are the same point release and patch level. The object type does not check the Oracle version.

Field Definitions

Figure A-8 shows sample data for the OraApps AKLOAD Migrate AK Setups object type when adding a package line. *Table A-6* provides definitions for the fields.

The screenshot shows a dialog box titled "Add Line" with the following fields and values:

- Object Type Information:**
 - Object Type: OraApps AKLOAD Migrate AK Setups
 - Sequence: 1
 - Application Code: None
- Parameters / User Data:**
 - Application: (empty)
 - Business Object Type: (empty)
 - Sub-Path: kintana_akload_extracts/
 - File Name: (empty)
 - Business Object: (empty)
 - Object Name for Audit: (empty)
 - Download from Source: Yes No
 - Upload to Destination: Yes No
 - Upload Options: UPDATE:Update database per the flatfile d:
 - Get Options: GET:Download specific Business Objects
 - Clean the Extract files?: Yes No
 - Check for Application Differences?: Yes No
 - Check for Raw ID differences?: Yes No

Buttons at the bottom: Clear, OK, Add, Cancel. Status bar: 'OraApps AKLOAD Migrate AK Setups' parameters loaded.

Figure A-8. OraApps AKLOAD Migrate AK Setups object type sample data

Table A-6. OraApps AKLOAD Migrate AK Setups object type field definitions

Field Name	Definition
Application	Application from which objects are downloaded and uploaded.
Business Object Type	Type of business object being migrated: REGION, FLOW, ATTRIBUTE, or OBJECT . This value must be reselected when the Application selection changes.
Sub-Path	Subdirectory under the base path where the extract file should go.
File Name	Extract file name. (.jlt extension)
Business Object	Specific business entity being migrated. More than one entity from the specified Application may be selected. This value depends on the values for Application and Business Object Type, and must be re-selected when either of these values change.
Object Name for Audit	Name that will be used as an audit trail for the AKLOAD migration. It will show up in the environment contents and history reporting for the object type. By default, the value selected will be the same as the Business Object field, but delimited with commas rather than semicolons.
Download from Source	Export is performed only if this field is set to Yes .
Upload to Destination	Import is performed only if this field is set to Yes .
Upload Options	Upload options for import.
Get Options	Download options for import.
Clean the Extract files	Whether or not to remove the extract files from the source and destination after successful migration.
Check for Application Differences	If this field is set to Yes , all Source Applications' short names and application IDs are validated against the destination DB, and an error results if they do not match.
Check for Raw ID Differences	If this field is set to Yes , all object_id and menu_id values in the extract file are validated against the destination DB, and an error results if they do not match.



Note

Any database views upon which your Web setup depends should already exist in the destination instance. Their existence is not validated by the object type.



Note

Migration of security data is not supported. The security information can be migrated using the AOL:Resp migrator.

OraApps Oracle Workflow Object Type

This section provides a description, configuration considerations, and field definitions for the OraApps Oracle Workflow object type.

Description

The OraApps Oracle Workflow object type allows users to migrate embedded Oracle workflow definitions between Oracle E-Business Suite Release 11i instances. By using Mercury IT Governance Center to migrate these definitions, users can increase the security, control, and visibility of these changes, and integrate with a source code control system.

This object type uses the standard Oracle WFLOAD utility to accomplish the upload or download of the workflow definition. Three migration types are supported (all three types can be integrated with source code control tools):

- **Upload from File**—The object moves an existing workflow definition file from the source environment to the destination environment and loads it into the destination database. The file is left at the destination.
- **Download to File**—The workflow definition is downloaded from the source database to the source server.
- **Upload from Database**—The workflow definition is downloaded from the source database, moved to the destination server, and then uploaded to the destination database. The temporary file used to transport the definition is removed.



The workflow versions in the source and destination should be the same. This is not validated by the object type.

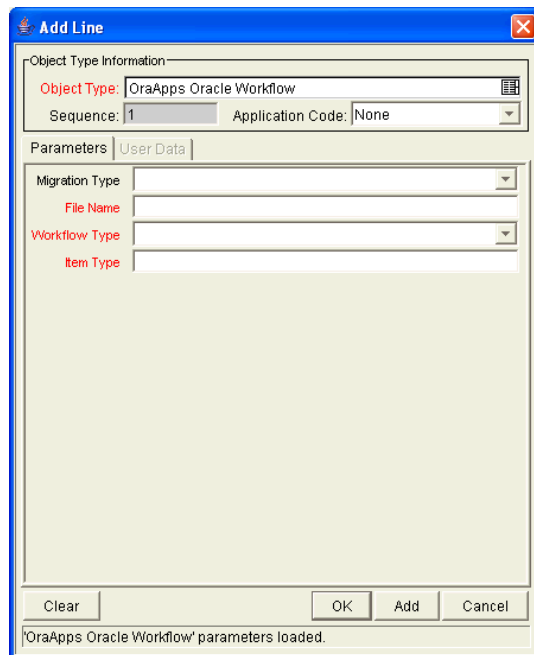
Configuration Considerations

Any environment defined in Mercury IT Governance Center that will be a source or target of a workflow migration needs to have a valid value in the UTL_FILE_DIR field in the Environment window, **Extension Data** tab, **Oracle Applications** subtab. This field defines the location where workflows will be downloaded to or uploaded from, and should correspond to a UTL_FILE_DIR entry in the environment database's `init.ora` file.

Each usage option can be configured for use with a source control tool. The **Upload from File** option can be integrated to retrieve the file from a source control system. The **Download to File** option can be configured to check the file into a source control system. The **Upload from Database** option can be configured to check in the downloaded definition to a source control system. If this is required, the derivation and storage of a temporary filename should be removed.

Field Definitions

Figure A-9 shows sample data for the OraApps Oracle Workflow object type when adding a package line. *Table A-7* provides definitions for the fields.



The screenshot shows a dialog box titled "Add Line" with a close button in the top right corner. The dialog is divided into several sections:

- Object Type Information:**
 - Object Type: OraApps Oracle Workflow
 - Sequence: 1
 - Application Code: None
- Parameters:**
 - User Data (selected tab)
 - Migration Type (dropdown menu)
 - File Name (text input field)
 - Workflow Type (dropdown menu)
 - Item Type (text input field)
- Buttons:** Clear, OK, Add, Cancel
- Status Bar:** OraApps Oracle Workflow' parameters loaded.

Figure A-9. OraApps Oracle Workflow object type sample data

Table A-7. OraApps Oracle Workflow object type field definitions

Field Name	Definition
Migration Type	Type of migration: Upload from File, Upload from Database, or Download to File.
File Name	Name of workflow file with extension (.wft).
Workflow Type	Oracle Application Seeded Workflow, New Custom Workflow, or Changed to Custom Workflow.
Item Type	Workflow Item Type (for example, OEOL/OEOH). Multiple values are allowed and must be space delimited.

Patch and Administration Object Types

With a small amount of configuration, it is possible to automate a large portion of the patch application and tracking process using the Extension patch management tools. The Extension automates Oracle's ADPATCH and ADADMIN utilities through the standard Mercury IT Governance Center workflow and execution engines, with sophisticated extras such as rules-based intelligence.

The Extension applies Oracle server-side and middle-tier patches that use the ADPATCH program. It cannot be used to apply Oracle database server patches or Oracle 10SC (client-side) patches.

This section provides reference information on the patch and administration object types listed in [Table A-1 on page 236](#):

R10 Oracle Patch and R11 Oracle Patch Object Types

This section provides a description and field definitions for the R10 Oracle Patch object type and the R11 Oracle Patch object type.

Description

The Extension contains two different types of object types for applying patches: the R10 Oracle Patch object type and the R11 Oracle Patch object type. If you attempt to apply a Release 10 patch to a Release 11 environment, or vice versa, the migration fails. For more information, see [Chapter 9, *Managing Patches*](#), on page 197.



Note

Oracle's product documentation includes specific instructions for applying patches using unified drivers to multi-tier Oracle E-Business Suite instances. If the workflows you are using to apply patches to your Oracle E-Business Suite instances conflict with Oracle's guidelines, you should modify the workflows prior to applying patches with unified drivers.



Note

In UNIX environments, the object type expects to call an environment file to set the Oracle context. On Windows, the object type expects the environment information to be already set. You may need to modify the object type or define the appropriate environment files in your Oracle instance to make sure that the Oracle environment context is set prior to the object type's invoking ADPATCH.

Field Definitions

[Figure A-10](#) and [Figure A-11](#) show sample data for the default R10 Oracle Patch and default R11 Oracle Patch object types when adding a package line. [Table A-8](#) provides definitions for the displayed fields and for some fields that are hidden by default.

Add Line

Object Type Information

Object Type: R10 Oracle Patch

Sequence: 1 Application Code: None

Parameters | User Data

Patch Type: Ora Apps Backend Server

Bug No.:

Patch Archive:

Tar Number:

Copy Patch to Dest? Yes No

Execute Patch @Dest? Yes No

Restart? Yes No

Drv Files (*.drv, *.drv ...): patch.drv

Log File Name: adpatch_[P.P_BUG_NO].log

No. of Parallel Workers: 4

Clear OK Add Cancel

R10 Oracle Patch' parameters loaded.

Figure A-10. R10 Oracle Patch object type sample data

Add Line

Object Type Information

Object Type: R11 Oracle Patch

Sequence: 1 Application Code: None

Parameters | User Data

Patch Type: Ora Apps Backend Server

Bug No.:

Patch Archive:

Tar Number:

Copy Patch to Dest? Yes No

Execute Patch @Dest? Yes No

Restart? Yes No

Drv Files (*.drv, *.drv ...): patch.drv

Driver Execution Options: All Portions

Log File Name: adpatch_[P.P_BUG_NO].log

No. of Parallel Workers: 4

Batch Size: 1000

Email Adpatch Failures? Yes No

Email Recipients (spc delim): applmgr

Terminate on Worker Failure? Yes No

Reapply Patch? Yes No

Tolerate Applications System Name conflict? Yes No

Clear OK Add Cancel

R11 Oracle Patch' parameters loaded.

Figure A-11. R11 Oracle Patch object type sample data

Table A-8. R10 Oracle Patch and R11 Oracle Patch object types field definitions

Field Name	Definition
Patch Type	The OraApps Backend Server setting causes the source or destination environment's server information to be used for the patch. The OraApps Forms Server setting causes the source or destination environment's client information to be used for the patch.
Bug No	The bug number for which the patch is relevant. Used for auditing.
Patch Archive	The file from which the patch is to be extracted.
Tar Number	Tracking No. From Oracle Support.
Copy Patch To Dest	Whether or not the patch files will be copied to each environment being patched. Set to No if the patch already exists in each physical environment.
Execute Patch @Dest	Whether or not the patch will be executed in the destination environment.
Restart	In the event of patch failure, indicates if ADPATCH should continue from the point of failure, or restart.
Drv Files (*.drv, *.drv ...)	Names of the driver files to be used in the patch application, separated by commas or spaces.
Driver Execution Options (Applies only to R11 or R11i Oracle Patches)	Indicates which portions of the patch driver to execute based on the kind of driver specified. Specific portions of the driver can be chosen only when a U-driver is specified, and should be chosen in accordance with Oracle's patch instructions. Options are All Portions , Copy Portion , Database Portion , or Generate Portion .
Log File Name	Name of the file in the Oracle instance to which the patch application output will be written.
No. of Parallel Workers	Used by the Oracle ADPATCH utility to determine the number of parallel processes used to handle the patch.
Batch Size (Applies only to R11 or R11i Oracle Patches)	Used by the Oracle ADPATCH utility to determine processing batch size.
Email Adpatch Failures (Applies only to R11 or R11i Oracle Patches)	Standard with ADPATCH. Indicates whether or not email notifications of failures should be sent. This is not required if notifications have been set up for the workflow, which is the recommended way of sending email notifications with Mercury products.
Email Recipients (spc delim) (Applies only to R11 or R11i Oracle Patches)	Email recipients receiving ADPATCH failure notifications. Space delimited.

Table A-8. R10 Oracle Patch and R11 Oracle Patch object types field definitions [continued]

Field Name	Definition
Apply Multi-Lingual Patch Now (Hidden by default. Applies only to R11 or R11i Oracle patches.)	The default value is No . Select Yes to apply this multilingual patch. Oracle's ADPATCH utility requests confirmation when applying patches to multilingual environments, because multiple patch files may be required.
Terminate on Worker Failure (Applies only to R11 or R11i Oracle Patches)	The default value is No . If the value is No , Mercury IT Governance Center continues to wait for outside intervention to fix failed workers. Select Yes to have the patch execution fail if all workers fail.
Append Patch Data (Hidden by default. Applies only to R11 or R11i Oracle Patches.)	Whether or not the entire C-driver file and U-driver file are evaluated to derive patch detail data if the patch already exists in the detail tables. Initially Yes , set this field to No after the patch is captured the first time for the package line.
Reapply Patch	Whether or not to reapply the patch if it is already applied to the destination environment. Default is No . To reapply the patch, set this field to Yes .
Tolerate Applications System Name conflict	Whether or not to allow the ADPATCH utility to continue without error if the Applications System Name stored in the database and the name in the APPL_TOP file system are different, which usually indicates a misconfiguration. Default is No .

OraApps R11i Merge Patch Object Type

This section provides a description, configuration considerations, and field definitions for the OraApps R11i Merge Patch object type.

Description

The OraApps R11i Merge Patch object type allows users to merge multiple compatible Oracle Release 11i patches into a single patch using the Oracle standard ADMRGPCH utility. This decreases the number of individual patches to be tracked and applied, and can decrease the total time dedicated to tracking and applying patches.

This object type takes specified patch archive files from the Patch Stage central repository, uses ADMRGPCH to merge them into a single patch in an Oracle Release 11i instance, and then returns the merged patch archive to the patch stage repository.

The object type produces a new patch archive with a user-specified name that can be used with the “Oracle Patch” object types to apply the patch.

AD Merge Patch does not merge patches of different releases, different platforms, or different parallel modes.

AD Merge Patch does not consider any dependencies regarding the order in which patches should be applied. Users must be careful regarding the patches they merge together if there are application-order dependencies.

Configuration Considerations

The patch archives to be merged must exist in the patch stage area, and the Patch Stage environment must be defined. An environment must be defined where ADMRGPCH can run. If patches with Unified drivers will be merged, the environment should support Unified drivers.

Since merging patches is a one-time-only occurrence, it may be useful to create a simple workflow just to accomplish the merge.

The Merged Archive name assigned will be used by Patch Applicator to apply the merged patch, and will identify the patch in the `Environment History` file. You should select a meaningful name.

Field Definitions

Figure A-12 shows sample data for the OraApps R11i Merge Patch object type when adding a package line. *Table A-9* provides definitions for the fields.

The screenshot shows a dialog box titled "Add Line" with a blue title bar and a close button. The dialog is divided into two main sections: "Object Type Information" and "Parameters".

Object Type Information:

- Object Type: OraApps R11i Merge Patch
- Sequence: 1
- Application Code: None

Parameters:

- Archives to be Merged: (empty text field)
- Merged Archive (New Bug#): (empty text field)
- Merged Archive Type: zip (dropdown menu)
- Tar Numbers: (empty text field)
- Overwrite Flag: Yes No
- Partial Allowed: Yes No

At the bottom of the dialog, there are four buttons: "Clear", "OK", "Add", and "Cancel". A status bar at the very bottom reads: "OraApps R11i Merge Patch' parameters loaded."

Figure A-12. OraApps R11i Merge Patch object type sample data

Table A-9. OraApps R11i Merge Patch object type field definitions

Field Name	Definition
Archives to be Merged	List of archives that needs to be merged. File names must be separated by commas or spaces. The list can contain up to a total of 200 characters. (A shorter length may be required depending upon the limitations of your operating system, because the list is used in operating system commands.)
Merged Archive (New Bug#)	Name of the Merged Patch without file extension. This value should be a valid UNIX file name and should not contain any spaces or control characters. This value is used as a Bug Number when the Merged Patch archive is being applied through Patch Applicator, and also for patch detail reporting.
Merged Archive Type	The type of the merged archive to be created (for example, zip or tar).
Tar Numbers	List of <code>tar</code> numbers for which patches are being merged. For Information and reporting purpose only.
Overwrite Flag	Whether or not to overwrite an existing merged file.
Partial Allowed	Whether or not to create the merged patch if some of the patch archives mentioned in the list of archives could not be found in the patch stage area.

OraApps ADADMIN Compile APPS Schema Object Type

This section provides a description, configuration considerations, and field definitions for the OraApps ADADMIN Compile APPS Schema object type.

Description

The OraApps ADADMIN Compile APPS Schema object type recompiles one or more APPS schemas in an Oracle R11 or R11i instance by invoking the ADADMIN utility. The object type responds to ADADMIN prompts dynamically, based on environment and package line information, eliminating the need for the user to answer the prompts interactively. The object type then navigates the ADADMIN menus, selecting the correct options to compile the APPS schemas.

ADADMIN spawns parallel workers to recompile invalid entities in the APPS schemas. This task can be run at any time, but is usually run after applying packages that alter packages in the APPS schema, or after installing custom packages that need compilation. If more than one APPS schema is defined, ADADMIN compiles all the schemas. After executing, the object type exits ADADMIN. A log of the interactive session is attached to the Mercury IT Governance Center execution log.

Configuration Considerations

In UNIX environments, the object type executes a file to set the required Oracle environment information. You can modify the object type if the default mechanism is not appropriate.

You must set Oracle context information prior to invoking ADADMIN.



The ADADMIN functions are supported on Release 11 and 11i only. Attempting to administer a Release 10 environment with these object types results in a failed migration.

Field Definitions

Figure A-13 shows sample data for the OraApps ADADMIN Compile APPS Schema object type when adding a package line. Table A-10 provides definitions for the displayed fields and for some fields that are hidden by default.

The screenshot shows a dialog box titled "Add Line" with a close button in the top right corner. The dialog is divided into two tabs: "Parameters" (selected) and "User Data".

Object Type Information:

- Object Type: OraApps ADADMIN Compile APPS Schema
- Sequence: 1
- Application Code: None

Parameters:

- Environment tier: Ora Apps Backend Server
- No. of Parallel Workers: 4
- Invoker Rights Mode: Incremental
- Restart?: Yes No
- Log File Name: adadmin_[PKG.NUMBER].log
- Batch Size: 1000
- Email Adadmin Failures?: Yes No
- Email Recipients (spc delim): applmgr
- Terminate on Worker Failure?: Yes No

Buttons at the bottom: Clear, OK, Add, Cancel.

Status bar: 'OraApps ADADMIN Compile APPS Schema' parameters loaded.

Figure A-13. OraApps ADADMIN Compile APPS Schema object type sample data

Table A-10. OraApps ADADMIN Compile APPS Schema object type field definitions

Field Name	Definition
Environment tier	The OraApps Backend Server setting causes the object to use the environment's server information. The OraApps Forms Server setting causes the object to use the environment's client information.
Disregard init.ora errors (Hidden by default)	Select Yes to continue running ADADMIN if it detects potential issues with <code>init.ora</code> . The default value is No .
No. of Parallel Workers	Number of workers to run in parallel.
Invoker Rights Mode	Mode for invoker rights.
Restart	Whether or not to restart an interrupted session. Mercury strongly recommends retaining the default setting of No .
Log File Name	Log name to be created in Oracle instance.
Batch Size	Size of the batch to run.
Email Adadmin Failures	Whether or not Oracle should send an email if ADADMIN fails.
Email Recipients (spc delim)	List of users who should get email notices. Space delimited.
Terminate on Worker Failure	If the value is No (the default), Mercury IT Governance Center continues to wait for outside intervention to fix failed workers. Select Yes to have the patch execution automatically fail if all workers fail.

OraApps ADADMIN Compile Flexfields Object Type

This section provides a description and field definitions for the OraApps ADADMIN Compile Flexfields object type.

Description

The OraApps ADADMIN Compile Flexfields object type recompiles all flexfields defined in an Oracle R11 or R11i instance, using the ADADMIN utility. This function may be required after patching or other changes in order to make sure that the flexfield views are accurate.



This ADADMIN function is supported on Release 11 or 11i only. Attempting to administer a Release 10 environment with this object type will result in a failed migration.

Field Definitions

Figure A-14 shows sample data for the OraApps ADADMIN Compile APPS Schema object type when adding a package line. *Table A-11* provides definitions for the displayed fields and for some fields that are hidden by default.

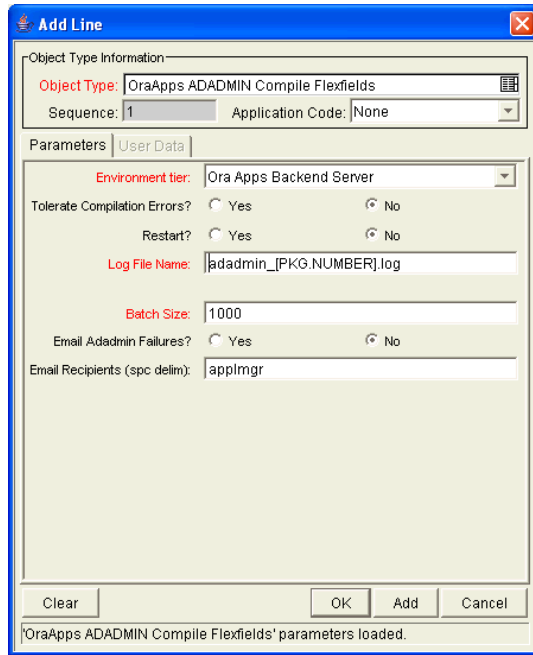


Figure A-14. OraApps ADADMIN Compile Flexfields object type sample data

Table A-11. OraApps ADADMIN Compile Flexfields object type field definitions

Field Name	Definition
Environment tier	The OraApps Backend Server setting causes the object to use the environment's server information. The OraApps Forms Server setting causes the object to use the environment's client information.
Disregard Init.ora Errors (Hidden by default)	Select Yes to continue running ADADMIN if it detects potential issues with <code>init.ora</code> . The default value is No .
Tolerate Compilation Errors	Whether compilation errors are treated as a failure (No), or whether they are tolerated and the execution completes successfully even if compilation errors are encountered (Yes).
Restart	Whether to restart an interrupted session. Mercury strongly recommends retaining the default setting of No .
Log File Name	Log name to be created in the Oracle instance.
Batch Size	Size of the batch.
Email Adadmin Failures	Whether or not Oracle should send an email if ADADMIN fails.
Email Recipients (spc delim)	List of recipients for email notices.

OraApps ADADMIN Generate Jar Files Object Type

This section provides a description and field definitions for the OraApps ADADMIN Generate Jar Files object type.

Description

The OraApps ADADMIN Generate Jar Files object type generates either out of date or all (FORCE option) `jar` files in an Oracle R11i instance, using the ADADMIN utility. This task can be run at any time, but is usually run after applying forms patches.



This ADADMIN function is supported on Release 11i only. Attempting to administer a Release 10 or 11 environment with this object type results in a failed execution.

Field Definitions

Figure A-15 shows sample data for the OraApps ADADMIN Generate Jar Files object type when adding a package line. *Table A-12* provides definitions for the fields.

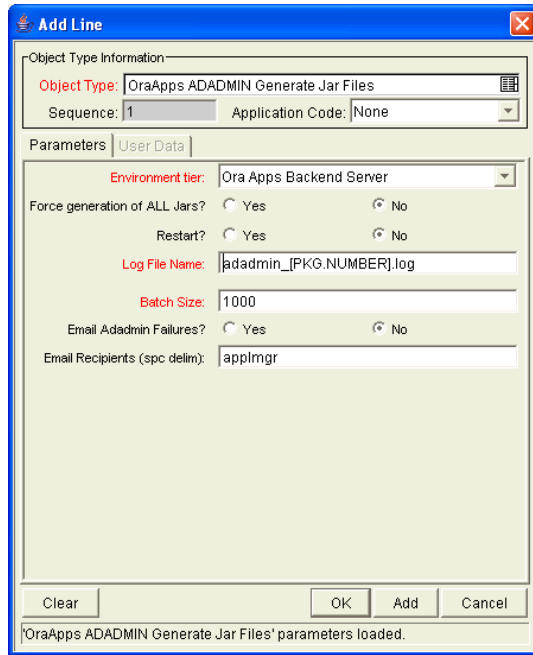


Figure A-15. OraApps ADADMIN Generate Jar Files object type sample data

Table A-12. OraApps ADADMIN Generate Jar Files object type field definitions

Name	Definition
Environment tier	The OraApps Backend Server setting causes the object to use the environment’s server information. The OraApps Forms Server setting causes the object to use the environment’s client information.
Force generation of ALL Jars	Whether or not to force generation of all jar files. Selecting Yes forces generation of all jar files.
Restart	Whether to restart an interrupted session. Mercury strongly recommends retaining the default setting of No .
Log File Name	Log name to be created in Oracle instances.
Batch Size	Size of the batch.
Email Adadmin Failures	Whether Oracle should send an email if ADADMIN fails.
Email Recipients (spc delim)	List of recipients for email notices.



Generating all jar files can take a considerable length of time. The timeout for this object type should be adjusted to reflect the user’s environment.

OraApps ADADMIN Generate Messages Object Type

This section provides a description and field definitions for the OraApps ADADMIN Generate Messages object type.

Description

The OraApps ADADMIN Generate Messages object type uses the Oracle ADADMIN utility to generate binary message files for applications in an Oracle R11 or R11i instance. This task is usually run after applying a patch that requests message regeneration, but it can be run on an incremental basis. Oracle uses the message files to display online messages to users. Some versions allow the user to specify the products (applications) and languages for which to generate messages, while earlier versions always generate messages for all languages.



The ADADMIN functions are supported on Release 11 and 11i only. Attempting to administer a Release 10 environment with these object types results in a failed migration.

Field Definitions

Figure A-16 shows sample data for the OraApps ADADMIN Compile APPS Schema object type when adding a package line. Table A-13 provides definitions for the fields.

The screenshot shows a dialog box titled "Add Line" with a blue title bar. The main content area is divided into sections. The "Object Type Information" section at the top contains a text field for "Object Type" with the value "OraApps ADADMIN Generate Messages", a "Sequence" field with the value "1", and an "Application Code" dropdown menu set to "None". Below this is a "Parameters" section with a "User Data" tab selected. The parameters include: "Environment tier" (dropdown: Ora Apps Backend Server), "Restart?" (radio buttons: Yes, No, with "No" selected), "Log File Name" (text field: |adadmin_[PKG.NUMBER].log), "Batch Size" (text field: 1000), "Products (inv ap ...)" (text field: all), "Languages (US DK ...)" (text field: all), "No. of Parallel Workers" (text field: 4), "Terminate on Worker Failure?" (radio buttons: Yes, No, with "No" selected), "Email Adadmin Failures?" (radio buttons: Yes, No, with "No" selected), and "Email Recipients (spc delim)" (text field: applmgr). At the bottom of the dialog are "Clear", "OK", "Add", and "Cancel" buttons. A status bar at the very bottom reads: "'OraApps ADADMIN Generate Messages' parameters loaded."

Figure A-16. OraApps ADADMIN Generate Messages object type sample data

Table A-13. OraApps ADADMIN Generate Messages object type field definitions

Name	Definition
Environment tier	The OraApps Backend Server setting causes the object to use the environment's server information. The OraApps Forms Server setting causes the object to use the environment's client information.
Restart	Whether to restart an interrupted session. Mercury strongly recommends retaining the default setting of No .
Log File Name	Log name to be created in Oracle instance.
Batch Size	Size of the batch.
Products (inv ap...)	List of products (application codes) for which to generate messages, or generates messages for all products. Multiple values should be space delimited. Default value is all . Applicable only if ADADMIN allows generation of messages for specific products.
Languages (US DK...)	List of languages (language codes) for which to generate messages, or generates messages in all installed languages. Multiple values should be space delimited. Default value is all . Applicable only if ADADMIN allows generation of messages for specific languages.
No. of Parallel Workers	Number of parallel processes used to handle the administration activity. Applicable only if ADADMIN uses parallel workers for message generation.
Terminate on Worker Failure	Yes if Patch Applicator should force a termination when all workers fail, or No if it should continue to wait.
Email Adadmin Failures	Whether or not Oracle should send an email if ADADMIN fails.
Email Recipients (spc delim)	List of recipients for email notices.

Instance Management Object Type

This section provides reference information on the (one) instance management object type listed in [Table A-1](#).

OraApps 11i Cloning Object Type

This section provides a description, configuration considerations, and field definitions for the OraApps 11i Cloning object type.

Description

The OraApps 11i Cloning object type is used in the process of cloning instances. Using Oracle E-Business Suite requires management of a number of ongoing operating instances and temporary implementation and upgrade instances. These instances must be efficiently created, maintained, and periodically cloned without risking the stability of the system and disruption of day-to-day activities.

The OraApps 11i Cloning object type provides a means to clone an Oracle E-Business Suite instance by automating the execution of major steps in the process. Information pertinent to the environments to be refreshed is captured in the object type. This information is to be used during the execution of commands that clone the instance. Only the databases and directories targeted to be cloned are affected. This object type works in conjunction with the OraApps 11i Cloning workflow.

When the object type is run, it executes the following actions based on which step the workflow is currently on:

- Verifies if all cloning requirements are met. This includes checking for adequate disk space, the existence of Perl on the destination machine, whether the Oracle patch for cloning has been applied to the destination instance, and whether the files required for handling the database exist.
- Copies and executes cloning scripts in pre-clone mode. This preserves configuration information regarding the destination instance and shuts down the various application services.
- Generates a list of database files to be deleted and shuts down the destination database in preparation for the clone.
- Deletes all database files in the list.
- Copies all database files from the source to the destination.

- Recreates control files needed to be able to get the database in working order.
- Restarts the database.
- Cleans up concurrent request tables that may contain executions relevant only to the source instance.
- Deletes and copies application files from the source to the destination.
- Copies and executes cloning scripts in post-clone mode. This restores the configuration information of the destination instance and restarts the various application services.

The process implemented by this object type and its associated workflow follow the guidelines set in the *Cloning Oracle Applications Release 11i* white paper published by Oracle.

Configuration Considerations

If any of the standard workflow step names in the OraApps 11i Cloning workflow are altered, you need to apply the proper modifications to the OraApps 11i Cloning object type command condition, because the various commands execute based on the workflow step name.

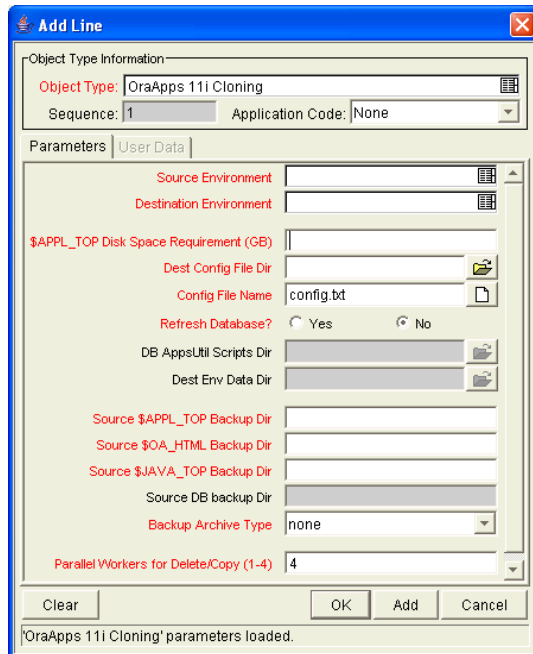


Note

Cloning your Oracle instance is a complex process, and misconfigurations can yield unexpected consequences. This object type and its associated workflow provide a template for cloning your instances, but requires tailoring for your environment. Mercury recommends that you utilize Mercury's Consulting Services to facilitate a successful implementation.

Field Definitions

Figure A-17 shows sample data for the OraApps 11i Cloning object type when adding a package line. Table A-14 provides definitions for the fields.



The screenshot shows a dialog box titled "Add Line" with a close button in the top right corner. The dialog is divided into two main sections: "Object Type Information" and "Parameters".

Object Type Information:

- Object Type: OraApps 11i Cloning
- Sequence: 1
- Application Code: None

Parameters:

- Source Environment: [Empty text box]
- Destination Environment: [Empty text box]
- \$APPL_TOP Disk Space Requirement (GB): [Empty text box]
- Dest Config File Dir: [Empty text box]
- Config File Name: config.bt
- Refresh Database?: Radio buttons for Yes and No. The No button is selected.
- DB AppsUtil Scripts Dir: [Empty text box]
- Dest Env Data Dir: [Empty text box]
- Source \$APPL_TOP Backup Dir: [Empty text box]
- Source \$OA_HTML Backup Dir: [Empty text box]
- Source \$JAVA_TOP Backup Dir: [Empty text box]
- Source DB backup Dir: [Empty text box]
- Backup Archive Type: none
- Parallel Workers for Delete/Copy (1-4): 4

At the bottom of the dialog are four buttons: Clear, OK, Add, and Cancel. Below the buttons, a status bar reads: "OraApps 11i Cloning" parameters loaded.

Figure A-17. OraApps 11i Cloning object type sample data

Table A-14. OraApps 11i Cloning object type field definitions

Field Name	Definition
Source Environment	Source environment to be used for cloning.
Destination Environment	Destination environment to be used for cloning.
\$APPL_TOP Disk Space Requirement (GB)	Amount of space (in gigabytes) required at the destination.
Dest Config File Dir	Location of the installation configuration file used by the destination instance.
Config File Name	Name of the installation configuration file.
Refresh Database	Whether or not to refresh the database as well as the application code.
DB AppsUtil Scripts Dir	Location of database script files <code>addbctl.sh</code> and <code>addlnctl.sh</code> . These files are used to stop and start the listener and database.
Dest Env Data Dir	Destination data file directory.
Source \$APPL_TOP Backup Dir	Directory where the source \$APPL_TOP code backup files are stored.
Source \$OA_HTML Backup Dir	Directory where the source \$OA_HTML code backup files are stored.
Source \$JAVA_TOP Backup Dir	Directory where the source \$JAVA_TOP code backup files are stored.
Source DB backup Dir	Directory where the source database backup files are stored.
Backup Archive Type	Type of archiving used to back up source files (none , tar , or zip).
Parallel Workers for Delete/Copy (1-4)	Number of parallel workers to be used for delete and copy processes.

Appendix B Request Types

In This Appendix:

- *Overview of Request Types*
- *Overview of the Request Process*
- *Reference Request Types*
- *List of Request Types*
- *OraApps Application Issue Request Type*
 - *Description*
 - *Configuration Considerations*
 - *Field Definitions*
- *OraApps Cloning Request Request Type*
 - *Description*
 - *Configuration Consideration*
 - *Field Definitions*
- *OraApps Conversion Request Request Type*
 - *Description*
 - *Configuration Consideration*
 - *Field Definitions*
- *OraApps Design & Development Request Type*
 - *Description*
 - *Configuration Considerations*
 - *Field Definitions*
- *OraApps Enhancement Request Request Type*
 - *Description*
 - *Configuration Consideration*
 - *Field Definitions*
- *OraApps GAP Analysis Request Request Type*

- *Description*
 - *Configuration Consideration*
 - *Field Definitions*
 - *OraApps Interface Request Request Type*
 - *Description*
 - *Configuration Consideration*
 - *Field Definitions*
 - *OraApps Report Request Request Type*
 - *Description*
 - *Configuration Consideration*
 - *Field Definitions*
 - *OraApps Setup Change Request Request Type*
 - *Description*
 - *Configuration Consideration*
 - *Field Definitions*
 - *OraApps Status Update Request*
 - *Description*
 - *Configuration Considerations*
 - *Field Definitions*
-

Overview of Request Types

This appendix provides reference information about the Oracle-specific request types provided in the Extension. The request types are described in alphabetical order as shown on the previous page.

Migration and compilation of request types can be driven by commands included within the request types. For more information about commands in the Mercury IT Governance Center environment, see the *Commands, Tokens, and Validations Guide and Reference*. For more information about using request types in packages, see the *Mercury Change Management User's Guide*.

Subsequent figures of request type sample data show screens you can use to create requests from existing request types. You can access this screen as follows:

1. Make sure that the desired request type is enabled. In the Workbench, click **Demand Mgmt** and click the **Request Types** icon. If the desired request type needs to be enabled, open and enable it.
2. In the standard interface, select **Create > Request**. The Create New Request page appears.
3. Select the Request Type of interest from the drop-down list and click **Create**.

All requests follow the same general process, which is described in the following section. Each request type works in conjunction with a particular workflow that is also part of the Extension.

Overview of the Request Process

When functionality (for example, a gap analysis, report, setup change, or cloning request) is required for a certain business area, a business analyst is notified of this need by the group or individual with the requirement.

The business analyst gathers detailed information related to the request. These detailed requirements are checked for appropriateness and completeness, and are then forwarded to an approver (probably a business owner). The business owner looks at the requirements, asks for more information, if necessary, and then accepts or rejects them.

At this stage or at any of the later stages of the process, information such as screen shots, reports, and comments can be attached to the request. A Mercury request type consolidates all relevant information for a request in one location so that requestors, analysts, and approvers can all access it.

If the requirements are accepted, they are prioritized and matched with existing functionality (if such functionality exists). A solution review team, usually consisting of the business owner and the track owner, reviews the existing functionality and confirms or rejects it as a suitable solution to the request.

If the existing functionality matches the requirements, the business owner is notified and the request is closed.

If the existing functionality does not match the requirements, the track owner is notified to prioritize the request, which now becomes a request for development.

The track owner prioritizes the request and begins the detailed functional design phase. Functional designs are created and signed off, and the team lead is notified for approval.

The team lead may request more information about the functional designs, and this dialog may lead to design changes. Modified functional designs need to go back through the signoff process.

When the team lead accepts the design, a developer or team of developers is assigned. The developer creates technical designs, and these designs are reviewed against the requirements. During the technical approval phase, the technical designs may need to be modified.

When the technical design is approved, development begins. Until deployment is complete, the request waits at the package creation step. When the development is complete, the request closes and the original requestor is notified.

Each workflow step in the process has security groups assigned to it, so only authorized persons can act on those steps. At each approval step, package creation and completion notifications can be sent to appropriate users.

Subsequent figures in this appendix show screens you can use to create requests from existing request types. You can access these screens as follows:

1. Make sure that the desired request type is enabled. In the Workbench, click **Demand Mgmt** and click the **Request Types** icon. If the desired request type needs to be enabled, open and enable it.
2. In the standard interface, select **Create > Request**. The Create New Request page appears.

Reference Request Types

Reference (REFERENCE) request types cannot be edited, but you can copy and rename them and edit the copies to meet your needs. You can also use existing non-reference request types as is or configure them further to meet your needs.

List of Request Types

Table B-1 lists the request types included in the Extension. Each is described in subsequent sections.

Table B-1. Request types included in the Extension

Request Type Name	Definition
OraApps Application Issue	Tracks Oracle issues and prompts requestors for relevant information about the environment.
OraApps Cloning Request	Streamlines the process of initiating and tracking a request to clone an Oracle E-Business Suite instance.
OraApps Conversion Request	Initiates the process of converting data from legacy systems or third-party software for importing into Oracle systems.
OraApps Design & Development	Consolidates information about Oracle configuration and customization requirements and the analysis needed to review and approve them.
OraApps Enhancement Request	Initiates the process of requesting enhancements for an Oracle system.
OraApps GAP Analysis Request	Initiates the process of requesting a gap analysis of the differences between existing company processes and those of Oracle E-Business Suite.
OraApps Interface Request	Initiates the process of requesting a new interface to import or export data between Oracle tables and other software programs.
OraApps Report Request	Initiates the process of developing a new or modified report for an Oracle system.
OraApps Setup Change Request	Initiates the process of requesting a new setup for an Oracle system.
OraApps Status Update Request	Initiates the process of gathering and reviewing status information.

OraApps Application Issue Request Type

This section provides a description, configuration considerations, and field definitions for the OraApps Application Issue request type.

Description

During an Oracle E-Business Suite implementation or upgrade, there is a constant influx of new problems, questions, and requests. There is usually no central repository of information to store the issues, and they can originate from a number of different sources. Issues can “fall through the cracks” and end up being stored in testing scripts, design documents, or emails. Unless these issues are captured and addressed quickly, they often do not get resolved and end up creating a backlog that jeopardizes the success of the project.

The OraApps Application Issue request type is used to track issues at an Oracle implementation site. It provides a means for entering specific information needed to resolve an issue. Requestors are prompted for such Oracle environment information as the application in which the issue is occurring, the browser being used, and the operating system. This request type is designed to work with the OraApps Application Issue workflow.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For more information about the reports associated with the OraApps Application Issue request type, see [OraApps Apps Issues Summary Report on page 387](#) and [OraApps Apps Issues Detail Report on page 383](#).

For more information about the workflow associated with this request type, see [OraApps Application Issue Workflow on page 324](#).

Configuration Considerations

Most of the fields in the Details section of the OraApps Application Issue request type refer to a drop-down list of values. You can tailor each of these validation lists to give you the list of values appropriate to your business.

The default Details fields for an OraApps Application Issue request type are suggested information you might want to capture about Oracle Application issues. However, you can remove and add fields as required. If you remove fields, you must make corresponding changes to the reports associated with the Oracle issue, because these reports reference some of the Details fields.

You may decide to make some of the fields required or not required at different stages of issue resolution. You may also decide to place security on individual fields. You make these changes in the Request Type window.

You should also consider the following before you use or configure the OraApps Application Issue request type:

- The request type must be enabled before it can be used.
- By default, all users can create this type of request.
- The OraApps Application Issue workflow should be the default workflow for this request type and the request type should use only this workflow.
- Most of the Details fields in this request type refer to a drop-down list of values, which can be configured as required.
- You can configure security levels for this request type.

Field Definitions

Figure B-1 shows the screen for creating an OraApps Application Issue request based on the provided request type. *Table B-2* contains field definitions for the request type.

Create New OraApps Application Issue

Expand All Collapse All Submit Cancel

Header

Summary

Details

Problem/Resolution

***Problem:**

***Business Area Affected:** **Required Date:**

***Source:** **Source Category:** **Project Team:**

Resolution:

Solution:

Environment

Environment:

Apps Application: **Browser Version:**

Apps Responsibility: **Desktop OS:**

Apps Username: **Initiator Version:**

Apps Form: **DB Version:**

Analysis

Business Benefit: **Technical Impact:**

Level of Effort: **Est Completion Date:**

Scope Change: Yes **Internal/External:** Internal

TAR #: **Oracle Bug No.:**

Notes

References

Submit Cancel

Figure B-1. OraApps Application Issue request creation

Table B-2. OraApps Application Issue request type field definitions

Field Name	Definition
Summary tab (not expanded in Figure B-1)	
Workflow	OraApps Application Issue
Problem/Resolution tab	
Problem	Summary of the specific problem, issue, or enhancement request.
Business Area Affected	General business group or department affected by the issue.
Source	Origin of the issue, such as GAP Analysis , CRP A , or Integration Test .

Table B-2. OraApps Application Issue request type field definitions [continued]

Field Name	Definition
Required Date	Required date by which the issue must be resolved.
Source Category	Nature of the issue. The field has Mercury-supplied values such as Form , Interface , and Report .
Project Team	Project team responsible for the functional area addressed in the issue.
Resolution	Fixed set of ways the issue can be resolved. The field can have Mercury-supplied values such as Implemented , Not Reproducible , and Out of Scope .
Solution	Detailed resolution of the issue.
Environment tab	
Environment	Name of the environment with which there is an issue.
Apps Application	Oracle Application in the environment experiencing the issue.
Apps Responsibility	Oracle Responsibilities in the environment experiencing the issue.
Apps Username	Oracle Applications user in the environment experiencing the issue.
Apps Form	Name of the Oracle Applications Form with which there is an issue.
Browser Version	Browser version being used when the issue was found.
Desktop OS	Desktop operating system.
Jinitiator Version	Jinitiator version.
DB Version	DB version of the database.
Analysis tab	
Business Benefit	Level of business benefit (High , Medium , or Low) estimated by the request creator.
Level of Effort	Level of effort (Very Easy , Easy , Medium , Hard , Very Hard) estimated by the assigned developer.
Scope Change	Whether the problem is out of the original scope and needs a scope change to be resolved.
Internal/External	Whether the request needs internal or external resolution. External indicates that there is an Oracle patch required to solve the request.
TAR #	Unique identifier for the TAR. This information is provided by Oracle Support.
Technical Impact	Level of system impact (High , Medium , or Low) estimated by the analyst or assigned developer.
Est Completion Date	Estimated completion date for resolving the request.
Oracle Bug No.	Oracle bug number for the issue.

OraApps Cloning Request Request Type

This section provides a description, configuration consideration, and field definitions for the OraApps Cloning Request request type.

Description

The OraApps Cloning Request request type lets you streamline the process of initiating and tracking a request to have an Oracle instance cloned. This request type works with the OraApps 11i Cloning object type and the OraApps Cloning Process workflow.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For more information about the object type associated with the OraApps Cloning Request request type, see [OraApps 11i Cloning Object Type on page 276](#).

For more information about the workflow associated with this request type, see [OraApps Cloning Process Workflow on page 329](#).

Configuration Consideration

The OraApps Cloning Process workflow should be the default and the only allowed workflow assigned to the OraApps Cloning Request request type.

Field Definitions

Figure B-2 shows the screen for creating an OraApps Cloning Request based on the provided request type. *Table B-3* contains field definitions for the request type.

Create New OraApps Cloning Request

Expand All Collapse All Submit Cancel

Header

Summary

Details

Business Details

Business Area: Business Benefit:

*New Environment: Yes No Application:

*Business Justification:

*Source Environment: Dest Environment:

*Database Only ? Yes No *Requested By:

Analysis

Technical Impact: Est. Effort:

Est. Start Date: Est. Completion Date:

Analysis:

Notes

References

Submit Cancel

Figure B-2. OraApps Cloning Request creation

Table B-3. OraApps Cloning Request request type field definitions

Field Name	Definition
Summary tab (not expanded in Figure B-2)	
Workflow	OraApps Cloning Process
Business Details tab	
Business Area	General business group or department requesting the clone.
Business Benefit	Level of business benefit (High , Medium , or Low) the cloning initiative is expected to have.
Application	List of applications under a business area that are requesting the cloning.
New Environment	Whether or not a new destination environment is required.
Business Justification	Justification for creating a clone of an instance.
Source Environment	Source environment used for cloning.
Destination Environment	Destination environment used for cloning.
Database Only	Whether or not only the database needs to be refreshed.
Requested By	Requested completion date.
Analysis tab	
Technical Impact	Level of technical impact (High , Medium , or Low) the cloning initiative is expected to have.
Est. Effort	Number of estimated hours required for the cloning effort.
Est. Start Date	Estimated date the cloning will begin.
Est. Completion Date	Estimated date the cloning will be complete and ready for use.
Analysis	Analysis of the requested task.

OraApps Conversion Request Request Type

This section provides a description, configuration consideration, and field definitions for the OraApps Conversion Request request type.

Description

Organizations using Oracle E-Business Suite may have used other legacy systems for similar functionality in the past. The data from these systems needs to be transformed and imported into Oracle using interfaces, and the correct tables need to be populated with this data. For example, an organization that decides to use Oracle CRM needs to obtain all the contact information from legacy systems and put it into Oracle before it can start using Oracle for CRM.

Organizations using Oracle E-Business Suite may also have used third-party software applications in the past. In order to be brought into Oracle, the data from these applications needs to go through conversion, as well. In both these examples, data conversion is a one-time-only occurrence.

The OraApps Conversion Request request type initiates the process of converting data from either legacy systems or third-party software into an Oracle system. This request type works in conjunction with the OraApps Conversion Process workflow.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For more information about the workflow associated with the OraApps Conversion Request request type, see [OraApps Conversion Process Workflow on page 331](#).

Configuration Consideration

The OraApps Conversion Process workflow should be the default and the only allowed workflow assigned to the OraApps Conversion Request request type.

Field Definitions

Figure B-3 shows the screen for creating an OraApps Conversion Request based on the provided request type. *Table B-4* contains field definitions for the request type.

Create New OraApps Conversion Request

Expand All Collapse All Submit Cancel

Header

Summary

Details

Conversion Details

*Conversion Name:

Business Area: Application:

*Requested By:

*Description Detail:

External System Name: External System Owner:

Analysis

Technical Effort: Est. Effort (hrs):

Est. Completion Date: Project Team:

Analysis:

Notes

References

Submit Cancel

Figure B-3. OraApps Conversion Request creation

Table B-4. OraApps Conversion Request request type field definitions

Field Name	Definition
Summary tab (not expanded in Figure B-3)	
Workflow	OraApps Conversion Process
Conversion Details tab	
Conversion Name	Name of the conversion.
Description Detail	Detailed description of the conversion.
Business Area	Business area requesting the conversion.
Application	Application under the business area requesting the conversion.
Requested By	Date by which the conversion is needed.
External System Name	External system name.
External System Owner	External system owner.
Analysis tab	
Technical Effort	Technical effort required for the conversion of data.
Est. Effort	Estimated effort (in hours) required for conversion.
Est. Completion Date	Estimated completion date.
Analysis	Details of the conversion analysis.
Project Team	Project team assigned.

OraApps Design & Development Request Type

This section provides a description, configuration considerations, and field definitions for the OraApps Design & Development request type.

Description

During a typical Oracle E-Business Suite implementation or upgrade, organizations devote a significant amount of time and effort to designing, developing, and rolling out new or updated configurations and changes. The high volume of configurations and customizations is difficult to track and control. A process for managing the configurations and customizations is necessary for a successful implementation.

Before any development work is done, it is essential that the specific requirements driving the change be identified and reviewed. It is also important that the requirements be approved and prioritized before significant design and development time is spent. Without this, resources, time, and money are spent in areas that do not significantly help the business while higher priority needs are left unattended or with inadequate resources.

The OraApps Design & Development request type gathers summary information about the requirement, including information such as a summary description, specific business areas being addressed, and the business benefit the request is expected to provide. It also gathers the major analysis information that is needed when reviewing and approving the requirement. Finally, through notes and attachments, the request type serves as the central repository for detailed requirements and justification documents, so that everyone in the design and development process has access to all the information surrounding the specific effort.

The OraApps Design & Development request type is designed to work in conjunction with the OraApps Design & Development workflow.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For more information about the workflow associated with the OraApps Design & Development request type, see [OraApps Design & Development Workflow on page 341](#).

Configuration Considerations

The OraApps Design & Development workflow should be the default and the only allowed workflow assigned to the OraApps Design & Development request type.

Most of the Details fields in the OraApps Design & Development request type refer to a drop-down list of values. You can tailor each of these validation lists to give you the list of values appropriate to your business.

If you want to gather additional information during the design and development process, you can add fields to the OraApps Design & Development request type.

Field Definitions

Figure B-4 shows the screen for creating an OraApps Design & Development request based on the provided request type. *Table B-5* contains field definitions for the request type.

The screenshot shows a web form titled "Create New OraApps Design & Development". At the top right of the form area are "Submit" and "Cancel" buttons. Below the title bar are expand/collapse controls and a list of sections: Header, Summary, Details, and Dev. Details. The "Details" section is expanded and contains the following fields:

- *Application: (dropdown menu)
- Business Area: (dropdown menu)
- Business Benefit: (text input)
- Level of Effort: (text input)
- Est. Completion Date: (text input)
- *Summary: (text area)
- Unit Category: (dropdown menu)
- *Source: (dropdown menu)
- Technical Impact: (text input)
- Estimated Effort: (text input)

At the bottom of the form are sections for "Notes" and "References", and another set of "Submit" and "Cancel" buttons.

Figure B-4. OraApps Design & Development request creation

Table B-5. OraApps Design & Development request type field definitions

Field Name	Definition
Summary tab (not expanded in Figure B-4)	
Workflow	OraApps Design & Development
Dev. Details tab	
Application	List of the applications by which design and development units should be categorized. This can be set to be general categories such as Manufacturing or Finance or more granular ones such as individual Oracle Application codes
Business Area	General business group or department that will be affected by the unit category. While multiple groups might be affected, this field signifies the group that is most affected.
Business Benefit	Level of business benefit (High , Medium , or Low) the cloning initiative is expected to have.
Level of Effort	Length of time it will take to design and develop this new unit.
Est. Completion Date	Estimated date the unit will be complete and ready for deployment.
Summary	Short summary of the unit. This allows a user to quickly get to the overview of the unit without going through the notes or opening up any attachments.
Unit Category	Technical nature of the unit. Mercury supplies such values as Form, Interface, and Report.
Source	List of where the request for this unit came from. This could include values such as GAP Analysis, CRP 1, or Conversion Test.
Technical Impact	Level of business impact (High , Medium , or Low) the cloning initiative is expected to have. Example: A new report might have a low impact, while an update to a major form might have a high impact.
Estimated Effort	Number of days required for the design and development effort.

OraApps Enhancement Request Request Type

This section provides a description, configuration considerations, and field definitions for the OraApps Enhancement Request request type.

Description

As business requirements within an Oracle system change, the existing functionality must be modified or new functionality enhanced. Proposed enhancements to the system go through a process of business and technical review and approval.

The OraApps Enhancement request type initiates the process of requesting enhancements for an Oracle system. This request type works with the OraApps Enhancement Process workflow.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For more information about the workflow associated with the OraApps Enhancement request type, see [OraApps Enhancement Request Request Type on page 300](#).

Configuration Consideration

The OraApps Enhancement Process workflow should be the default and the only allowed workflow assigned to the OraApps Enhancement Request request type.

Field Definitions

Figure B-5 shows the screen for creating an OraApps Enhancement Request based on the provided request type. *Table B-6* contains field definitions for the request type.

Create New OraApps Enhancement Request

Expand All Collapse All Submit Cancel

Header

Summary

Details

Enhancement Details

*Enhancement Name:

*Detailed Description:

*New Enhancement: Yes No Business Area:

*Requested By: Application:

Analysis

Technical Effort: Est. Effort (hrs):

Est. Completion Date: Project Team:

Analysis:

Notes

References

Submit Cancel

Figure B-5. OraApps Enhancement Request creation

Table B-6. OraApps Enhancement Request request type field definitions

Field Name	Definition
Summary tab (not expanded in Figure B-5)	
Workflow	OraApps Enhancement Process
Enhancement Details tab	
Enhancement Name	Name of the requested enhancement.
Detailed Description	Detailed description of the enhancement.
New Enhancement	Whether or not a new enhancement is required.
Business Area	Business area requesting the enhancement.
Requested By	Date by which the enhancement is required to be completed.
Application	Application under the business area requesting the enhancement.
Analysis tab	
Technical Effort	Technical effort required for the enhancement.
Est. Effort (hrs)	Estimated effort (in hours) required for the enhancement.
Est. Completion Date	Estimated completion date.
Analysis	Details of the enhancement analysis.
Project Team	Project team assigned.

OraApps GAP Analysis Request Request Type

This section provides a description, configuration consideration, and field definitions for the OraApps GAP Analysis Request request type.

Description

An implementation or upgrade in an Oracle system can mean significant changes to the system's business processes. The difference between the existing processes and those provided by the packaged application is referred to as a "gap." Stakeholders in an Oracle-related project need to be able to request a "gap analysis" for specific Oracle functionality.

The OraApps GAP Analysis Request request type initiates the process of requesting a gap analysis for an Oracle system. Project managers, track owners, and project sponsors use this request type to create a gap analysis.

Once a request for a gap analysis has been created, it should be attached to relevant Release 11i project tasks.

The OraApps GAP Analysis Request request type works with the OraApps GAP Analysis Process workflow, which consists of approval, assignment, and analysis steps.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For more information about the workflow associated with the OraApps GAP Analysis Request request type, see [OraApps GAP Analysis Process Workflow on page 348](#).

Configuration Consideration

The OraApps GAP Analysis Process workflow should be the default and the only allowed workflow assigned to the OraApps GAP Analysis Request request type.

Field Definitions

Figure B-6 shows the screen for creating an OraApps GAP Analysis Request based on the provided request type. *Table B-7* contains field definitions for the request type.

Create New OraApps GAP Analysis Request

Expand All Collapse All Submit Cancel

Header

Summary

Details

GAP Details

*GAP Name:

Business Area: Application:

*Requested By:

*Business Requirement:

*GAP Description:

Analysis

Technical Effort: Est. Effort (hrs):

Est. Completion Date: Project Team:

Analysis

Notes

References

Submit Cancel

Figure B-6. OraApps GAP Analysis Request creation

Table B-7. OraApps GAP Analysis Request request type field definitions

Field Name	Definition
Summary tab (not expanded in Figure B-6)	
Workflow	OraApps GAP Analysis Process
GAP Details tab	
GAP Name	Name of the gap.
Requested By	Date by which the gap analysis needs to be completed.
Business Area	Business area requesting the gap analysis.
Application	Application under the business area requesting the gap analysis.
Business Requirements	Description of the business requirements.
GAP Description	Description of the gap.
Analysis tab	
Technical Effort	Technical effort required for the gap analysis.
Est. Effort (hrs)	Estimated effort (in hours) required for the gap analysis.
Est. Completion Date	Estimated completion date of the gap analysis.
Analysis	Gap analysis solution details.
Project Team	Project team assigned.

OraApps Interface Request Request Type

This section provides a description, configuration consideration, and field definitions for the OraApps Interface Request request type.

Description

An Oracle E-Business Suite instance may have to use data from other ERP systems as well as third-party software for different functionality. Interfaces must be created that allow data to be extracted from other software programs to populate Oracle tables, and export data from Oracle to other software programs.

The OraApps Interface Request request type initiates the process of requesting a new interface for an Oracle system. This request type works with the OraApps Interfaces Process workflow.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For more information about the workflow associated with the OraApps Interface Request request type, see [OraApps Interfaces Process Workflow on page 352](#).

Configuration Consideration

The OraApps Interfaces Process workflow should be the default workflow and the only allowed workflow assigned to the OraApps Interface Request request type.

Field Definitions

Figure B-7 shows the screen for creating an OraApps Interface Request based on the provided request type. Table B-8 contains field definitions for the request type.

The screenshot shows a web form titled "Create New OraApps Interface Request". At the top right, there are "Submit" and "Cancel" buttons. Below the title, there are expand/collapse controls for sections: "Header", "Summary", "Details", and "Interface Details". The "Interface Details" section is expanded and contains the following fields:

- Interface Name:** Text input field.
- Requested By:** Text input field with a user selection icon.
- Business Area:** Dropdown menu.
- Application:** Dropdown menu.
- Description Detail:** Large text area with scrollbars.
- External System Name:** Text input field.
- External System Owner:** Text input field.
- Update Frequency:** Text input field.
- Interface Type:** Dropdown menu with "Extract From Oracle" selected.

Below the "Interface Details" section, there is an "Analysis" section with the following fields:

- Technical Effort:** Dropdown menu.
- Est. Effort (hrs):** Text input field.
- Est. Completion Date:** Text input field with a date selection icon.
- Project Team:** Dropdown menu.
- Analysis:** Large text area with scrollbars.

At the bottom of the form, there are sections for "Notes" and "References", and "Submit" and "Cancel" buttons.

Figure B-7. OraApps Interface Request creation

Table B-8. OraApps Interface Request request type field definitions

Name	Definition
Summary tab (not expanded in Figure B-7)	
Workflow	OraApps Interfaces Process
Interface Details tab	
Interface Name	Name of the requested interface.
Description Details	Detailed description of the interface.
Business Area	Business area requesting the interface.
Application	Application under the business area requesting the interface.
Requested By	Date by which the interface needs to be completed.
External System Name	Name of the external system.
External System Owner	Owner of the external system.
Update Frequency	How frequently the interface is updated or run.
Interface Type	Type of interface.
Analysis tab	
Technical Effort	Technical effort required for the interface generation.
Est. Effort	Estimated effort (in hours) required for the interface generation.
Est. Completion Date	Estimated completion date of the interface generation.
Analysis	Analysis for the interface generation.
Project Team	Project team assigned.

OraApps Report Request Request Type

This section provides a description, configuration consideration, and field definitions for the OraApps Report Request request type.

Description

Although Oracle E-Business Suite provides many reports, these standard reports are often insufficient to meet a company's business needs. For example, the inclusion of information captured in a Descriptive Flexfield may be important, or reporting on transactional data based on company-specific criteria. When a report does not currently exist that can provide the information that is required, a new report must be created.

The OraApps Report Request request type initiates the process of developing a new or modified report for an Oracle system. This request type works with the OraApps Reports Process workflow.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For more information about the workflow associated with the OraApps Report Request request type, see [OraApps Reports Process Workflow on page 362](#).

Configuration Consideration

The OraApps Reports Process workflow should be the default and the only allowed workflow assigned to the OraApps Report Request request type.

Field Definitions

Figure B-8 shows the screen for creating an OraApps Report Request based on the provided request type. *Table B-9* contains field definitions for the request type.

The screenshot displays a web form titled "Create New OraApps Report Request". At the top right, there are "Submit" and "Cancel" buttons. Below the title, there are expand/collapse controls for sections: "Header", "Summary", "Details", and "Report Details". The "Report Details" section contains the following fields:

- Report Name:** A text input field.
- Requested By:** A text input field with a user selection icon.
- New Report:** Radio buttons for "Yes" and "No", with "No" selected.
- Business Area:** A dropdown menu.
- Application:** A dropdown menu.
- Description Detail:** A large text area with a vertical scrollbar.

Below this is the "Analysis" section, which includes:

- Technical Effort:** A dropdown menu.
- Est. Effort (hrs):** A text input field.
- Est. Completion Date:** A date input field with a calendar icon.
- Project Team:** A dropdown menu.
- Analysis:** A large text area with a vertical scrollbar.

At the bottom, there are sections for "Notes" and "References", and another pair of "Submit" and "Cancel" buttons.

Figure B-8. OraApps Report Request creation

Table B-9. OraApps Report Request request type field definitions

Field Name	Definition
Summary tab (not expanded in Figure B-8)	
Workflow	OraApps Reports Process
Report Details tab	
Report Name	Name of the requested report.
New Report	Whether or not the report requested is new (Y) or a modification (N).
Business Area	Business area requesting the report.
Requested By	Date by which the report must be available.
Application	Application under the business area requesting the report.
Description Detail	Detailed description of the report.
Analysis tab	
Technical Effort	Technical effort required for the report generation.
Est. Completion Date	Estimated completion date of the report generation.
Est. Effort (hrs)	Estimated effort (in hours) required for the report generation.
Project Team	Project team assigned.
Analysis	Analysis for the report generation.

OraApps Setup Change Request Request Type

This section provides a description, configuration consideration, and field definitions for the OraApps Setup Change Request request type.

Description

Oracle systems use a number of different application setups to satisfy their business requirements. These application setups (referred to as “setup changes”) include modules such as Accounting Periods, Asset Categories, and Inventory Orgs. Each environment and operating unit has its own set of modules, and needs its own setup changes.

The OraApps Setup Change Request request type initiates the process of requesting a new setup for an Oracle system. This request type works with the OraApps Setup Change Process workflow.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For more information about the workflow associated with the OraApps Setup Change Request request type, see [OraApps Setup Change Process Workflow on page 366](#).

Configuration Consideration

The OraApps Setup Change Process workflow should be the default and the only allowed workflow assigned to the OraApps Setup Change Request request type.

Field Definitions

Figure B-9 shows the screen for creating an OraApps Setup Change Request based on the provided request type. *Table B-10* contains field definitions for the request type.

Create New OraApps Setup Change Request

Expand All Collapse All Submit Cancel

Header

Summary

Details

Setup Details

*Setup Name:

Business Area: Application:

*Requested By:

*Setup Description:

*Environment Name: Operating Unit:

Analysis

Technical Effort: Est. Effort (hrs):

Est. Completion Date: Project Team:

Dependencies:

Notes

References

Submit Cancel

Figure B-9. OraApps Setup Change Request creation

Table B-10. OraApps Setup Change Request request type field definitions

Field Name	Definition
Summary tab (not expanded in Figure B-9)	
Workflow	OraApps Setup Change Process
Setup Details tab	
Setup Name	Name of the requested setup.
Business Area	Business area requesting the setup change.
Application	Application under the business area requesting the setup change.
Requested By	Required date of completion for the setup change.
Setup Description	Detailed description of the requested setup change.
Operating Unit	Operating unit to which setup changes are made.
Environment Name	Environment to which setup changes are made.
Analysis tab	
Technical Effort	Technical effort required for the setup change.
Est. Effort	Estimated effort in hours required for the setup change.
Est. Completion Date	Estimated completion date of the setup change.
Dependencies	Dependencies for the setup changes.
Project Team	Project team assigned.

OraApps Status Update Request

This section provides a description, configuration considerations, and field definitions for the OraApps Status Update Request request type.

Description

Many Oracle organizations use manual status reports for gathering information on project status, exceptions, and risks. However, the process can be inefficient due to the time and resources spent on updating and consolidating information, and following up by managers to make sure group members are completing their tasks. Reviewers may not know when status reports are ready for review, whom to contact, or how to access the most current versions of the reports.

The OraApps Status Update Request request type initiates the process of gathering and reviewing status information. It allows managers to request information, automatically notifies group members of requirements for information, and monitors task progress to support timely completion. When tasks for completing a report have been done, all reviewers are automatically notified. All involved parties have access to the latest version of the report.

This request type works with the OraApps Status Update Request workflow.

For More Information

For an overview of request processes, see [Overview of the Request Process on page 284](#).

For more information about the workflow associated with the OraApps Status Update Request request type, see [OraApps Status Update Request Workflow on page 368](#).

Configuration Considerations

The OraApps Setup Change Process workflow should be the default and the only allowed workflow assigned to the OraApps Setup Change Request request type.

Decide which subteams need to produce status reports, and do the following:

- Create new Status and Lead fields for the new teams that are not predefined.
- Delete the Status and Lead fields that are not relevant to your project.

Decide whether the default for each Status field should be **Requested** or **Not Requested**.

If you have already identified the team leads for the project, make them the default values in the Team Lead fields.

Field Definitions

Figure B-10 shows the screen for creating an OraApps Status Update Request based on the provided request type. *Table B-11* contains field definitions for the request type.

Create New OraApps Status Update Request

Expand All Collapse All Submit Cancel

Header

Summary

Details

Status Request

*Request Date:

*Distribution Status: Requested Distribution Team Lead:

*Finance Status: Requested Finance Team Lead:

*Mfg Status: Requested Mfg Team Lead:

*CRM Status: Requested CRM Team Lead:

*Conversions Status: Requested Conversions Team Lead:

*Interfaces Status: Requested Interfaces Team Lead:

*Reports Status: Requested Reports Team Lead:

*Testing Status: Requested Testing Team Lead:

Notes

References

Submit Cancel

Figure B-10. OraApps Status Update Request creation

Table B-11. OraApps Status Update Request request type field definitions

Field Name	Definition
Summary tab (not expanded in Figure B-10)	
Workflow	OraApps Status Update Request
Status Request tab	
Request Date	Due date for the status reports requested by the request. This is a required field but does not drive any specific notification or workflow logic.
Distribution Status	Whether or not the requestor is asking for a status report from the distribution team (Requested or Not Requested). Requested activates the distribution team branch in the business process workflow, prompting for and collecting the distribution team status report.
Finance Status	Whether or not the requestor is asking for a status report from the finance team (Requested or Not Requested). Requested activates the finance team branch in the business process workflow, prompting for and collecting the finance team status report.
Mfg Status	Whether or not the requestor is asking for a status report from the manufacturing team (Requested or Not Requested). Requested activates the manufacturing team branch in the business process workflow, prompting for and collecting the manufacturing team status report.
CRM Status	Whether or not the requestor is asking for a status report from the CRM team (Requested or Not Requested). Requested activates the CRM team branch in the business process workflow, prompting for and collecting the CRM team status report.
Conversions Status	Whether or not the requestor asking for a status report from the conversion team (Requested or Not Requested). Requested activates the conversion team branch in the business process workflow, prompting for and collecting the conversion team status report.
Interfaces Status	Whether or not the requestor is asking for a status report from the interfaces team (Requested or Not Requested). Requested activates the interfaces team branch in the business process workflow, prompting for and collecting the interfaces team status report.
Reports Status	Whether or not the requestor is asking for a status report from the reports team (Requested or Not Requested). Requested activates the reports team branch in the business process workflow, prompting for and collecting the reports team status report.
Testing Status	Whether or not the requestor is asking for a status report from the testing team (Requested or Not Requested). Requested activates the testing team branch in the business process workflow, prompting for and collecting the testing team status report.

Table B-11. OraApps Status Update Request request type field definitions [continued]

Field Name	Definition
Distribution Team Lead	User name of the person who will provide the distribution team report. Required if Distribution Status is set to Requested .
Finance Team Lead	User name of the person who will provide the finance team report. Required if Finance Status is set to Requested .
Mfg Team Lead	User name of the person who will provide the manufacturing team report. Required if Manufacturing Status is set to Requested .
CRM Team Lead	User name of the person who will provide the CRM team report. Required if CRM Status is set to Requested .
Conversions Team Lead	User name of the person who will provide the conversions team report. Required if Conversions Status is set to Requested .
Interfaces Team Lead	User name of the person who will provide the interfaces team report. Required if Interfaces Status is set to Requested .
Reports Team Lead	User name of the person who will provide the reports team report. Required if Reports Status is set to Requested .
Testing Team Lead	User name of the person who will provide the testing team report. Required if Testing Status is set to Requested .

Appendix C Workflows

In This Appendix:

- *Overview of Workflows*
- *Reference Workflows*
- *List of Workflows*
- *General Configuration Considerations*
- *OraApps Application Issue Workflow*
 - *Workflow Diagram and Step Descriptions*
 - *Workflow Steps with Predefined Security*
- *OraApps Cloning Process Workflow*
 - *Workflow Diagram and Step Descriptions*
- *OraApps Conversion Process Workflow*
 - *Workflow Diagram and Step Descriptions*
 - *Steps with Predefined Workflow Security*
- *OraApps Customization/Configuration Deployment Workflow*
 - *Configuration Considerations*
 - *Workflow Diagram and Step Descriptions*
 - *Steps with Predefined Workflow Security*
- *OraApps Design & Development Workflow*
 - *Workflow Diagram and Step Descriptions*
 - *Steps with Predefined Workflow Security*
- *OraApps Enhancement Process Workflow*
 - *Workflow Diagram and Step Descriptions*
 - *Steps with Predefined Workflow Security*
- *OraApps GAP Analysis Process Workflow*
 - *Workflow Diagram and Step Descriptions*
 - *Steps with Predefined Workflow Security*

- *OraApps Interfaces Process Workflow*
 - *Workflow Diagram and Step Descriptions*
 - *Steps with Predefined Workflow Security*
 - *OraApps Patch Deployment Workflow*
 - *Configuration Considerations*
 - *Workflow Diagram and Step Descriptions*
 - *Steps with Predefined Workflow Security*
 - *OraApps Reports Process Workflow*
 - *Workflow Diagram and Step Descriptions*
 - *Steps with Predefined Workflow Security*
 - *OraApps Setup Change Process Workflow*
 - *Workflow Diagram and Step Descriptions*
 - *OraApps Status Update Request Workflow*
 - *Workflow Diagram and Step Descriptions*
 - *Steps with Predefined Workflow Security*
 - *OraApps 11i Cloning Workflow*
 - *Workflow Diagram and Step Descriptions*
-

Overview of Workflows

This appendix provides reference information about the Oracle-specific workflows provided in the Extension. These workflows are listed and defined in *Table C-1*.

Some workflows are associated with object types and Mercury Change Management, and include package execution. Other workflows are associated with request types and Mercury Demand Management, and describe decision-making processes. For more information about Mercury Change Management and Mercury Demand Management, see the *Mercury Change Management: Configuring a Deployment System* document and the *Mercury Demand Management: Configuring a Request Resolution System* document.

In the descriptions of each workflow, the object type or request type “partner” is identified. For more information about how each workflow and object type or request type work together, see the “partner” description referenced in the sections titled “For More Information.”

Migration and compilation of entities using workflows are driven by commands included within the entities or workflow steps. For more information about commands in the Mercury IT Governance Center environment, see the *Commands, Tokens, and Validations Guide and Reference*.

Reference Workflows

Reference (REFERENCE) workflows cannot be edited, but you can copy and rename them and edit the copies to meet your needs. You can also use existing non-reference workflows as is or configure them further to meet your needs.

List of Workflows

Table C-1 lists the workflows included in the Extension. Each is described in subsequent sections.

Table C-1. Workflows included in the Extension

Workflow Name	Product Scope	Definition
OraApps Application Issue	Mercury Demand Management	Evaluates requests and routes them according to priority, keeping requestors informed of status and prompting for action as needed.
OraApps Cloning Process	Mercury Demand Management	Initiates and tracks a request to clone an Oracle E-Business Suite instance.
OraApps Conversion Process	Mercury Demand Management	Helps to automates conversion and importing of data from legacy systems or third-party software into Oracle systems.
OraApps Customization/ Configuration Deployment	Mercury Change Management	Helps to automate deployment and review of customizations and configurations from development to test to production.
OraApps Design & Development	Mercury Demand Management	Manages functional analysis, technical design, and reviews, prompting for action as needed.
OraApps Enhancement Process	Mercury Demand Management	Provides a controlled process for business and technical approval of proposed enhancements to Oracle E-Business Suite.
OraApps GAP Analysis Process	Mercury Demand Management	Provides a process to analyze and address gaps between existing company processes and those of Oracle E-Business Suite.
OraApps Interfaces Process	Mercury Demand Management	Provides a process to design and develop interfaces to import and export data between Oracle tables and other software programs.
OraApps Patch Deployment	Mercury Change Management	Provides a process to review and apply Oracle patches in "vanilla," development, test, and production environments.
OraApps Reports Process	Mercury Demand Management	Provides a process for designing and developing new or modified reports for an Oracle system.
OraApps Setup Change Process	Mercury Demand Management	Provides a process for requesting a new setup for an Oracle system.
OraApps Status Update Request	Mercury Demand Management	Provides a process for gathering, coordinating, and reviewing status reports.
OraApps 11i Cloning	Mercury Change Management	Helps to clone an instance of Oracle E-Business Suite by automating major steps in the process.

General Configuration Considerations

Consider the following before you use or configure the workflows provided in the Extension:

- You must enable workflows before they can be used.
- Review the following areas to determine if configurations are required to meet the needs of your organization:
 - The list of notification recipients and text
 - The list of security groups. (You should probably configure security groups for each workflow step.)
 - The granularity level of steps. (Ask whether additional steps are needed or whether a subworkflow is needed.)

OraApps Application Issue Workflow

During an Oracle implementation or upgrade, issues can arise quickly. Unless these issues are resolved efficiently, they can grow into a backlog that may stall the project. Issues can come from a number of different sources and be recorded in various locations. A process must exist that consolidates Oracle-related issues so project members can create and review issues systematically.

The OraApps Application Issue workflow initially evaluates requests according to their priority level. Requests designated as critical are routed to project managers. All other requests are routed to the initial review step. During these initial reviews and analyses, requestors are kept informed of status through notifications. Anyone needing to take action is prompted to do so.

If detailed analysis determines that there is a system issue, the issue is automatically triaged to either an external Oracle bug resolution process, or an internal justification, design, and development process.

The external process includes actions such as contacting Oracle, opening a TAR, and getting an Oracle patch. Visibility is maintained through each of these steps, with clear responsibility as to who needs to perform each action.

The internal process can include justification and signoff steps if the issue involves a change in scope, or merely following the normal design and development process for changes with limited impact. Both the internal and external processes eventually converge into the automated deployment process that uses Mercury Change Management, leveraging additional content included in the Extension.

This workflow works in conjunction with the OraApps Application Issue request type.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For general configuration considerations for workflows, see [General Configuration Considerations on page 323](#).

For more information about the request type associated with the OraApps Application Issue workflow, see [OraApps Application Issue Request Type on page 287](#).

Workflow Diagram and Step Descriptions

Figure C-1 illustrates the workflow.

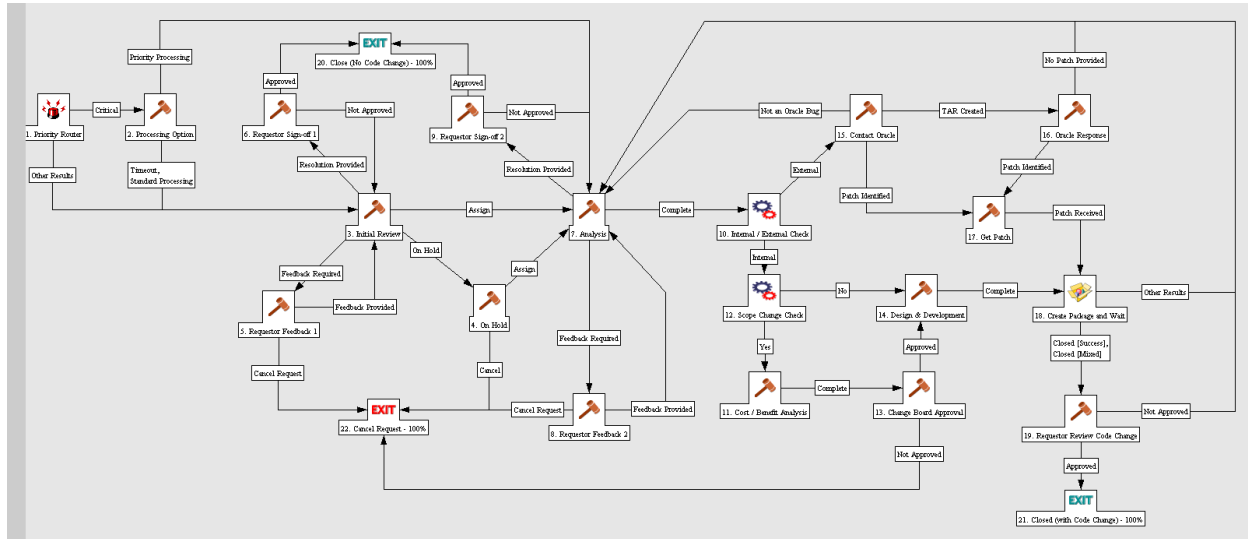


Figure C-1. OraApps Application Issue workflow

The workflow steps are described as follows:

- 1. Priority Router**—Once submitted, the priority router automatically routes the issue depending on the priority of the issue. Critical issues are routed so that the project managers can decide on the processing option. All other issues are routed to the Initial Review step.
- 2. Processing Option**—Project managers decide the processing option for critical issues. They can either decide on priority processing, in which case they assign a resource for analysis, or they can decide on the standard processing in which case the track owners make a decision on the issue at the Initial Review step.
- 3. Initial Review**—At the Initial Review step, the track owners review the issue. They may ask for more information from the requester or also provide a resolution to the requester. The track owners can also keep non-critical issues on hold until next development cycle using their judgment and available resource capacity.
- 4. On Hold**—Project managers and track owners can act on these issues and either cancel them or assign them for analysis.

5. Requestor Feedback 1—The requester can provide more information on issues where track owners request them.

6. Requestor Sign Off 1—The requester can accept the resolution provided or disapprove it.

7. Analysis—The business analyst performs a detailed review of the request and may make one more attempt at a resolution without a system change. If a resolution cannot be achieved, then it is assumed that a code or configuration change is required. The analyst determines if this change is within the current project scope and if it is external (a change to the standard Oracle Application code) or internal (a change to a customization or configuration) and moves the issue forward.

Once analysis is complete, the workflow automatically routes the request based on information gathered. If the issue requires an external change, the external vendor is contacted, otherwise the issue needs to go through internal approval and development.

8. Requestor Feedback 2—The requester can provide feedback to the business analyst assigned the issue.

9. Requestor Sign Off 2—The requester can accept the resolution provided by the business analyst analyzing the issue.

10. Internal/External Check—At this automatic step after analysis is complete, the request is routed depending on the analysis of whether the issue is external or internal. Internal issues are routed depending on whether it is a scope change.

11. Cost/Benefit Analysis—Changes that are out of the current project scope are analyzed for estimated effort as well the quantified business benefit. Once the justification information is gathered, the issue is passed on for approval. Usually, there is a specific group (here called the change board) that approves scope changes. This group can look at all issues currently pending review, look at the details on the request itself and make a decision on the change. If approved, the issue goes into design and development. Otherwise, it goes back to the analyst.

12. Scope Change Check—This step determines whether the changes are out of the project scope. If so, the workflow proceeds to the Cost/Benefit Analysis step (described above). If not, the workflow proceeds to the Change Board Approval step.

13. Change Board Approval—The change board can approve the change or reject it. Rejected changes is closed with failure. Approved changes move towards design and development.

14. Design & Development—This is a single step to indicate the full design and development process for the given change. At this step, a blocking request reference can be added to the issue, linking it to a request that models the full development process.

15. Contact Oracle—If the issue is external, then Oracle Support needs to be contacted. A new TAR may be created, or an existing patch may be located, or Oracle may indicate that the problem is not a bug. If a TAR is created, the TAR number is stored in the issue (in the TAR # field) and the issue waits for Oracle's response to the TAR. If an existing patch is found, the issue can move directly to the Get Patch step. If the problem is deemed not a bug, the issue returns to the analyst for further review.

16. Oracle Response—After the TAR has been created, Oracle provides a patch, commits to resolving the TAR in a future release, or indicates that the TAR will not be resolved. In the last two cases, no patch is provided, and the issue needs to go back to the analyst. At that point, the business analyst might determine that an internal workaround or change needs to be made or that the issue cannot be resolved.

17. Get Patch—If a patch is provided, it needs to be downloaded from Oracle Support and placed in the appropriate staging area.

18. Create Package and Wait—Once the development is done or once the Oracle patch is received, the change must go through the standard deployment cycle. For this, a new package is generated with either the Oracle Patch Deployment or Customization/Configuration Deployment workflow. The developer then adds the appropriate package lines and kicks off the deployment process.

19. Requestor Review Code Change—Once the package has completed the entire deployment process, the original requestor is notified and can approve the result (closing the issue) or set it to not approved (sending the issue back to the business analyst).

20. Close (No Code Change)—If the Requestor Sign Off 1 step or the Requestor Sign Off 2 step is an approval, the issues close with no code change.

21. Closed (with Code Change)—Approved issues close with success, whereas issues not approved get routed back for analysis.

22. Cancel Request—If the change board does not approve the request in the Change Board Approval step, the request gets cancelled.

Workflow Steps with Predefined Security

Table C-2 shows the steps of the workflow that have predefined security.

Table C-2. OraApps Application Issue workflow predefined security

Step	Step Name	Security Setting
2	Processing Option	OA - Management
3	Initial Review	OA - Track Owner
4	On Hold	OA - Management OA - Track Owner
7	Analysis	OA - Developer
11	Cost/Benefit Analysis	OA - Developer
13	Change Board Approval	OA - Management
14	Design & Development	OA - Developer
15	Contact Oracle	OA - Developer OA - Oracle Patch Admin
16	Oracle Response	OA - Developer OA - Oracle Patch Admin
17	Get Patch	OA - Developer OA - Oracle Patch Admin
18	Create Package and Wait	OA - Developer

OraApps Cloning Process Workflow

The OraApps Cloning Process workflow streamlines the process of initiating and tracking a request to clone an Oracle E-Business Suite instance.

This workflow works in conjunction with the OraApps Cloning Request request type.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For general configuration considerations for workflows, see [General Configuration Considerations on page 323](#).

For more information about the request type associated with the OraApps Cloning Process workflow, see [OraApps Cloning Request Request Type on page 291](#).

Workflow Diagram and Step Descriptions

Figure C-2 illustrates the workflow.

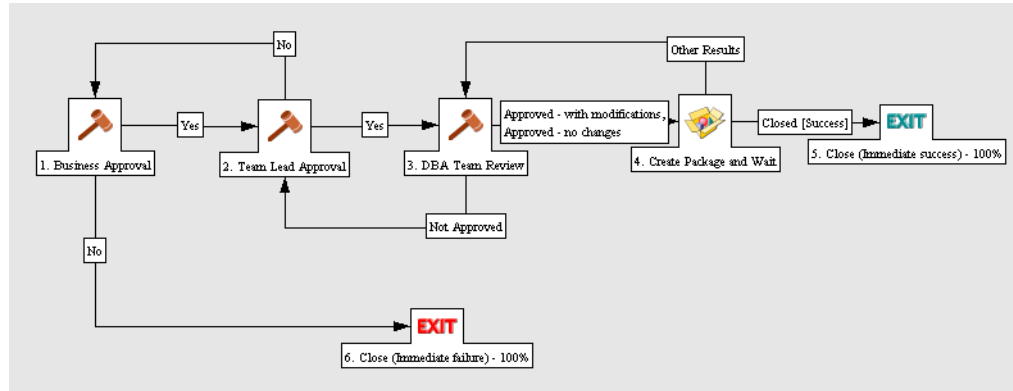


Figure C-2. OraApps Cloning Process workflow

The workflow steps are described as follows:

- 1. Business Approval**—Once submitted, the cloning request is reviewed by the business group.
- 2. Team Lead Approval**—The team lead reviews the request. If approved, the request is sent to the DBA team for review.
- 3. DBA Team Review**—At the DBA team analysis step, the request is analyzed for the technical feasibility, and if approved, a package is created for the environment cloning of the Oracle application.
- 4. Create Package and Wait**—The request waits at this step until the package completes the deployment.
- 5. Close (Immediate success)**—If the environment cloning is successful, the request closes out.
- 6. Close (Immediate failure)**—If the request does not obtain business approval in the Business Approval step, it is closed with a status of failure.

OraApps Conversion Process Workflow

Organizations using Oracle E-Business Suite may have used other legacy systems for similar functionality in the past. The data from other systems needs to be transformed and imported into Oracle using interfaces, and the correct tables should be populated with this data. For example, an organization that decides to use Oracle CRM needs to obtain all the contact information from legacy systems and put it into Oracle before they can start using Oracle for CRM.

Organizations using Oracle systems may have also used third-party software applications in the past. In order to be brought into Oracle, this data needs to go through conversion, as well. In both of these examples, data conversion is a one-time-only occurrence.

The OraApps Conversion Process workflow automates the conversion process.

This workflow works in conjunction with the OraApps Conversion Request request type.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For general configuration considerations for workflows, see [General Configuration Considerations on page 323](#).

For more information about the request type associated with the OraApps Conversion Process workflow, see [OraApps Conversion Request Request Type on page 294](#).

Workflow Diagram and Step Descriptions

Figure C-3 illustrates the workflow.

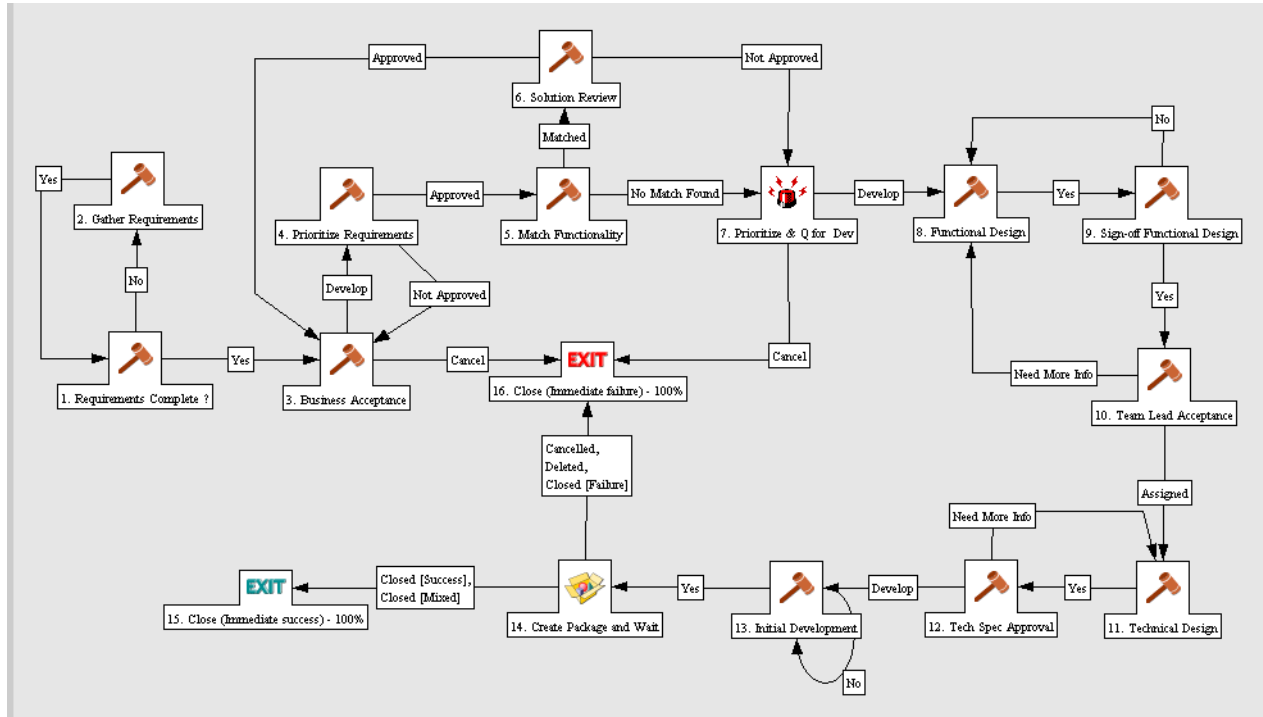


Figure C-3. OraApps Conversion Process workflow

The workflow steps are described as follows:

- 1. Requirements Complete?**—The OraApps Conversion Request is submitted. Once submitted, the requirements are checked for completeness.
- 2. Gather Requirements**—If requirements are unclear or more requirements are needed the requestor is notified for more information. After the requirements are signed off, the request moves to the Business Acceptance step.
- 3. Business Acceptance**—At this step, the business lead makes a decision to accept the request or may ask for more information on the requirements.
- 4. Prioritize Requirements**—If the requirements are accepted, they are prioritized based on business need and resource availability.
- 5. Match Functionality**—The prioritized requirements are matched with existing programs.

6. Solution Review—The solution review team analyzes and justifies the match. If a match is found the solution review team approves the request and notifies the business lead. If the suggested program does not match the requirements, the team disapproves the request and it moves to the Prioritize & Q for Dev step for prioritization.

7. Prioritize & Q for Dev—The track owner prioritizes the request for development work.

8. Functional Design—The functional designs are created.

9. Sign-Off Functional Design—The functional designs are reviewed and may be sent back if the reviewer needs more information on the designs. If approved, the request moves to the team lead for acceptance.

10. Team Lead Acceptance—The team lead may request more information on the functional designs that may lead to change in functional designs. The designs being modified need to be again signed off. When the team lead accepts the functional designs, a developer or team of developers is assigned.

11. Technical Design—The developer or team of developers creates technical designs and these designs are reviewed with the requirements.

12. Tech Spec Approval—During the technical approval, more information may be requested from the developer or the developer team that may require them to modify the technical designs.

13. Initial Development—On approval, the initial development begins.

14. Create Package and Wait—Until the deployment is complete, the request waits at the this step.

15. Close (Immediate success)—When the conversion program is complete, the request closes out and the requestor is notified.

16. Close (Immediate failure)—If the request does not obtain business acceptance in the Business Acceptance step, it is closed with a status of failure.

Steps with Predefined Workflow Security

Table C-3 shows the steps of the workflow that have predefined security.

Table C-3. OraApps Conversion Process workflow predefined security

Step	Step Name	Security Setting
3	Business Acceptance	OA - Business Lead
7	Prioritize & Q for Dev	OA - Track Owner
8	Functional Design	OA - Functional Designers
9	Sign-off Functional Design	OA - Functional Gap Reviewer
11	Technical Design	OA - Developer
12	Tech Spec Approval	OA - Technical Gap Reviewer

OraApps Customization/Configuration Deployment Workflow

It is crucial to the success of an Oracle E-Business Suite implementation, an upgrade, or ongoing production support that a process exist for managing configurations and customizations. After initial development, configurations and customizations need to be efficiently deployed through the various testing environments and finally to a production instance. This process can involve multiple testers and environment gatekeepers, so notifications for actions and for exceptions (such as failed tests and technical reviews) are important.

A key proven practice is to redeploy customizations first to the development environment before the formal unit test and before deployment to the QA environment. This accomplishes the following:

- It verifies that scripts can be rerun as they should, since they are being applied to an environment that should already have the changes.
- The developer should be the one initiating the automated deployment in the development environment. This individual can test the package for completeness and accuracy before sending it to the more controlled QA environment where someone else (a gatekeeper or operations person) initiates the deployment.
- There is better control of the changes being made to the development environment, giving visibility to the customizations and configurations in progress. This is very important when tracking down environment issues or when performing environment refreshes.

The OraApps Configuration/Customization Development workflow helps to automate this process.

The review steps incorporated towards the end of the deployment cycle are a key component of this workflow. For Oracle E-Business Suite, the smallest configuration or customization can have enormous performance impact on the production system. This is especially true with custom reports. All configurations and customizations should go through a performance review before being deployed into the production environment. Because of the higher impact of database changes, changes at the database level should go through additional DBA review.

The “horseshoe” shape of this workflow is based on the concept that if a non-recoverable failure ever occurs in a deployment step or review, the package should go back to the developer for rework. If the rework is successful, the package needs to go through the entire deployment process

again. It should not go back to the step where the original failure occurred. In this way, all environments get a consistent picture of the customization or configuration.

Configuration Considerations

If you want additional object types to be reviewed by your DBA team, add these as transition values from the Check for DB Objects step to the DBA Review & Approval step.

You may want to add additional SYNC steps so that the entire package moves in unison for every step in the process. The Mercury-supplied workflow limits this synchronization to the deployment steps.

If you have additional environments in your deployment cycle, add steps similar to the Deploy from DEV to TEST step and the SYNC (out of TEST) step.

Workflow Diagram and Step Descriptions

Figure C-4 illustrates the workflow.

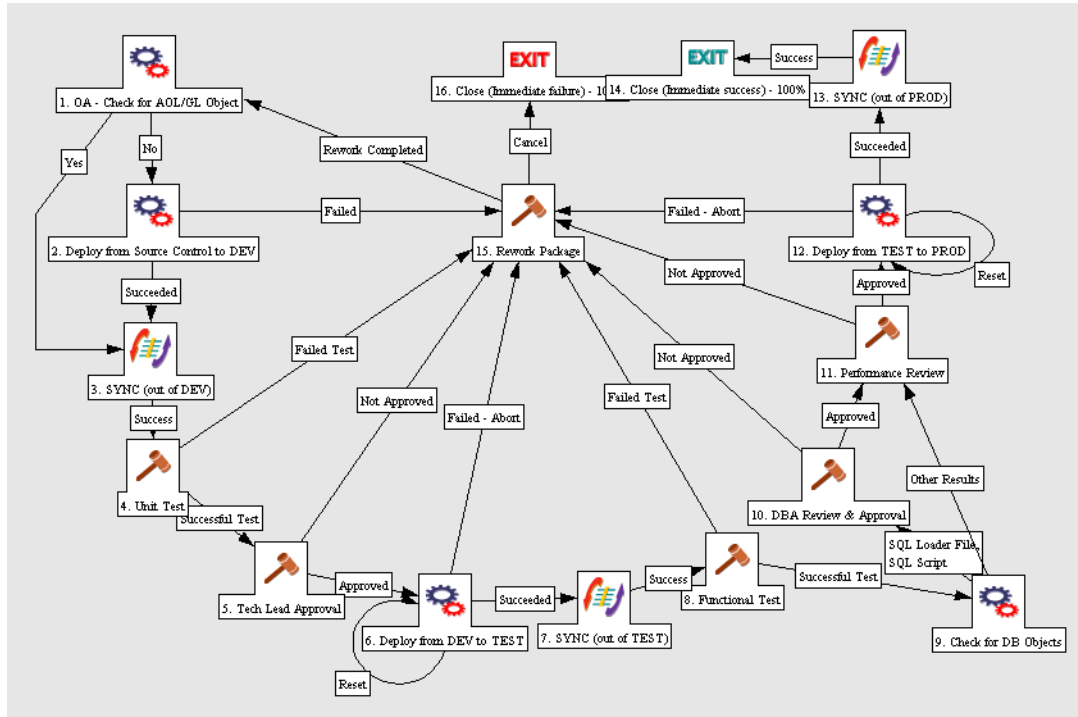


Figure C-4. OraApps Customization/Configuration Deployment workflow

The workflow steps are described as follows:

1. Check for AOL/GL Object—This is a automated token resolution step that checks to see if the object type on the package line is an entity that gets migrated by the Object Migrator or GL Migrator. If the object type is an AOL or GL object, the deployment to the DEV environment is bypassed, since migrating the configuration from DEV back to DEV does not add any value to the process. For other object types, this is done to make sure the scripts can be rerun and have been checked into version control.

2. Deploy from Source Control to DEV—This is an immediate execution step that deploys the package line into the DEV environment. The source environment should be set to the source control environment (or possibly the DEV environment) while the destination environment should be set to the development environment.

3. SYNC (out of DEV)—This step causes all package lines to wait until all lines have successfully deployed to DEV. The step makes sure that the package

moves through the various environments and deployment stages as a single entity and not as individual lines.

4. Unit Test—Once formally deployed to DEV, the Customization/Configuration is ready for unit testing. The assigned-to user for the package is notified of the deployment and prompted to perform the testing. Once the testing is done, the result is entered for this step. If testing was successful, the package line moves forward. If it failed, the line goes to the Rework Package step at the end of the workflow.

5. Tech Lead Approval—Once the package has passed unit test, it needs to be reviewed for adherence to technical standards and technical correctness. This is usually done by an architecture group or a technical team lead. The appropriate group is notified and should review the package. This might involve examining the actual code and scripts being deployed. Any review comments can be entered as notes on the package. The result of the review/approval is entered for this step. If approved, the line goes forward. Otherwise, it goes to the Rework Package step.

6. Deploy from DEV to TEST—This is an execution step that deploys the line from the DEV environment to the TEST environment. The developer or operations user needs to tell the system to perform the deployment. It does not automatically deploy as soon as the step becomes eligible. The source environment should be set to the development environment, and the destination environment should be set to the testing environment. If the deployment fails, the execution logs should be investigated. If the problem can be resolved without changing the original code (such as adding a tablespace to the database), the step should be reset and redeployed. If changes need to be made, the line moves to the Rework Package step.

7. SYNC (out of TEST)—This step causes package lines to wait until all lines have successfully deployed to TEST.

8. Functional Test—Once formally deployed to TEST, the Customization/Configuration is ready for QA testing. The QA team is notified of the deployment and prompted to perform the testing. Once the testing is done, the result is entered for this step. If testing was successful, the package line moves forward. If it failed, the line goes to the Rework Package step.

9. Check for DB Objects—This is an automatic token resolution that checks the object type for the package line. If it is a database-related object type, the line should be reviewed by a DBA, since there is a database impact. If it is not a database-related object type, the line skips the DBA Review & Approval step and goes to the Performance Review step.

10. DBA Review & Approval—For database-related object types, the DBA team is notified to review the appropriate package line. This can include reviewing the actual code being deployed. Any review comments can be entered as notes on the package. The result of the review/approval is entered for this step. If approved, the line goes forward. Otherwise, it goes to the Rework Package step.

11. Performance Review—Because of the significant performance impact changes can have to a production environment, every package should go through a review by the designated performance architecture team. Any review comments can be entered as notes on the package. The result of the review/approval is entered for this step. If approved, the line goes forward. Otherwise, it goes to the Rework Package step.

12. Deploy from TEST to PROD—This is an execution step that deploys the line from the TEST environment to the PROD environment. The operations user must tell the system to perform the deployment. It does not automatically deploy as soon as the step becomes eligible. The source environment should be set to the testing environment while the destination environment should be set to the production environment (or the “golden instance” if the system has not gone live yet.)

If the problem can be resolved without changing the original code (such as adding a tablespace to the database), the step should be reset and redeployed. If changes need to be made, the line should be moved to the Rework Package step.

13. SYNC (out of PROD)—This step causes all package lines to wait until all lines have successfully deployed to PROD.

14. Close (Immediate success)—Once all the lines have been successfully deployed to the PROD environment, the package is closed.

15. Rework Package—If there is any failure in deployment or review in the deployment workflow, the specific package line is sent back for rework. The original developer is notified of this action and should review the line to identify the problem. Once the problem is fixed, the line starts at the beginning of the deployment cycle, since the fix needs to be deployed through all the environments and be reviewed again. If a fix cannot be made, the line can be cancelled.

16. Close (Immediate failure)—If a package rework is cancelled at the Rework Package step, the package is closed with a status of failure.

Steps with Predefined Workflow Security

Table C-4 shows the steps of the workflow that have predefined security.

Table C-4. OraApps Customization/Configuration workflow predefined security

Step	Step Name	Security Setting
2	Deploy from Source Control to DEV	Assigned To Group Created By User Assigned To User
4	Unit Test	Assigned To Group Assigned To User Created By User
5	Tech Lead Approval	Assigned To Group Assigned To User
6	Deploy from DEV to TEST	Assigned To Group Assigned To User
8	Functional Test	Assigned To Group Assigned To User
10	DBA Review & Approval	Assigned To Group Assigned To User
11	Performance Review	Assigned To Group Assigned To User
12	Deploy from TEST to PROD	Assigned To Group Assigned To User
15	Rework Package	Assigned To Group Assigned To User Created By User

OraApps Design & Development Workflow

The design and development cycle requires a functional analysis and review followed by a technical design and review. After the functional and technical signoff, the core development is done, after which the configuration or customization is deployed through the standard deployment cycle, during which there is testing by other parties and possible rework.

During the process, it is important that individuals be proactively prompted to action and reminded if the actions are not performed in the specified period of time. The inefficiencies in the design and development process often occur in the approval and review process, where it might be unclear who needs to perform each action and when the actions are performed. The OraApps Design & Development workflow has multiple notifications at each step to do the following:

- Inform the appropriate individuals when they need to perform an action, such as reviewing a technical design.
- Remind the individuals of the pending actions until they actually perform the action.
- Let those involved in the process know the outcomes on each action, especially if the outcome was not expected (a disapproval or a cancellation).

The workflow also integrates with the deployment process by spawning the deployment package, maintaining visibility to the progress of the package through the deployment cycle, and waiting for the package to be fully deployed.

This workflow works with the OraApps Design & Development request type.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For general configuration considerations for workflows, see [General Configuration Considerations on page 323](#).

For more information about the request type associated with the OraApps Design & Development workflow, see [OraApps Design & Development Request Type on page 297](#).

Workflow Diagram and Step Descriptions

Figure C-5 illustrates the workflow.

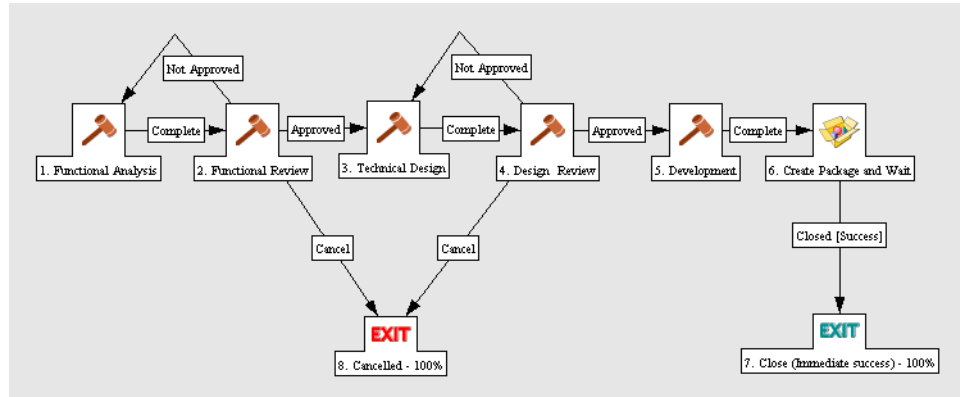


Figure C-5. OraApps Design & Development workflow

The workflow steps are described as follows:

1. Functional Analysis—At this step, a business analyst takes the requirement and creates a functional design to meet the requirement. This usually involves the creation of a functional design or high level design document, which should be attached to the request. The workflow supplied by Mercury assumes that the initial assigned-to user is the analyst and sends the notification for this step to that user.

2. Functional Review—Once the functional design document is complete, it needs to be reviewed. Usually, there is a group of individuals who review these documents. This step notifies all the members of this group, signified in the system as the OA - Functional Gap Reviewer Security Group. If the request does not pass the review, it notifies the creator of the request as well as the assigned-to user and sends the request back to the Functional Analysis step.

3. Technical Design and 4. Design Review—These steps are similar to the Functional Analysis and Functional Review steps but involve the detailed technical design and the developer and architect group instead of the analysts and functional review group.

5. Development—Once the design documents are complete and have been reviewed, the request goes into development. The assigned-to user now becomes the developer. Once initial development is done, this step is set to complete to move into the deployment process.

6. Create Package and Wait—At this step, a package is automatically created and referenced to the request. The developer should then update the package with the appropriate package lines for this unit. After the lines have been entered, the package should be released and travel through the appropriate deployment workflow. Once the package is complete, the request is automatically closed.

7. Close (Immediate success)—Once the package is complete, the request is automatically closed with a status of success.

8. Cancelled—If the Functional Review step or the Design Review step results in disapproval, the request is cancelled.

Steps with Predefined Workflow Security

Table C-5 shows the steps of the workflow that have predefined security.

Table C-5. OraApps Design & Development workflow predefined security

Step	Step Name	Security Setting
1	Functional Analysis	OA - Functional Designers Assigned To Group Assigned To User
2	Functional Review	OA - Functional Gap Reviewer
3	Technical Design	OA - Developer
4	Design Review	OA - Technical Gap Reviewer
5	Development	OA - Developer
6	Create Package and Wait	OA - Developer
8	Cancelled	OA - Functional Gap Reviewer OA - Technical Gap Reviewer

OraApps Enhancement Process Workflow

When using Oracle E-Business Suite, users may find that the system does not meet all of their business needs. This may be due to shortcomings within Oracle E-Business Suite, or to unusual circumstances or needs within the company. Modifications to existing functionality, or even entirely new functionality, may be required in order to meet the company's business needs.

Proposed enhancements must go through a process of business and technical approval to verify that they are truly necessary, that they will function well in conjunction with Oracle E-Business Suite, and that they will meet the business requirements. The OraApps Enhancement Process workflow implements the enhancement process, working in conjunction with the OraApps Enhancement Request request type.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For general configuration considerations for workflows, see [General Configuration Considerations on page 323](#).

For more information about the request type associated with the OraApps Enhancement Process workflow, see [OraApps Enhancement Request Request Type on page 300](#).

Workflow Diagram and Step Descriptions

Figure C-6 illustrates the workflow.

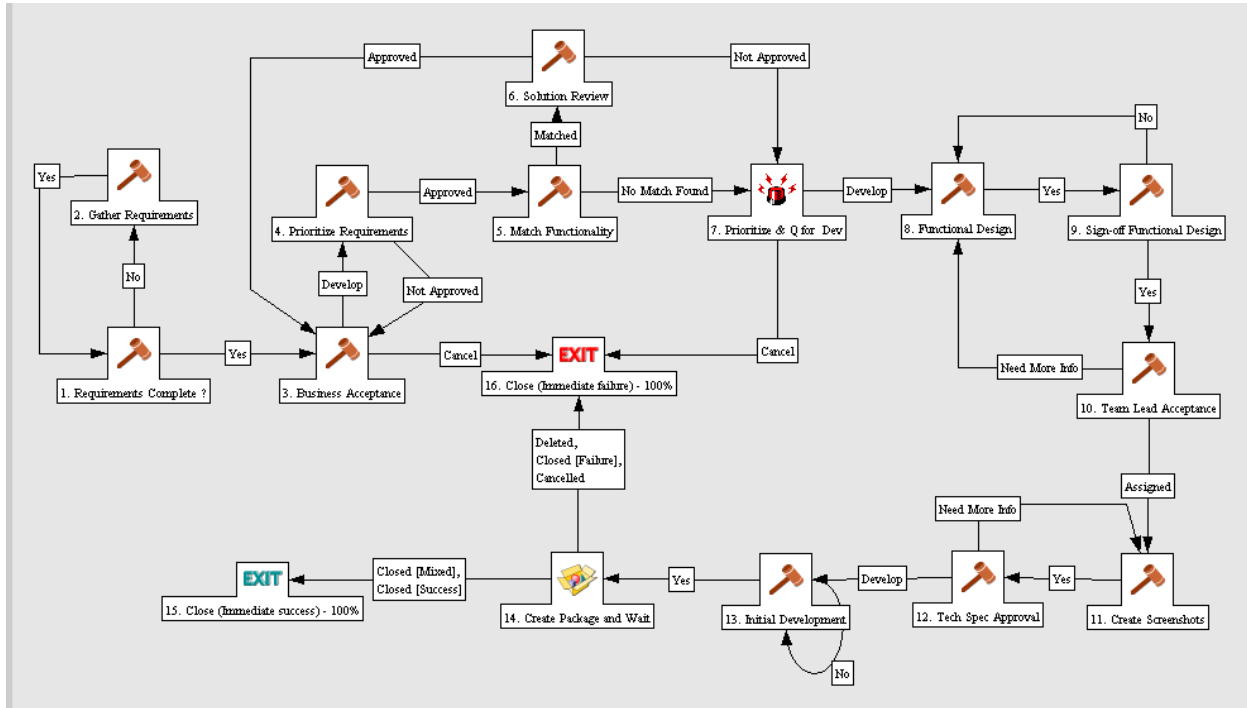


Figure C-6. OraApps Enhancement Process workflow

The workflow steps are described as follows:

- 1. Requirements Complete?**—The OraApps Conversion Request is submitted. Once submitted, the requirements are checked for completeness.
- 2. Gather Requirements**—If requirements are unclear or more requirements are needed, the requestor can be notified for more information. After the requirements are signed off, the request moves to the Business Acceptance step.
- 3. Business Acceptance**—At this step, the business lead makes a decision to accept the request, or may ask for more information about the requirements.
- 4. Prioritize Requirements**—If the requirements are accepted, they are prioritized based on business needs and resource availability.
- 5. Match Functionality**—The prioritized requirements are matched with existing functionality.

6. Solution Review—The solution review team analyzes cases where existing functionality was found to meet the business requirements. If the analysis shows that the functionality completely meets requirements, the solution review team approves the request and notifies the business lead. When the functionality does not match the requirements, the team disapproves the request and it moves to the Prioritize and Q for Dev step for prioritization.

7. Prioritize & Q for Dev—The track owner prioritizes the request for development work.

8. Functional Design—The functional design phase begins.

9. Sign-Off Functional Design—The functional designs are reviewed and may be sent back if the reviewer needs more information on the designs. If approved, the request moves to the team lead for acceptance.

10. Team Lead Acceptance—The team lead approves the functional design.-

11. Create Screenshots—When the team lead accepts the functional designs, a developer or team of developers is assigned. The developer or team of developers creates technical designs, including screenshots, mocks-ups, etc.

12. Tech Spec Approval—These designs are reviewed against the requirements, design standards, etc. During technical approval, more information may be requested from the developer or the developer team that may require them to modify the technical designs.

13. Initial Development—On approval, the initial development begins.

14. Create Package and Wait—Until the deployment is complete, the request waits at this step.

15. Close (Immediate success)—When the enhancement is complete, the request closes out and the requestor can be notified.

16. Close (Immediate failure)—If the request does not obtain business acceptance in the Business Acceptance step, if the track owner disapproves the request in the Prioritize & Q for Dev step, or if the request is cancelled at the Create Package and Wait step, the request is closed with a status of failure.

Steps with Predefined Workflow Security

Table C-6 shows the steps of the workflow that have predefined security.

Table C-6. OraApps Enhancement Process workflow predefined security

Step	Step Name	Security Setting
3	Business Acceptance	OA - Business Owner
4	Prioritize Requirements	OA - Business Owner
7	Prioritize & Q for Dev	OA - Track Owner
9	Sign-off Functional Design	OA - Functional Gap Reviewer
10	Team Lead Acceptance	OA - Track Owner
11	Create Screenshots	OA - Developer
12	Tech Spec Approval	OA - Technical Gap Reviewer

OraApps GAP Analysis Process Workflow

An implementation or upgrade of Oracle E-Business Suite can mean significant changes to the company's business processes. The difference between the existing processes and those provided by the packaged application is referred to as a "gap." Stakeholders in an Oracle-related project need to be able to request a "gap analysis" for specific Oracle functionality.

The OraApps GAP Analysis Process workflow implements the gap analysis process. This workflow operates in conjunction with the OraApps GAP Analysis request type.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For general configuration considerations for workflows, see [General Configuration Considerations on page 323](#).

For more information about the request type associated with the OraApps GAP Analysis Process workflow, see [OraApps GAP Analysis Request Request Type on page 303](#).

Workflow Diagram and Step Descriptions

Figure C-7 illustrates the workflow.

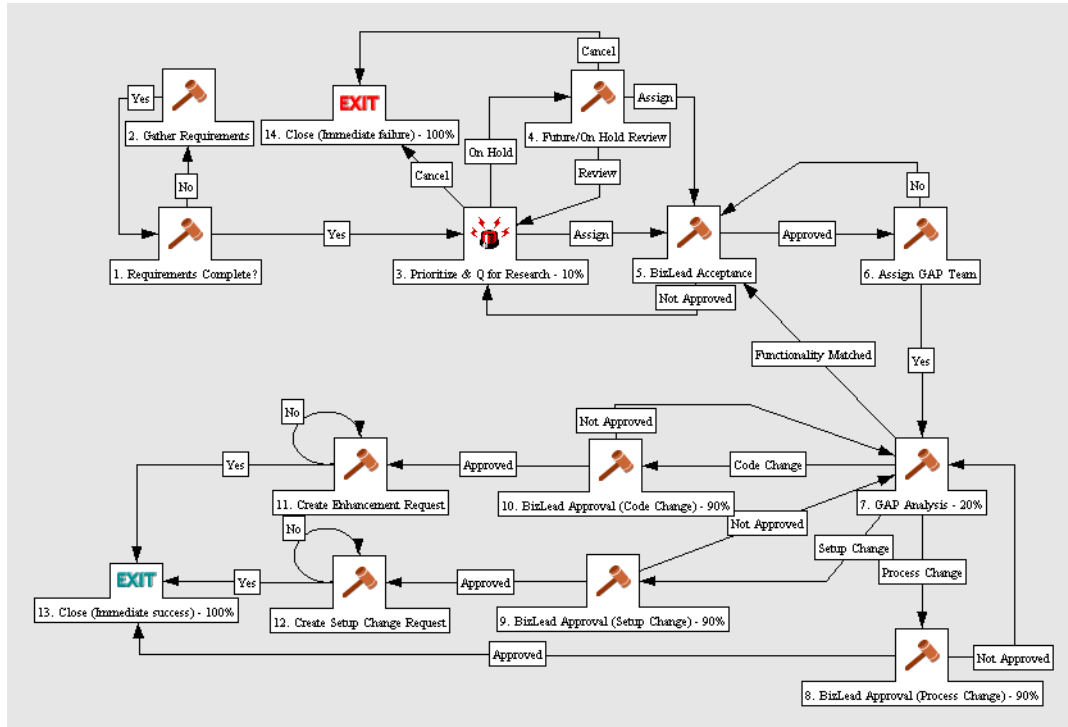


Figure C-7. OraApps GAP Analysis Process workflow

The workflow steps are described as follows:

1. Requirements Complete?—The OraApps GAP Analysis Request is submitted. Once submitted, the requirements are checked for completeness.

2. Gather Requirements—If requirements are unclear or more requirements are needed, the requestor is notified for more information. After the requirements are signed off, the request moves to the Prioritize & Q for Research step.

3. Prioritize & Q for Research—The manager prioritizes the request and assigns a business team lead to it.

4. Future/On Hold Review—The manager can also put the request on hold or cancel the request. If cancelled, the workflow moves to the Close (Immediate failure) step.

5. BizLead Acceptance—The business lead can accept the request and assign a team for gap analysis. Rejecting the request sends the request back to the manager.

6. Assign GAP Team—The business lead approves the request and assigns a team for gap analysis.

7. GAP Analysis—If the gap analysis team can find matching functionality in Oracle, they can send this information back to the business lead. The analysis team could also determine that a code change, a setup change, or process change is required for bridging the gap. The gap analysis team fills in all the details about the analysis and the time estimates for any changes that need to occur.

8. BizLead Approval (Process Change)—The business lead approves the creation of a process change if the gap analysis team determines that a process change is required. The gap analysis request closes as soon as the business lead approves the process change.

9. BizLead Approval (Setup Change)—The business lead approves the creation of a setup change request if the gap analysis team determines that a setup change is required.

10. BizLead Approval (Code Change)—The business lead approves the creation of a code change request if the gap analysis team determines that a code change is required.

11. Create Enhancement Request—An enhancement request is created.

12. Create Setup Change Request—The business lead approves the creation of a setup change request if the gap analysis team determines that a setup change is required. A setup change request is subsequently created. The gap analysis request closes after the setup change request is created.

13. Close (Immediate success)—The gap analysis request closes after the enhancement request is created.

14. Close (Immediate failure)—The request kept on hold by the manager during prioritizations can be later assigned to a business lead. The manager can also cancel the request, in which case it is closed with failure.

Steps with Predefined Workflow Security

Table C-7 shows the steps of the workflow that have predefined security.

Table C-7. OraApps GAP Analysis Process workflow predefined security

Step	Step Name	Security Setting
3	Prioritize & Q for Research	OA - Management
5	BizLead Acceptance	OA - Business Lead
6	Assign GAP Team	OA - Business Lead
8	BizLead Approval (Process Change)	OA - Business Lead
9	BizLead Approval (Setup Change)	OA - Business Lead
10	BizLead Approval (Code Change)	OA - Business Lead

OraApps Interfaces Process Workflow

An Oracle system may have to use data from other ERP systems as well as third-party software for different functionality. Interfaces must be created that allow extraction of data from other software programs to populate Oracle tables, and exporting of data from Oracle to other software programs.

The OraApps Interfaces Process workflow implements the interfaces process. This workflow works with the OraApps Interface request type.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For general configuration considerations for workflows, see [General Configuration Considerations on page 323](#).

For more information about the request type associated with the OraApps Interfaces Process workflow, see [OraApps Interface Request Request Type on page 306](#).

Workflow Diagram and Step Descriptions

Figure C-8 illustrates the workflow.

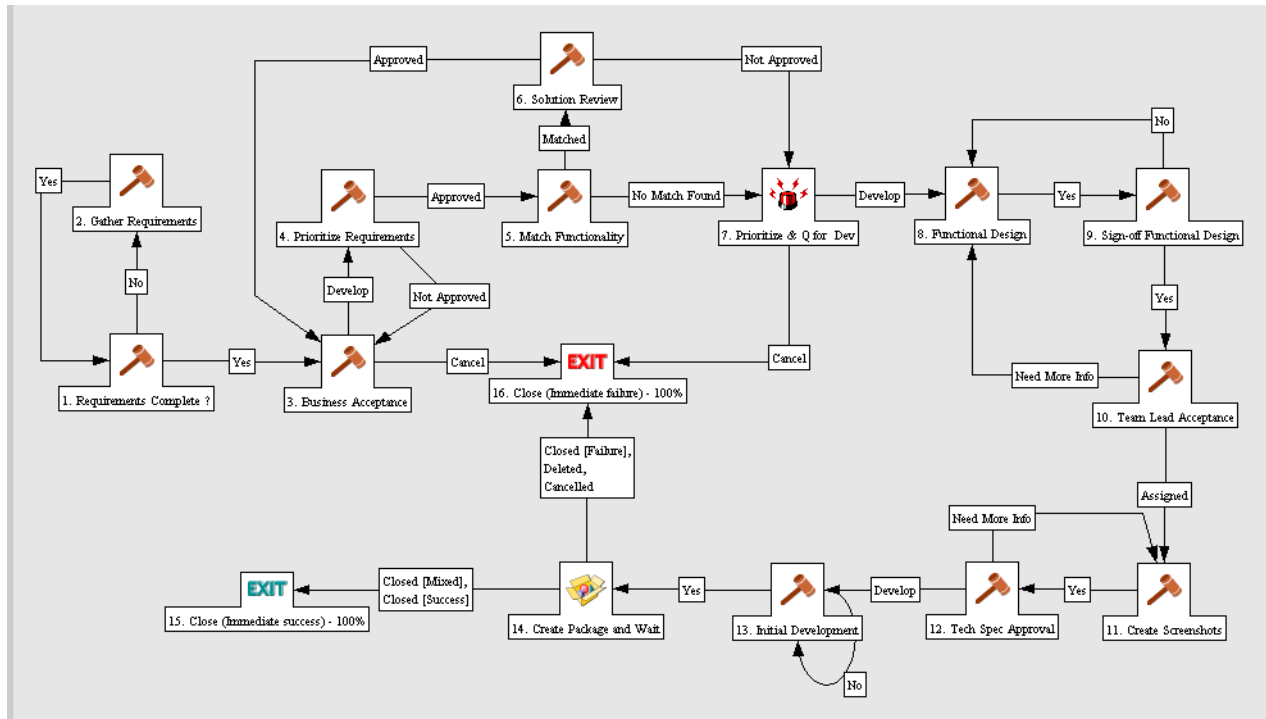


Figure C-8. OraApps Interfaces Process workflow

The workflow steps are described as follows:

- 1. Requirements Complete?**—The OraApps Interfaces Request is submitted. Once submitted, the requirements are checked for completeness.
- 2. Gather Requirements**—If requirements are unclear or more requirements are needed, the requestor is notified for more information. After the requirements are signed off, the request moves to the business acceptance step.
- 3. Business Acceptance**—At this step, the business lead makes a decision to accept the request, or may ask for more information on the requirements.
- 4. Prioritize Requirements**—If the requirements are accepted, they are prioritized according to business needs and resource availability.
- 5. Match Functionality**—The prioritized requirements are matched with the existing functionality.

6. Solution Review—The solution review team analyzes and justifies the need for a solution. If a match is found, the solution review team approves the request and notifies the business lead. When the functionality does not match the requirements, the team disapproves the request and it moves to the Prioritize and Q for Dev step for prioritization.

7. Prioritize & Q for Dev—The track owner prioritizes the request for development work.

8. Functional Design—The functional design phase begins.

9. Sign-off Functional Design—The functional designs are reviewed and may be sent back if the reviewer needs more information on the designs. If approved, the request moves to the team lead for acceptance.

10. Team Lead Acceptance—The team lead may request more information on the functional designs, which may lead to change in functional designs. Any modified designs need to be signed off again.

11. Create Screenshots—In this step, screenshots are taken.

12. Tech Spec Approval—When the team lead accepts the functional designs, a developer or team of developers is assigned. The developer or team of developers creates technical designs. These designs are reviewed with the requirements. During technical approval, more information may be requested from the developer or the developer team that may require them to modify the technical designs.

13. Initial Development—On approval, initial development begins.

14. Create Package and Wait—Until the deployment is complete, the request waits at this step.

15. Close (Immediate success)—When the interface is complete, the request closes out and the requestor is notified.

16. Close (Immediate failure)—If the request does not obtain business acceptance in the Business Acceptance step, if the track owner disapproves the request in the Prioritize & Q for Dev step, or if the request is cancelled at the Create Package and Wait step, the request is closed with a status of failure.

Steps with Predefined Workflow Security

Table C-8 shows the steps of the workflow that have predefined security.

Table C-8. OraApps Interfaces Process workflow predefined security

Step	Step Name	Security Setting
3	Business Acceptance	OA - Business Lead
4	Prioritize Requirements	OA - Business Lead
7	Prioritize & Q for Dev	OA - Track Owner
9	Sign-off Functional Design	OA - Functional Gap Reviewer
10	Team Lead Acceptance	OA - Track Owner
11	Create Screenshots	OA - Developer
12	Tech Spec Approval	OA - Technical Gap Reviewer

OraApps Patch Deployment Workflow

A significant number of patches are distributed for each release of Oracle E-Business Suite. The Extension provides automation to streamline and standardize the application of Oracle patches to all your environments. Because of the volume and impact of the patches, they must be monitored and managed. The environments to which they are applied must be monitored and managed as well.

The OraApps Patch Deployment workflow provides a process for viewing and controlling your Oracle patch activities.

Once an Oracle patch has been identified and received from Oracle, it should go through a structured process of review and deployment to maintain visibility to the patch and reduce the risk of applying a bad patch to a critical environment.

Because of the complex nature of Oracle patches, it is important that each patch be reviewed for impact, pre- and post-patch application steps, and general relevancy to your specific environment. After the review process, the patch needs to be applied first to a “vanilla” Oracle instance. This is an environment that does not include any in-house customization. This is a standard practice to isolate the Oracle patch in a clean environment and verify its correctness. If the patch works in this environment and not in the other instances, then the problem might be due to a customization or configuration. After the successful deployment to the vanilla instance, the Oracle patch proceeds through the development, testing, and production environments, getting verified in each instance.

The “horseshoe” shape of this workflow is based on the concept that if a non-recoverable failure ever occurs in a deployment or verification step, the Oracle patch should go back for review. This re-review determines the problem with the patch and, if correctable, moves the patch through the full deployment cycle. If the patch cannot be fixed, then the deployment is cancelled and Oracle Support should be contacted.

The OraApps Patch Deployment workflow calls the OraApps Patch Data Capture Subworkflow at several steps.

Configuration Considerations

Oracle Release 11i has a capability that allows you to take multiple patches and create a single patch that combines all the individual changes. This lets you reduce the number of post-application steps such as recompiling database objects or Oracle forms. If you plan to allow merging of individual Oracle patches, you must modify the workflow supplied by Mercury. This may involve significant changes in the process and the automation to account for the fact that a single deployment includes multiple individual patches. Please contact your Mercury IT Governance Center service provider for help in this area, if needed.

A proven practice for the Oracle Patch Application process is to put all patches in a central staging area and deploy patches to the different environments from this staging area. This establishes a single repository for patch files and drivers. Therefore, the source environment for each deployment step in the workflow should be set to the staging environment.

The “Apply to...” steps and the “Capture Patch...” steps should be set up in pairs, with the same destination environments specified.

Workflow Diagram and Step Descriptions

Figure C-9 illustrates the OraApps Patch Deployment workflow.

Figure C-10 illustrates the OraApps Patch Data Capture Subworkflow.

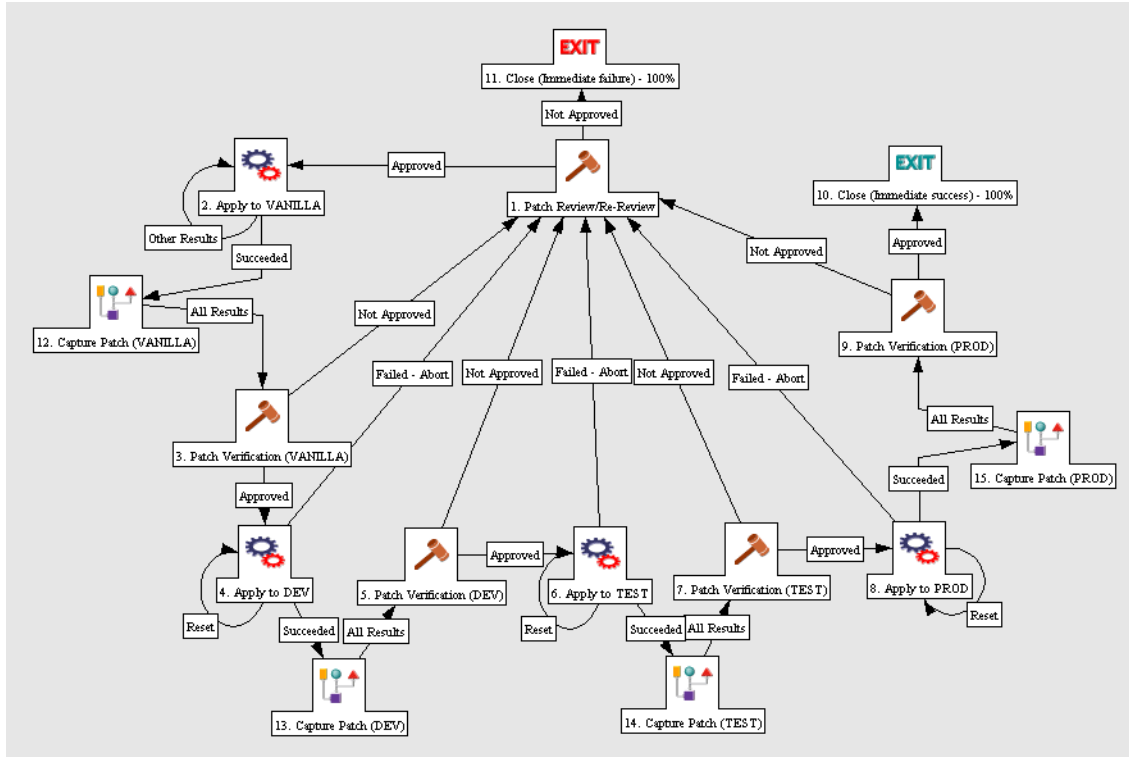


Figure C-9. OraApps Patch Deployment workflow

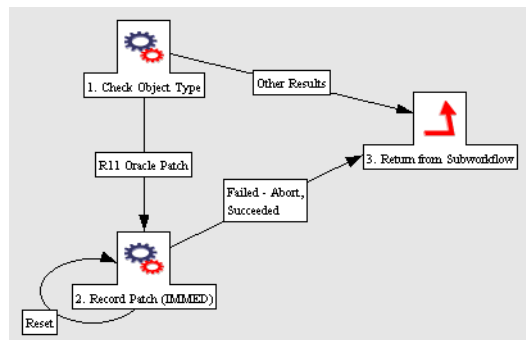


Figure C-10. OraApps Patch Data Capture Subworkflow

The workflow steps are described as follows:

1. Patch Review/Re-Review—Once the package is initially submitted, the Oracle patch administration group is notified to review the patch. This typically includes reviewing the `README` file along with the driver files. The patch review looks at the impact of the patch, the pre- and post-patch application steps, and general relevancy to the specific environment or problem. If the patch is approved, it moves to the Apply to VANILLA step.

This step also serves as the re-review step, in case the patch fails in deployment or verification during the deployment cycle. In this case, patch administrators should reexamine the patch to see if the problem can be fixed. For example, a patch application might fail because a prerequisite patch was not applied to the environment. If the problem can be fixed, the patch is moved to the start of the deployment cycle (where it is applied to the vanilla instance). Otherwise, the package line containing this patch is cancelled.

2. Apply to VANILLA—This is an execution step that applies the patch into the vanilla environment. The developer or operations user must tell the system to perform the deployment once any required pre-application steps are performed. It does not automatically deploy as soon as the step becomes eligible. The source environment should be set to the staging environment and the destination environment should be set to the vanilla environment.

If deployment fails, the execution logs should be investigated. These logs include links to the `ADPATCH` and `ADADMIN` logs, as well. If the problem can be resolved, the step should be reset and redeployed. If not, the patch should be sent back for re-review.

12. Capture Patch (VANILLA)—Once the patch has been applied to the VANILLA instance, details of the contents of the patch are captured to allow analysis and detail reporting, if the patch is eligible for this capture. This step calls the OraApps Patch Data Capture Subworkflow.

3. Patch Verification (VANILLA)—Once formally applied to VANILLA, the patch is ready for testing. The assigned-to user for the package is notified of the deployment and prompted to perform the testing. Once the testing is done, the result is entered for this step. If testing was successful, the package line moves forward. If it failed, the line goes to the Patch Review/Re-Review step.

4. Apply to DEV, 13. Capture Patch (DEV), and 5. Patch Verification (DEV)—These steps follow the same logic as the Apply to VANILLA, Capture Patch (VANILLA), and Patch Verification (VANILLA) steps above, including calling the OraApps Patch Data Capture Subworkflow, but they apply to the development environment. This is the first environment that includes customizations.

6. Apply to TEST, 14. Capture Patch (TEST), and 7. Patch Verification (TEST)—These steps follow the same logic as the Apply to DEV, Capture Patch (DEV), and Patch Verification (DEV) steps above, including calling the OraApps Patch Data Capture Subworkflow, but they apply to the test environment.

8. Apply to PROD, 15. Capture Patch (PROD), and 9. Patch Verification (PROD)—These steps follow the same logic as the Apply to TEST, Capture Patch (TEST), and Patch Verification (TEST) steps above, including calling the OraApps Patch Data Capture Subworkflow, but they apply to the production environment. This is the “live” production instance or the “golden” instance, if Oracle Applications have not been rolled out to production yet.

10. Close (Immediate success)—If the Patch Verification (PROD) step results in approval, the deployment is closed with a status of success.

11. Close (Immediate failure)—If the Patch Review/Re-Review step results in disapproval, the patch deployment is closed with status of failure.

Steps with Predefined Workflow Security

Table C-9 shows the steps of the workflow that have predefined security.

Table C-9. OraApps Patch Deployment workflow predefined security

Step	Step Name	Security Setting
1	Patch Review/Re-Review	OA - Oracle Patch Admin Assigned To Group Assigned To User Created By User
2	Apply to VANILLA	OA - Oracle Patch Admin Assigned To Group Assigned To User Created By User

Table C-9. OraApps Patch Deployment workflow predefined security

Step	Step Name	Security Setting
3	Patch Verification (VANILLA)	OA - Oracle Patch Admin Assigned To Group Assigned To User Created By User
4	Apply to DEV	OA - Oracle Patch Admin Assigned To Group Assigned To User Created By User
5	Patch Verification (DEV)	OA - Oracle Patch Admin Assigned To Group Assigned To User Created By User
6	Apply to TEST	OA - Oracle Patch Admin Assigned To Group Assigned To User Created By User
7	Patch Verification (TEST)	OA - Oracle Patch Admin Assigned To Group Assigned To User Created By User
8	Apply to PROD	OA - Oracle Patch Admin Assigned To Group Assigned To User Created By User
9	Patch Verification (PROD)	OA - Oracle Patch Admin Assigned To Group Assigned To User Created By User

OraApps Reports Process Workflow

Although Oracle E-Business Suite provides many reports, these standard reports are often insufficient to meet a company's business needs. For example, it may be important to include information captured in a Descriptive Flexfield, or to report on transactional data based on company-specific criteria. When a report does not currently exist that can provide the required information, a new report must be created. The OraApps Reports Process workflow implements the process of developing a new or modified report for an Oracle system. This workflow works with the OraApps Report Request request type.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For general configuration considerations for workflows, see [General Configuration Considerations on page 323](#).

For more information about the request type associated with the OraApps Reports Process workflow, see [OraApps Report Request Request Type on page 309](#).

Workflow Diagram and Step Descriptions

Figure C-11 illustrates the workflow.

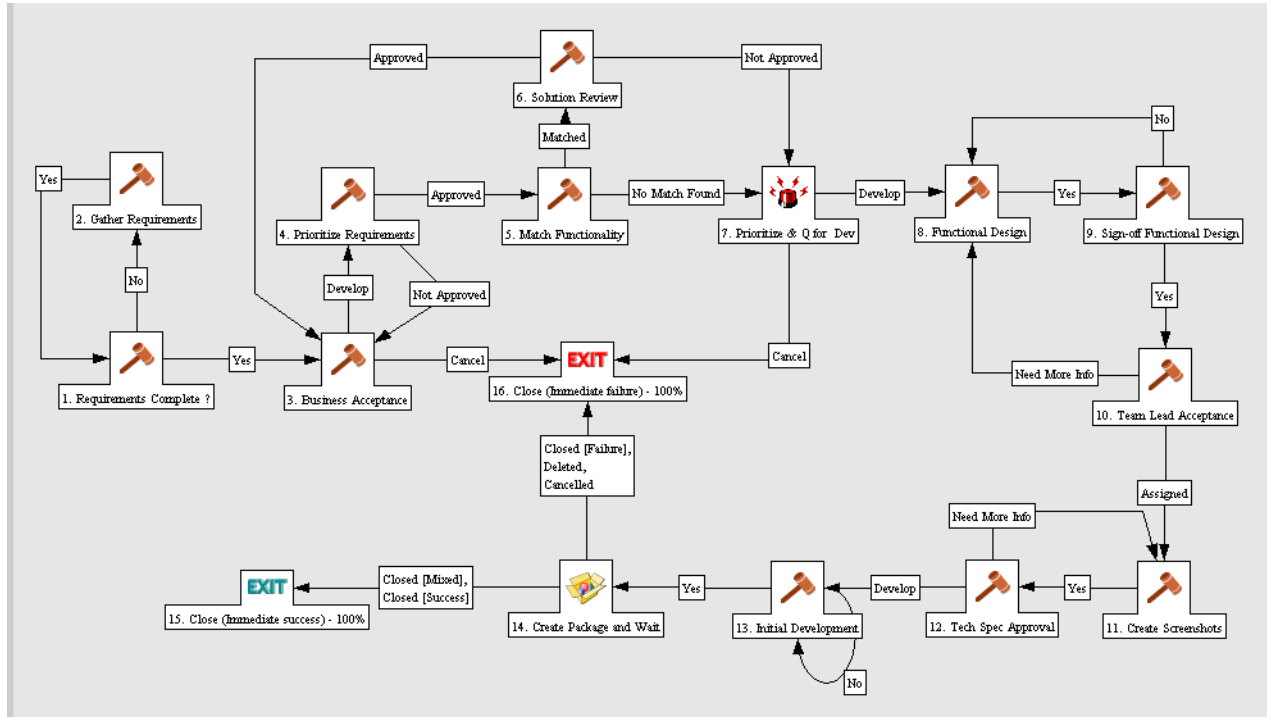


Figure C-11. OraApps Reports Process workflow

The workflow steps are described as follows:

- 1. Requirements Complete?**—The OraApps Reports Request is submitted. Once submitted, the requirements are checked for completeness.
- 2. Gather Requirements**—If requirements are unclear or more requirements are needed, the requestor is notified for more information. After the requirements are signed off, the request moves to the Business Acceptance step.
- 3. Business Acceptance**—At this step, the business lead makes a decision to accept the request, or may ask for more information about the requirements.
- 4. Prioritize Requirements**—If the requirements are accepted, they are prioritized according to business needs and resource availability.
- 5. Match Functionality**—The prioritized requirements are matched with existing functionality.

6. Solution Review—The solution review team analyzes and justifies a possible solution. If a match is found, the solution review team approves the request and notifies the business lead. When the functionality does not match the requirements, the team disapproves the request and it moves to the Prioritize and Q for Dev step for prioritization.

7. Prioritize & Q for Dev—The track owner prioritizes the request for development work.

8. Functional Design—The functional design phase begins.

9. Sign-off Functional Design—The functional designs are reviewed and may be sent back if the reviewer needs more information about the designs. If approved, the request moves to the team lead for acceptance.

10. Team Lead Acceptance—The team lead may request more information about the functional designs, which in turn may lead to changes in functional designs. The modified designs need to be signed off again.

11. Create Screenshots—In this step, screenshots are taken.

12. Tech Spec Approval—When the team lead accepts the functional designs, a developer or team of developers is assigned. The developer or team of developers creates technical designs and report layout mock-ups. These designs are reviewed with the requirements. During technical approval, more information may be requested from the developer or the developer team that may require them to modify the technical designs.

13. Initial Development—On approval, initial development begins.

14. Create Package and Wait—Until the deployment is complete, the request waits at this step.

15. Close (Immediate success)—When the report is complete, the request closes out and the requestor is notified.

16. Close (Immediate failure)—If the request does not obtain business acceptance in the Business Acceptance step, if the track owner disapproves the request in the Prioritize & Q for Dev step, or if the request is cancelled at the Create Package and Wait step, the request is closed with a status of failure.

Steps with Predefined Workflow Security

Table C-10 shows the steps of the workflow that have predefined security.

Table C-10. OraApps Reports Process workflow predefined security

Step	Step Name	Security Setting
3	Business Acceptance	OA - Business Lead
4	Prioritize Requirements	OA - Business Lead
7	Prioritize & Q for Dev	OA - Track Owner
9	Sign-off Functional Design	OA - Functional Gap Reviewer
10	Team Lead Acceptance	OA - Track Owner
11	Create Screenshots	OA - Developer
12	Tech Spec Approval	OA - Technical Gap Reviewer

OraApps Setup Change Process Workflow

Oracle systems use a number of different application setups to satisfy their business requirements. These application setups (referred to as “setup changes”) include modules such as Accounting Periods, Asset Categories, and Inventory Orgs. Each environment and operating unit has its own set of modules and needs its own setup changes.

The OraApps Setup Change Process workflow implements the process of requesting a new setup for an Oracle system. This workflow works with the OraApps Setup Change Request request type.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For general configuration considerations for workflows, see [General Configuration Considerations on page 323](#).

For more information about the request type associated with the OraApps Setup Change Process workflow, see [OraApps Setup Change Request Request Type on page 312](#).

Workflow Diagram and Step Descriptions

Figure C-12 illustrates the workflow.

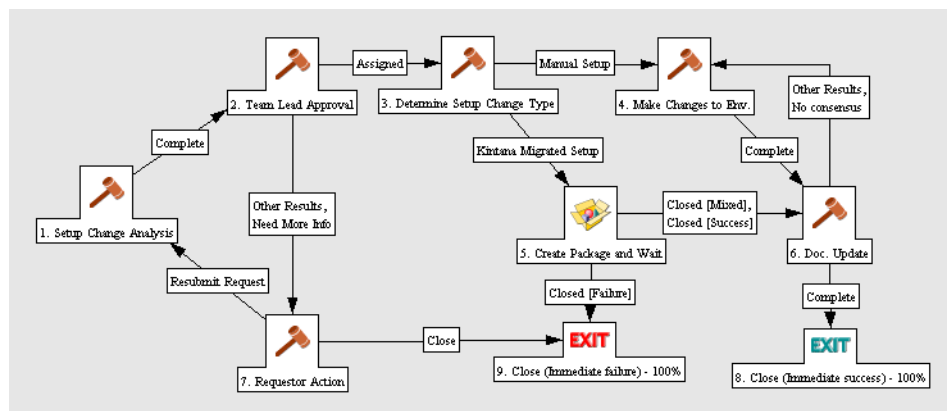


Figure C-12. OraApps Setup Change Process workflow

The workflow steps are described as follows:

- 1. Setup Change Analysis**—The functional lead analyzes the proposed change, determines its estimated effort and dependencies, if any, and determines the estimated completion date for the change.
- 2. Team Lead Approval**—The technical lead approves the setup change for the environment and assigns a resource.
- 3. Determine Setup Change Type**—The assigned developer determines if the request requires a manual setup change or a Mercury IT Governance Center migrated setup change. If it is a manual setup change, the request moves to the Make Changes to Env step. If it is a Mercury IT Governance Center migrated setup, the request moves to the Create Package and Wait step.
- 4. Make Changes to Env.**—The setup changes are implemented by the assigned developer.
- 5. Create Package and Wait**—For Mercury IT Governance Center migrated setups, a package is created by the developer, and it waits here until implemented.
- 6. Doc. Update**—The setup changes are documented by the developer.
- 7. Requestor Action**—If the team lead needs more information about the setup change or disapproves it, the change request goes to back to the requestor at this step. The requestor can resubmit the request or close it.
- 8. Close (Immediate success)**—After the setup changes are documented, the request closes with status of success.
- 9. Close (Immediate failure)**—If the requestor closes the request in the Requestor Action step or the package is closed with failure at the Create Package and Wait step, the change is closed with a status of failure.

OraApps Status Update Request Workflow

Many organizations use manual status reports for gathering information on project status, exceptions, and risks. However, the process can be inefficient due to the time and resources spent on updating and consolidating information, and following up by managers to make sure group members are completing their tasks. Reviewers may not know when status reports are ready for review, whom to contact, or how to access the most current versions of the manual reports.

The OraApps Status Update Request workflow implements the process of gathering and reviewing status reports. It allows managers to request information, automatically notifies group members of requirements for information, and monitors task progress to support timely completion. When tasks for completing a report have been done, all reviewers are automatically notified. All involved parties have access to the latest version of the report.

This workflow works with the OraApps Status Update Request request type.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For general configuration considerations for workflows, see [General Configuration Considerations on page 323](#).

For more information about the request type associated with the OraApps Status Update Request workflow, see [OraApps Status Update Request on page 315](#).

Workflow Diagram and Step Descriptions

Figure C-13 illustrates the workflow.

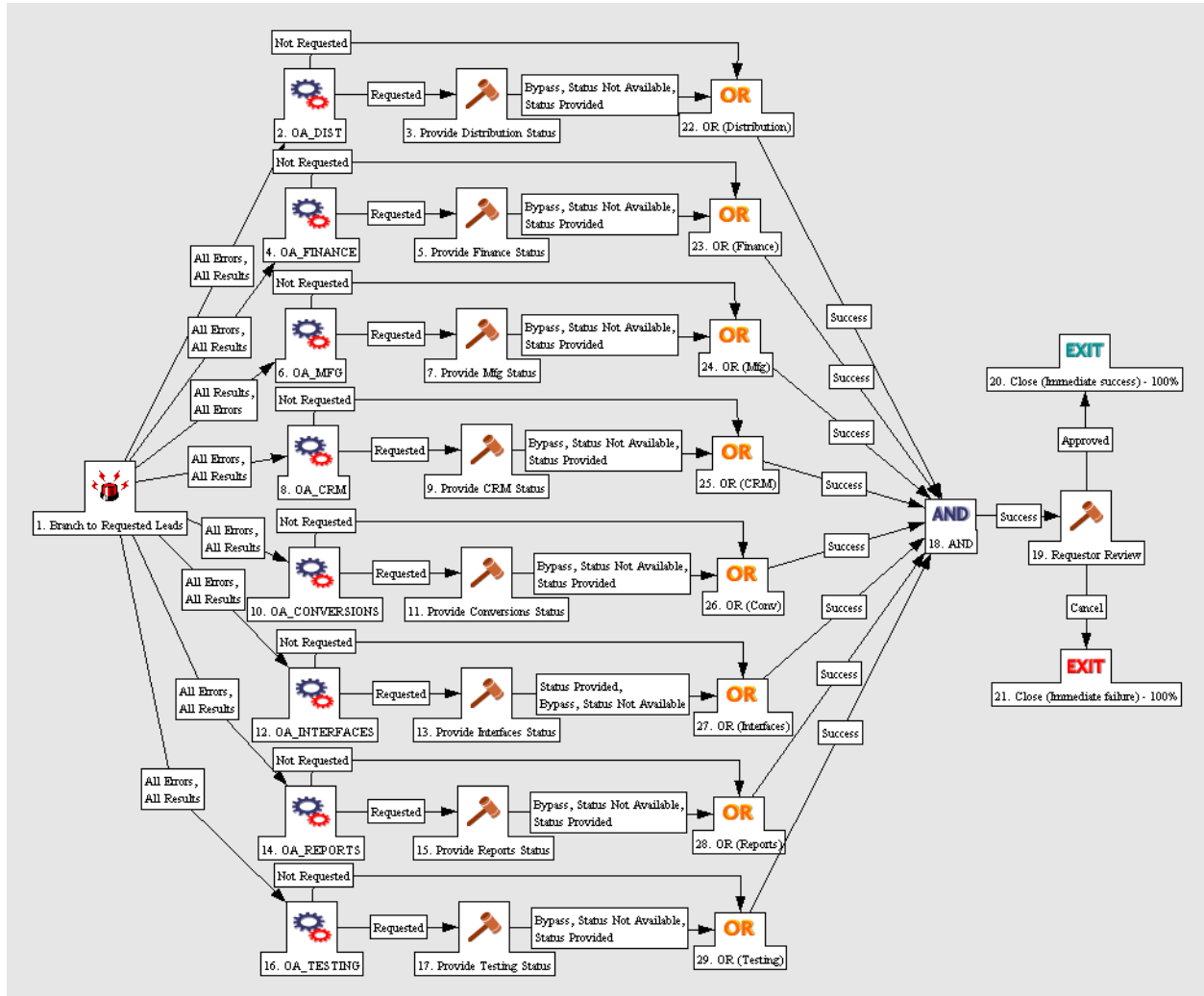


Figure C-13. OraApps Status Update Request workflow

The workflow steps are described as follows:

1. Branch to Requested Leads—This is an automatic execution step that starts each of the team branches in the workflow.

2. OA_DIST—This is an automatic token resolution step that looks at the value of the Distribution Status field (field token = OA_DIST). If the value was **Requested**, it moves the workflow to the next step. If the value was **Not Requested**, it skips to the end of the step. Note that the key to this token resolution is that the step name exactly matches the request field token name.

3. Provide Distribution Status—This step notifies the user that was specified in the Distribution Team Lead field to complete a status report and attach it to the request. This notification is set to send a reminder every two days until the task is done. (The amount of time is configurable.) The user attaches the report, when completed, and sets this step to Status Provided. The user or requestor can also chose to set this step to Bypass, Status Not Available in case no report is going to be provided and the workflow should move forward.

4.–17. Other branches—All the other branches in the workflow (such as Provide Finance Status, Provide Mfg Status and so on) follow the same logic as the Provide Distribution Status step.

19. Requestor Review—Once all the branches are complete (either with the completion of the Provide Status step or by skipping the branch because the report was not requested), the original creator of the request is notified that the reports are ready. The requestor should review the reports and either set this step to Approved or Cancel. Either result closes the request.

Steps with Predefined Workflow Security

Table C-11 shows the steps of the workflow that have predefined security.

Table C-11. OraApps Status Update Request workflow predefined security

Step	Step Name	Security Setting
2	OA_DIST	Assigned To Group Assigned To User Created By User
3	Provide Distribution Status	Assigned To Group Assigned To User Created By User [REQD.P.OA_DIST_LEAD]
4	OA_FINANCE	Assigned To Group Assigned To User Created By User
5	Provide Finance Status	Assigned To Group Assigned To User Created By User [REQD.P.OA_FINANCE_LEAD]
6	OA_MFG	Assigned To Group Assigned To User Created By User
7	Provide Mfg Status	Assigned To Group Assigned To User Created By User [REQD.P.OA_MFG_LEAD]
8	OA_CRM	Assigned To Group Assigned To User Created By User
9	Provide CRM Status	Assigned To Group Assigned To User Created By User [REQD.P.OA_CRM_LEAD]
10	OA_CONVERSIONS	Assigned To Group Assigned To User Created By User

Table C-11. OraApps Status Update Request workflow predefined security

Step	Step Name	Security Setting
11	Provide Conversions Status	Assigned To Group Assigned To User Created By User [REQD.P.OA_CONV_LEAD]
12	OA_INTERFACES	Assigned To User Created By User Assigned To Group
13	Provide Interfaces Status	Assigned To Group Assigned To User Created By User [REQD.P.OA_INTERFACES_LEAD]
14	OA_REPORTS	Assigned To Group Assigned To User Created By User
15	Provide Reports Status	Assigned To Group Assigned To User Created By User [REQD.P.OA_REPORTS_LEAD]
16	OA_TESTING	Assigned To Group Assigned To User Created By User
17	Provide Testing Status	Assigned To Group Assigned To User Created By User [REQD.P.OA_TESTING_LEAD]
19	Requestor Review	Assigned To Group Assigned To User Created By User

OraApps 11i Cloning Workflow

Using Oracle E-Business Suite requires management of a number of ongoing operating instances and temporary implementation or upgrade instances. These instances must be efficiently created, maintained, and periodically cloned without risking the stability of the system and disruption of day-to-day activities.

The OraApps 11i Cloning workflow works in conjunction with the OraApps 11i Cloning object type to clone an instance of Oracle E-Business Suite by automating the major steps in the process. The initial request for a clone usually comes from the OraApps Cloning Request request type and its workflow, the OraApps Cloning Process.

Information pertinent to the environments to be refreshed is captured in the object type. This information is to be used during execution of commands that clone the instance. Only the databases and directories that are supposed to be cloned are affected.

When the OraApps Cloning Process workflow reaches the Create Package and Wait step, the user creates a package that uses the OraApps 11i Cloning workflow and the OraApps 11i Cloning object type in its package lines.

The process implemented by this workflow and its associated object type follow the guidelines in the *Cloning Oracle Applications Release 11i* white paper published by Oracle.

The OraApps 11i Cloning workflow follows these general steps:

- Verifies the cloning requirements
- Copies the Mercury cloning scripts
- Runs the scripts in pre-clone mode to preserve the instance's configuration information
- Shuts the database down, deletes any database files that need to be deleted, and copies database files from the source backup to their destination
- Creates the control files for the destination database
- Starts the destination database
- Cleans up any concurrent requests copied from the source
- Copies various Oracle code files from the source backup directory to the destination
- Executes the cloning scripts in post-cloning mode

If the package was invoked from a request, the outcome status of the package would be returned to the request.



Note

Cloning your Oracle instance is a complex process, and misconfiguration can cause serious consequences. This object type and its associated workflow provide a template for cloning your instances, but require tailoring for your environment. Mercury recommends that you utilize Mercury Services to facilitate a successful implementation.

For More Information

For a description of request processes, see [Overview of the Request Process on page 284](#).

For general configuration considerations for workflows, see [General Configuration Considerations on page 323](#).

For more information about the request type associated with the OraApps 11i Cloning workflow, see [OraApps 11i Cloning Object Type on page 276](#).

For more information about the OraApps Cloning Process workflow, see [OraApps Cloning Process Workflow on page 329](#).

Workflow Diagram and Step Descriptions

Figure C-14 illustrates the workflow.

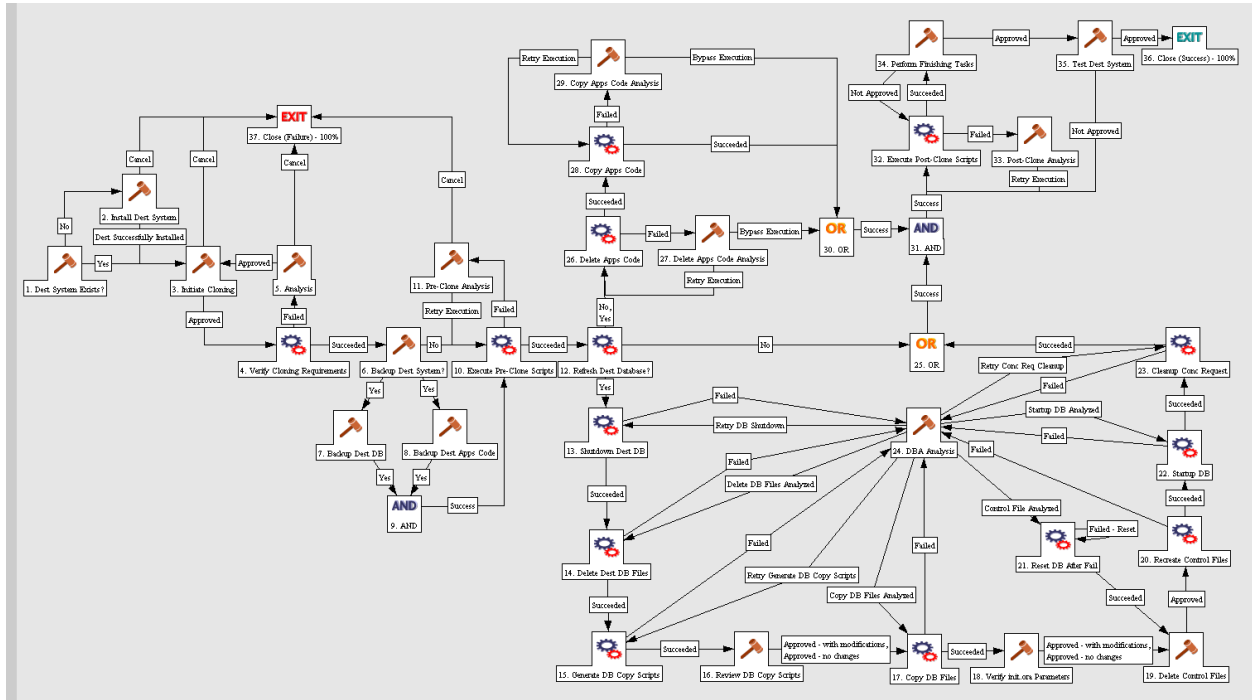


Figure C-14. OraApps 11i Cloning workflow

The workflow steps are described as follows:

1. Dest System Exists?—To clone an instance, the destination system must be installed. The user is asked to verify the existence of the destination system.

2. Install Dest System—If the destination system does not exist, the user must install it. After installation, the user goes to the Initiate Cloning step.

3. Initiate Cloning—If the destination system exists, the cloning process is initiated.

4. Verify Cloning Requirements—The requirements for cloning (such as having sufficient disk space, the existence of Perl, and the application of the cloning patch) are verified.

5. Analysis—If any of the cloning requirements are not met, an analysis of the problem is conducted. The problem must be resolved before the user can move on to the Backup Dest System? step.

6. Backup Dest System?—If all the cloning requirements are met, the user is prompted to back up the destination system.

7. Backup Dest DB and **8. Backup Dest Apps Code**—The user can choose whether to perform these backup steps. If the user chooses not to perform these steps, if the backup has already been performed, or after the backup completes, the workflow proceeds to the Execute Pre-Clone Scripts step.

10. Execute Pre-Clone Scripts—The pre-cloning scripts are executed.

11. Pre-Clone Analysis—If the Execute Pre-Clone Scripts step fails, a pre-clone analysis is performed. The cloning can be cancelled or a fix can be implemented for retry of the Execute Pre-Clone Scripts step.

12. Refresh Dest Database?—The user can choose to refresh the database and application code simultaneously, or only refresh the application code. During the database refresh, the following steps are executed:

- **13. Shutdown Dest DB**—The database is shut down.
- **14. Delete Dest DB Files**—The destination database files are deleted.
- **15. Generate DB Copy Scripts**—Copy scripts are generated to copy the database files from source to destination.

16. Review DB Copy Scripts—The user is prompted to review the generated copy scripts and make changes, if necessary.

17. Copy DB Files—The copy scripts are executed and the database files are copied to the destination.

Verification steps:

- **18. Verify init.ora Parameters**—Verify that the `init.ora` fields are correct.
- **19. Delete Control Files**—Decide whether to delete any control files indicated in the `init.ora` file.

After the verification, the following steps are executed:

- **20. Recreate Control Files**—The control files are recreated.
- **21. Reset DB After Full**—If the Recreate Control Files step, the Startup DB step, or the Cleanup Conc Request step fails, the DBA Analysis step is called. After the control file is analyzed, the database is reset in this step.
- **22. Startup DB**—The database is opened.
- **23. Cleanup Conc Request**—Any concurrent requests are cleaned up.

24. DBA Analysis—Any errors that require analysis and resolution by the DBA are identified.

26. Delete Apps Code—During the refresh of the application code, the destination files are deleted.

27. Delete Apps Code Analysis—If any errors are encountered during the application code refresh, analysis is requested, and the step is rerun.

28. Copy Apps Code—The source application code is copied to the destination.

29. Copy Apps Code Analysis—If any errors were found during the Copy Apps Code step, an analysis is performed and the step is rerun.

32. Execute Post-Clone Scripts—Once the refresh is completed, the post-clone scripts are executed.

33. Post-Clone Analysis—If any errors are found after the scripts are executed, an analysis is performed and the step is rerun.

34. Perform Finishing Tasks—Any necessary finishing tasks are performed.

35. Test Dest System—The destination system is tested.

36. Close (Success)—The workflow is closed with a status of success.

37. Close (Failure)—The workflow is closed with a status of failure.

Appendix D Report Types

In This Appendix:

- *Overview of Report Types*
- *Reference Report Types*
- *List of Report Types*
- *General Configuration Considerations*
- *OraApps Apps Issues Detail Report*
 - *Description*
 - *Configuration Considerations*
 - *Field Definitions*
- *OraApps Apps Issues Summary Report*
 - *Description*
 - *Configuration Considerations*
 - *Field Definitions*
- *OraApps Critical Requests Summary Report*
 - *Description*
 - *Configuration Considerations*
 - *Field Definitions*
- *OraApps IT Demand Summary Report*
 - *Description*
 - *Configuration Considerations*
 - *Field Definitions*
- *OraApps Ready Task List Report*
 - *Description*
 - *Configuration Consideration*
 - *Field Definitions*
- *OraApps Unassigned Ready Tasks Report*

- *Description*
 - *Configuration Consideration*
 - *Field Definitions*
 - *Patch-Related Reports*
 - *OraApps Patch Analysis Report*
 - *OraApps Patch Detail Report*
 - *Patch Application Comparison Report*
 - *Patches Applied to an Environment Report*
 - *Pending Patches Report*
-

Overview of Report Types

This appendix provides reference information about the Oracle-specific report types provided in the Extension. The report types are described in alphabetical order, except that patch-related reports are consolidated at the end of this appendix.

The content or output of a report is controlled by what you specify in the fields for the report type, as described in this appendix. Some fields allow multiple entries. Some fields are hidden by default, but can be enabled to allow further control of the report output.

Execution of reports is driven by commands included within the report types. For more information about commands in the Mercury IT Governance Center environment, see the *Commands, Tokens, and Validations Guide and Reference*.

Report types are configured in the Workbench (from the shortcut bar, click **Configuration** and click the **Report Types** icon). Reports are run (submitted) from the standard interface (**Reports** menu).



In addition to the report types provided by the Extension and described here, Mercury IT Governance Center includes the Compare Oracle Environments Report, which you can use to compare the data model of two Oracle schemas. For more information, see the *Reports Guide and Reference*.

For More Information

For information about running reports, see the *Reports Guide and Reference*.

Reference Report Types

Reference (REFERENCE) report types cannot be edited, but you can copy and rename them and edit the copies to meet your needs. You can also use existing non-reference report types as is or configure them further to meet your needs.

List of Report Types

Table D-1 lists and defines the report types included in the Extension. Each is described in subsequent sections.

Table D-1. Report types included in the Extension

Report Type Name	Definition
OraApps Apps Issues Detail	For the OraApps Application Issue request type, provides information similar to the Request Detail (Filter by Custom Fields) Report in Mercury Demand Management.
OraApps Apps Issues Summary	For the OraApps Application Issue request type, provides information similar to the Request Summary (Filter by Custom Fields) Report in Mercury Demand Management.
OraApps Critical Requests Summary	Lists a summary of Extension-related requests with Critical priority, similar to the Request Quick View Report in Mercury Demand Management
OraApps IT Demand Summary	Lists a summary of selected Extension-related requests, similar to the Request Summary Report in Mercury Demand Management.
OraApps Ready Task List	Lists Oracle project tasks that are in the Ready state, similar to the Project Task Assignment Report in Mercury Project Management.
OraApps Unassigned Ready Tasks	Lists Oracle project tasks that are in the Ready state but are not yet assigned.

Table D-1. Report types included in the Extension [continued]

Report Type Name	Definition
Patch-Related Reports	
OraApps Patch Analysis	Lists details of selected patches. Similar to the OraApps Patch Detail Report but does not allow loading of patch data.
OraApps Patch Detail	Lists details of selected patches or loads patch data for unapplied patches.
Patch Application Comparison	Lists Oracle patches migrated to or applied to up to six different environments, similar to the Environment Comparison by Objects Migrated Report in Mercury Change Management.
Patches Applied to an Environment	Lists Oracle patches migrated to or applied to a specific environment, similar to the Environment/Objects Detail Report in Mercury Change Management.
Pending Patches	Lists package lines for Oracle patches that are waiting at a migration step. This functionality is an Oracle-specific subset of the Packages Pending Report in Mercury Change Management.

General Configuration Considerations

Consider the following for all report types:

- Report types need to be enabled before they can be used.
- Any security restrictions required for the reports need to be configured. By default, the reports are available to all users.

OraApps Apps Issues Detail Report

This section provides information about the OraApps Apps Issues Detail Report.

Description

This report is similar to the Request Detail (Filter by Custom Fields) report in Mercury Demand Management, but it is specialized for the OraApps Application Issue request type. You can filter by the standard header fields as well as some of the specific detail fields, for example, TAR #, Business Benefit, and Environment.

Configuration Considerations

Consider the following in configuring this report:

- If you have removed detail fields in the OraApps - Application Issue request type, you should disable any fields that refer to those deleted fields.
- If you change the name of the request type to something other than OraApps Application Issue, you need to do the following:
 - Change the default for the Request Type field.
 - Create a copy of the OA - Apps Issue Reporting Request Types validation and make the appropriate changes to the values within the copy.
 - Create a copy of the OA - Apps Issue Request Types validation and modify it to point to the lookup type used by the new version of the OA - Apps Issue Reporting Request Types validation you created in the bullet above.
 - Change the Request Type field to use the new version of the OA - Apps Issue Request Types validation you created in the bullet above.

Field Definitions

Figure D-1 shows the Submit Report window you use to specify which requests you want to appear in the OraApps Apps Issues Detail Report and which information you want to appear for each request. *Table D-2* provides

definitions for the displayed fields and for some fields that are hidden by default.

Figure D-1. OraApps Apps Issues Detail Report

Table D-2. OraApps Apps Issues Detail Report field definitions

Field Name (*Required)	Definition	Hidden by Default?
Request Numbers	Request numbers to include in the report.	No
*Include Closed Requests	Whether or not to include requests that have been closed or cancelled.	No
*Request Type	Requests having this request type.	No
Status	Requests having this status.	No
Business Priority	Requests having this business priority,	No
Assigned To	Requests assigned to this user.	No
Assigned To Group	Requests assigned to this group.	No
Created By	Requests created by this user.	No
Request Sub Type	Requests having this sub-type.	No
Department	Requests logged against this department.	No
Workflow	Requests that are associated with this workflow.	No
Request Group	Requests using request group(s) Customization, Upgrade, or Setup .	No
Contact	Requests having this contact name.	No
Company Name	Requests having this company name.	No
Creation Date From	Requests created on this date or later.	No
Creation Date To	Requests created on this date or earlier.	No
Last Update Date From	Requests last updated on this date or later.	No
Last Update Date To	Requests last updated on this date or earlier.	No
Problem Contains	Requests having this text string.	No
Internal/External	Requests with this value of Internal or External.	No
Business Area Affected	Requests for this business area.	No
TAR #	Requests with this TAR number.	No
Source	Requests with this source.	No
Environment	Requests associated with this Oracle environment.	No
Business Benefit	Requests with this business benefit from resolving a particular issue.	No

Table D-2. OraApps Apps Issues Detail Report field definitions [continued]

Field Name (*Required)	Definition	Hidden by Default?
Required within next X days	Requests required within a particular number of days.	No
Est. Comp. within next X days	Requests with an estimated completion date within a particular number of days.	No
Field Prompt1	Custom field to use as a filter.	No
Field Value1	Value for Field Prompt1.	No
Field Prompt2	Custom field to use as a filter.	No
Field Value2	Value for Field Prompt2.	No
Report Title	Title of the report.	No
*Order By	Criterion to use to sort the requests in the report: Application, Assigned To, Created By, Creation Date, Department, Last Update Date, Priority, Request Group, Request Number, Request Sub Type, Request Type or Status.	No
*Show Header Fields	Whether or not the report will show the full header fields of each request.	No
*Show Detail Fields	Whether or not the report will show the detail fields of each request.	No
*Hide Prompts for Empty Fields	Whether or not the report will hide prompts that have empty fields.	No
*Show Contents of Table Fields	Whether or not the report will show table fields for requests that have them.	No
*Show Field Audit History	Whether or not the report will show the transaction history of each request.	No
*Show Notes	Whether or not the report will show the notes attached to each request.	No
*Filter Notes	Whether the report will show all notes or only user notes.	No
*Show Status	Whether or not the report will show the workflow steps and current step status for each request.	No
*Show References	Whether or not the report will show the references associated with each request.	No

OraApps Apps Issues Summary Report

This section provides information about the OraApps Apps Issues Summary Report.

Description

This report is similar to the Request Summary (Filter by Custom Fields) Report in Mercury Demand Management, but it is specialized for the OraApps Application Issue request type. You can filter by both the standard header fields and some of the specific detail fields, for example, TAR #, Business Benefit, and Environment. You can also summarize by many of the detailed data, using the dynamic Group By field.

Configuration Considerations

Consider the following in configuring this report:

- If you have removed detailed fields in the OraApps Application Issue request type, disable any fields referring to the deleted fields.
- If you change the name of the request type to something other than OraApps Application Issue, you need to do the following:
 - Change the default for the Request Type field.
 - Create a copy of the OA - Apps Issue Reporting Request Types validation and make the appropriate changes to the values within the copy.
 - Create a copy of the OA - Apps Issue Request Types validation and modify it to point to the lookup type used by the new version of the OA - Apps Issue Reporting Request Types validation you created in the bullet above.
 - Change the Request Type field to use the new version of the OA - Apps Issue Request Types validation you created in the bullet above.
- You can add grouping columns to OraApps Apps Issues Summary Report, but if detail fields are used, they must be part of the first field batch.

Field Definitions

Figure D-2 shows the Submit Report window you use to specify which requests you want to appear in the OraApps Apps Issues Summary Report and which information you want to appear for each request. Table D-3 provides definitions for the displayed fields and for some fields that are hidden by default.

The screenshot shows the 'Submit Report: OraApps Apps Issues Summary Report' window. At the top, there are 'Submit' and 'Cancel' buttons. Below is a 'Report Parameters' section with a 'Restore Default' button. The form is organized into two columns of fields:

- Left Column:** Request Numbers, Request Type (dropdown: OraApps Application Issue), Status, Assigned To, Created By, Department, Workflow, Contact, Creation Date From, Last Update Date From, Description Contains, Internal/External (dropdown: Internal), TAR #, Environment, Required within next X days, Field Prompt 1, Field Prompt 2, Report Title (text: OraApps Apps Issues Summary Rep), Group By.
- Right Column:** Include Closed Requests (radio: Yes selected, No), Priority, Assigned To Group, Request Sub Type, Request Group, Creation Date To, Last Update Date To, Business Area Affected (dropdown), Source (dropdown), Business Benefit (dropdown), Est. Comp. within next X days, Field Value 1, Field Value 2.

At the bottom, there are sections for 'Scheduling' and 'Advanced Notifications', followed by 'Submit' and 'Cancel' buttons, and a 'Close Window' button in the footer.

Figure D-2. OraApps Apps Issues Summary Report

Table D-3. OraApps Apps Issues Summary Report field definitions

Field Name (*Required)	Definition	Hidden by Default?
Request Numbers	Request numbers to include in the report.	No
*Include Closed Requests	Whether or not to include requests that have been closed or cancelled.	No
*Request Type	Request type on which the report can be run.	No
Status	Requests having this status.	No
Priority	Requests having this business priority.	No
Assigned To	Requests assigned to this user.	No
Assigned To Group	Requests assigned to this group.	No
Created By	Requests created by this user.	No
Request Sub Type	Requests having this sub-type.	No
Department	Requests logged against this department.	No
Workflow	Requests that are associated with this workflow.	No
Request Group	Requests using request group(s) Customization, Upgrade, or Setup .	No
Contact	Requests having this contact name.	No
Creation Date From	Requests created on this date or later.	No
Creation Date To	Requests created on this date or earlier.	No
Last Update Date From	Requests last updated on this date or later.	No
Last Update Date To	Requests last updated on this date or earlier.	No
Description Contains	Requests having this description text string (not a case sensitive search).	No
Internal/External	Requests with this value of Internal or External.	No
Business Area Affected	Requests for this business area.	No
TAR #	Requests with this TAR number.	No
Source	Requests with this source.	No
Environment	Requests associated with this Oracle environment.	No
Business Benefit	Requests with this business benefit from resolving a particular issue.	No

Table D-3. OraApps Apps Issues Summary Report field definitions [continued]

Field Name (*Required)	Definition	Hidden by Default?
Required within next X days	Requests required within a particular number of days.	No
Est. Comp. within next X days	Requests with an estimated completion date within a particular number of days.	No
Field Prompt1	Custom field to use as a filter.	No
Field Value1	Value for Field Prompt1.	No
Field Prompt2	Custom field to use as a filter.	No
Field Value2	Value for Field Prompt2.	No
Report Title	Title of the report.	No
*Group By	How to group the report results by a particular field or fields from the request.	No
*Include Subtotals for First Group Column	Whether or not the report will show subtotals for the first column specified in the Group by field.	No

OraApps Critical Requests Summary Report

This section provides information about the OraApps Critical Requests Summary Report.

Description

The OraApps Critical Requests Summary Report lists a summary of Extension-related requests with Critical priority. This report is similar to the Request Quick View Report in Mercury Demand Management. It displays summary information about the critical requests opened and closed in the current week as well as the numbers that are open by request type. In addition, the report lists detailed information for each request.

Configuration Considerations

Consider the following in configuring this report:

- Users can further configure the report.
- Additional hidden fields can be configured to restrict the set of data in the report.

Field Definitions

This report is pre-configured to have a single field displayed.

Figure D-3 shows the Submit Report window you use to specify which requests you want to appear in the OraApps Critical Requests Summary Report and which information you want to appear for each request. *Table D-4* provides definitions for the displayed fields and for some fields that are hidden by default.

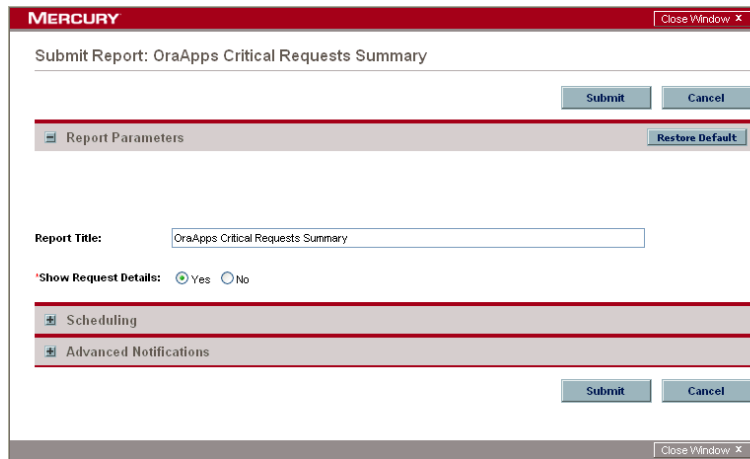


Figure D-3. OraApps Critical Requests Summary Report

Table D-4. OraApps Critical Requests Summary Report field definitions

Field Name (*Required)	Definition	Hidden by Default?
Report Title	Title of the report.	No
*Show Request Details	Whether or not the report will show the Request Details table.	No
Request Type	Requests having this request type.	Yes
Status	Requests having this status.	Yes
Priority	Critical requests, in keeping with this report's purpose.	Yes
Assigned To	Requests assigned to this user.	Yes
Assigned To Group	Requests assigned to this group.	Yes
Department	Requests logged against this department.	Yes
Application	Requests that are associated with this application.	Yes
Workflow	Requests that are associated with this workflow.	Yes
Request Group	Requests using this request group.	Yes
Creation Date From	Requests created on this date or later.	Yes
Creation Date To	Requests created on this date or earlier.	Yes
Order By	Criterion to use to sort the requests in the report.	Yes

OraApps IT Demand Summary Report

This section provides information about the OraApps IT Demand Summary Report.

Description

The OraApps IT Demand Summary Report lists a summary of selected Extension-related requests. The report is similar to the Request Summary Report in Mercury Demand Management. It displays the following columns:

- Request Type
- Priority
- Assigned Group
- Count

Optionally, this report provides subtotals. This report is available to all Mercury IT Governance Center users by default.

Configuration Considerations

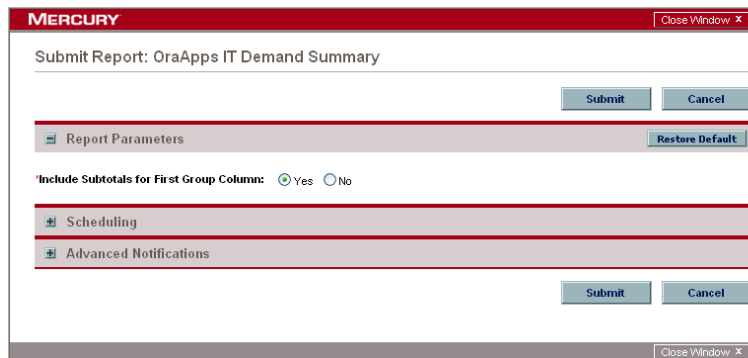
Consider the following in configuring this report:

- Users can further configure the report.
- Additional hidden fields can be configured to restrict the set of data in the report.

Field Definitions

This report is pre-configured to have a single field displayed.

Figure D-4 shows the Submit Report window you use to specify which information you want to appear for each request in the OraApps IT Demand Summary Report. Hidden fields, when enabled, allow you to limit which requests are included in the report. *Table D-5* provides definitions for the displayed fields and for some fields that are hidden by default.



The screenshot shows a web-based interface for submitting a report. At the top, there is a red header bar with the word "MERCURY" on the left and a "Close Window" button with a small 'x' icon on the right. Below the header, the text "Submit Report: OraApps IT Demand Summary" is displayed. Underneath this text are two buttons: "Submit" and "Cancel". A horizontal separator line follows. Below the separator is a section titled "Report Parameters" with a "Restore Default" button to its right. Underneath "Report Parameters" is a radio button group for "Include Subtotals for First Group Column:", with "Yes" selected and "No" unselected. Another horizontal separator line is present. Below it are two more sections: "Scheduling" and "Advanced Notifications", each with a small icon to its left. At the bottom of the form area are two buttons: "Submit" and "Cancel". A final "Close Window" button with an 'x' icon is located at the very bottom right of the window.

Figure D-4. OraApps IT Demand Summary Report

Table D-5. OraApps IT Demand Summary Report field definitions

Field Name (*Required)	Definition	Hidden by Default?
*Include Subtotals for First Group Column	Whether or not the report will show subtotals for the first column specified in the Group by field.	No
Request Numbers	Request numbers to include in the report.	Yes
Include Closed Requests	Whether or not to include requests that have been closed or cancelled.	Yes
Request Type	Requests having this request type.	Yes
Status	Requests having this status.	Yes
Priority	Requests having this priority.	Yes
Assigned To	Requests assigned to this user.	Yes
Assigned To Group	Requests assigned to this group.	Yes
Created By	Requests created by this user.	Yes
Request Sub Type	Requests having this sub-type.	Yes
Department	Requests logged against this department.	Yes
Application	Requests that are associated with this application.	Yes
Workflow	Requests that are associated with this workflow.	Yes
Request Group	Requests using this request group.	Yes
Contact	Requests having this contact name.	Yes
Company Name	Requests having this company name.	Yes
Creation Date From	Requests created on this date or later.	Yes
Creation Date To	Requests created on this date or earlier.	Yes
Last Update Date From	Requests last updated on this date or later.	Yes
Last Update Date To	Requests last updated on this date or earlier.	Yes
Description Contains	Requests having this description text string (not a case sensitive search).	Yes
Group By	How to group the report results by a particular field or fields from the request.	Yes

OraApps Ready Task List Report

This section provides information about the OraApps Ready Task List Report.

Description

The OraApps Ready Task List Report lists all the Oracle project tasks that are in the Ready state. This report is similar to the Project Task Assignment Report in Mercury Project Management. It displays the following columns:

- State
- Scheduled Duration
- Scheduled Start Date
- Scheduled Finish Date
- Resource/Project Manager
- Project Template

Configuration Consideration

The report uses a validation that lists only projects having a Master Project Name such as %OraApps%. This is done to restrict the list to Oracle-related projects. This validation can be copied and modified to list projects named in accordance with your organization's naming convention.

Field Definitions

This report is pre-configured to have a single field displayed.

Figure D-5 shows the Submit Report window you use to specify which Ready tasks you want to appear in the OraApps Ready Task List Report and which information you want to appear for each task. *Table D-6* provides definitions for the displayed fields and for some fields that are hidden by default.

Figure D-5. OraApps Ready Task List Report

Table D-6. OraApps Ready Task List Report field definitions

Field Name (*Required)	Definition	Hidden by Default?
*Project Name	Name of the Oracle Master Project.	No
Resource	Name of the user assigned as a resource.	Yes
Resource Group	Name of the security group associated with the task.	Yes
Task Name	Name of the task to which the user has been assigned.	Yes
Task Category	Category for the task.	Yes
Exceptions of Type	Tasks with exceptions of a specific exception type (for example, late tasks or unassigned tasks) or a group of exception types.	Yes
Department	Department or division responsible for the project.	Yes
Project Template	Template for the project.	Yes
Sched Start Date From	Tasks scheduled to start on this date or later.	Yes
Sched Start Date To	Tasks scheduled to start on this date or earlier.	Yes
Sched Finish Date From	Tasks scheduled to finish on this date or later.	Yes
Sched Finish Date To	Tasks scheduled to finish on this date or earlier.	Yes
Project Manager	Name of the project manager.	Yes

OraApps Unassigned Ready Tasks Report

This section provides information about the OraApps Unassigned Ready Tasks Report.

Description

The OraApps Unassigned Ready Tasks Report lists Oracle project tasks that are in the Ready state but are not yet assigned. It displays the following columns:

- State
- Scheduled Duration
- Scheduled Start Date
- Scheduled Finish Date
- Resource/Project Manager
- Project Template

Configuration Consideration

The report uses a validation that lists projects having a Master Project Name such as %OraApps%. This is done to restrict the list to Oracle-related projects. This validation can be copied and modified to list projects named in accordance with your organization's naming convention.

Field Definitions

This report is pre-configured to have a single field displayed.

Figure D-6 shows the Submit Report window you use to specify which unassigned Ready tasks you want to appear in the OraApps Unassigned Ready Tasks Report and which information you want to appear for each task.

Table D-7 provides definitions for the displayed fields and for some fields that are hidden by default.

Figure D-6. OraApps Unassigned Ready Tasks Report

Table D-7. OraApps Unassigned Ready Tasks Report field definitions

Field Name (*Required)	Definition	Hidden by Default?
*Project Name	Name of the Oracle Master Project.	No
Resource	Name of the user assigned as a resource.	Yes
Resource Group	Name of the security group associated with the task.	Yes
Task Name	Name of the task to which the user has been assigned.	Yes
Task Category	Category for the task.	Yes
Exceptions of Type	Tasks with exceptions of a specific exception type (for example, late tasks or unassigned tasks) or a group of exception types.	Yes
Department	Department or division responsible for the project.	Yes
Project Template	Template for the project.	Yes
Sched Start Date From	Tasks scheduled to start on this date or later.	Yes
Sched Start Date To	Tasks scheduled to start on this date or earlier.	Yes
Sched Finish Date From	Tasks scheduled to finish on this date or later.	Yes
Sched Finish Date To	Tasks scheduled to finish on this date or earlier.	Yes
Project Manager	Name of the project manager.	Yes

Patch-Related Reports

In addition to the Change Management reports common for all package processing, there are reports specific to the Patch Management functionality of the Extension. These reports provide detailed information about patches that have been applied or are in progress.

OraApps Patch Analysis Report

This section provides information about the OraApps Patch Analysis Report.

Description

The OraApps Patch Analysis Report is similar to the OraApps Patch Detail Report, except that it does not allow loading of patch data. Also, rather than giving information in a master-detail format, it consolidates all the information into a single table, which allows you to bring the information into spreadsheet programs such as Microsoft Excel for additional reporting. In addition to having a variety of filter criteria, you can also configure which information to show in the table and in what order. For example, the user might only want to see the patch header and included bug information, or just the bug and file information.

You can use the OraApps Patch Analysis Report to show, for example, which patches have been applied to a given environment or which patches contain a specific file. Only unique value combinations are displayed, eliminating the need to filter through multiple records.

Field Definitions

Figure D-7 shows the Submit Report window you use to specify which patches you want the OraApps Patch Analysis Report to analyze and which information you want to appear for each patch. *Table D-8* provides definitions for the fields.

Figure D-7. OraApps Patch Analysis Report

Table D-8. OraApps Patch Analysis Report field definitions

Field Name (*Required)	Definition
*Report Title	Title of the report.
Patch	Patch number to analyze.
Bug	Bug (subpatch) number to analyze. If the Include All Data of Selected Patches field is set to Yes , the report lists all the information for patches that contain this bug. If it is set to No , it lists only information relevant to this bug.
Patch Application	Application to which the patch, subpatch, or bug belongs.
Environment	Environment for which applied patches will be reported.
Applied Since	Patches applied on this date or later.

Table D-8. OraApps Patch Analysis Report field definitions [continued]

Field Name (*Required)	Definition
File Type	Patches that include files of this file type. If the Include All Data of Selected Patches field is set to Yes , the report lists information for patches that contain files of this file type. If it is set to No , it lists only information relevant to the files of this file type.
File Application	Patches that include files for this application. See File Type field for analogous behavior.
File Name	Patches that include this file name. See File Type field for analogous behavior. This field is case sensitive.
File Version	Patches that include files of this file version. See File Type field for analogous behavior.
*OraApps Release	Oracle release to report on. Default is 11i .
*Include All Data of Selected Patches	When filtering for subpatch or file information, whether or not to show all patch information for the patches that have bugs or files meeting the filter criteria (as opposed to showing only the specific filtered information). Default is No (show only specific filtered information).
*Include Non-Applied Patches	Whether or not to include patches that have never been applied.
*Patch Info Sequence	Whether or not to show the patch header (patch number, patch application) in the report, and if so, what order to place it in the output table. If multiple fields have the same sequence number, the report orders the information based on the field ordering in the report type. Also, if a sequence is missed (for example, the sequences are set to 1, 3, 4, and Don't Show), the report ignores the gap. Finally, if all the columns are set to Don't Show, the report lists only the patch header information.
*SubPatch Info Sequence	Whether or not to show the subpatch header (for example, bug number, bug app) in the report, and if so, what order to place it in the output table. See Patch Info Sequence for additional information
*Env Info Sequence	Whether or not to show environment information (environment name, last applied) in the report, and if so, what order to place it in the output table. See Patch Info Sequence for additional information.
*File Info Sequence	Whether or not to show file information (for example, file name, file version) in the report, and if so, what order to place it in the output table. See Patch Info Sequence for additional information.

OraApps Patch Detail Report

This section provides information about the OraApps Patch Detail Report.

Description

The OraApps Patch Detail Report lists details of selected patches. This report is also used to load patch data for unapplied patches. Users have control over the types of information displayed and the patches selected. If fields are selected to load patch data, the report type calls a special command to load the data prior to executing the report. If required, the report will call the same command to remove the patch data after report completion.

Configuration Considerations

You should carefully control who is allowed to load and purge patch data. If multiple groups of users need access to OraApps Patch Detail Report, but not all of them should be able to load and purge patch data, make additional copies of this report for these users, hiding the load and purge fields.

Field Definitions

Figure D-8 shows the Submit Report window you use to specify which patches you want the OraApps Patch Detail Report to report on or load, and which information you want to appear for each patch. Table D-9 provides definitions for the displayed fields and for some fields that are hidden by default.

Figure D-8. OraApps Patch Detail Report

Table D-9. OraApps Patch Detail Report field definitions

Field Name (*Required)	Definition	Hidden by Default?
Patch	Patch number to report on or load. Required to load data.	No
Includes Bug	Patches containing this bug or subpatch. Can be configured only when the Load Data field is set to No .	No
*OraApps Release	Oracle release to report on. Default is 11i .	No
Application	Application to which the patch, subpatch, or bug belongs. Can be configured only when the Load Data field is set to No .	No
Environment	Environment for which applied patches will be reported. Can be configured only when the Load Data field is set to No .	No

Table D-9. OraApps Patch Detail Report field definitions [continued]

Field Name (*Required)	Definition	Hidden by Default?
Applied Since	Patches applied on this date or later. Can be configured only when the Load Data field is set to No .	No
Includes File	Patches that include this file. Can be configured only when the Load Data field is set to No .	No
Includes File Version	Patches that include this file version. Can be configured only when the Load Data field is set to No .	No
*Order Files By	If files are listed, controls the sort order of the file records.	No
*Order Bugs By	If bugs are listed, controls the sort order of the bug records.	No
*Show SubPatch Info	Whether or not the report will show subpatch details.	No
*Show Env Deploy Info	Whether or not the report will show environment patch application details.	No
*File Details Display	Whether and how file details are included in the output. Options are None and Unique Files , that is, files associated with the patch.	No
*Load Data	Whether the C- or U-driver for the patch will be parsed and loaded prior to running the report. Used to load or report on patches prior to applying them to any environment.	No
Patch Archive	If patch data is being loaded, the name of the .zip patch archive file where the C- or U-driver can be found. This archive should be in the patch repository (patch stage environment).	No
Remove Data after Run	Whether or not patch data should be removed after running the report, if it was loaded beforehand. Should be set to Yes if data is not required for additional reporting prior to application in OraApps environments. Can be configured only when the Load Data field is set to Yes .	No
C or U Driver File	If patch data is being loaded, name of the C- or U-driver file to extract from the Patch Archive. This driver file is used to load information regarding the patch.	No
Append Data	Whether or not to evaluate the entire C- or U-driver if the patch already exists in the detail tables. The default value of No should not be changed under normal circumstances.	Yes

Patch Application Comparison Report

This section provides information about the Patch Application Comparison Report.

Description

This report is similar to the Environment Comparison by Objects Migrated Report in Mercury Change Management, but it is specialized for Oracle patches. This report can be used to obtain a list of patches migrated to or applied to up to six different environments. The report lists each patch number and the last migration date for the patch into each environment, allowing you to determine whether a patch was applied out of order or was missed in one environment.

The Patch Type field for this report represents the object types used for patching. By default, the available values are **R10 Oracle Patch** and **R11 Oracle Patch**. The user can extend the list of values available to the report by adding values to the OA - Patching Object Type Names validation. Two examples of extending the list of values available to the report would be:

- Modifying the name of an object type
- Adding special-purpose object types

The Meaning field of the validation entry must exactly match the name of the object type to be reported on.

Field Definitions

Figure D-9 shows the Submit Report window you use to specify which patches and environments you want the Patch Application Comparison Report to report on. *Table D-10* provides definitions for the fields.

The screenshot shows a web-based form titled "MERCURY" with a "Close Window" button in the top right. The main heading is "Submit Report: Patch Application Comparison Report". Below the heading are "Submit" and "Cancel" buttons. The form is organized into three main sections:

- Report Parameters:** This section is expanded and contains the following fields:
 - *Patch Type: A dropdown menu.
 - *Environment 1: A text input field with a help icon.
 - Environment 2: A text input field with a help icon.
 - Environment 3: A text input field with a help icon.
 - Environment 4: A text input field with a help icon.
 - Environment 5: A text input field with a help icon.
 - Environment 6: A text input field with a help icon.
 - Transaction Date From: A date input field with a calendar icon.
 - Transaction Date To: A date input field with a calendar icon.
 - App Code: A text input field.
 - Performed by: A text input field with a user selection icon.
 - Order by: A dropdown menu with "Last Update Date" selected.
- Scheduling:** A collapsed section.
- Advanced Notifications:** A collapsed section.

At the bottom of the form, there are "Submit" and "Cancel" buttons, and a "Close Window" button in the bottom right corner.

Figure D-9. Patch Application Comparison Report

Table D-10. Patch Application Comparison Report field definitions

Field Name (*Required)	Definition
*Patch Type	Patch type for which to obtain migration information (by default, R10 Oracle Patch or R11 Oracle Patch).
*Environment 1	Environment to be reported on.
Environment 2 through Environment 6	Other environments to report on.
Transaction Date From	Transactions that occurred on this date or later.
Transaction Date To	Transactions that occurred on this date or earlier.
App Code	Transactions performed on package lines using this App Code.
Performed by	Transactions performed by this user.
Order by	Criterion to use to sort the report: Patch Number or Last Update Date .

Patches Applied to an Environment Report

This section provides information about the Patches Applied to an Environment Report.

Description

Similar to the Environment/Objects Detail Report in Mercury Change Management, the Patches Applied to an Environment Report lists Oracle patches migrated to or applied to a specific environment. You can use this report to see the history of Change Management executions for these patches. For example, you can determine whether an Oracle Application patch has been applied multiple times to the same environment. You can also use this report to obtain a list of all the patches applied in a specific date range or all the patches run after a specific patch was applied. This report also has Web links to execution logs and `README.txt` files for each patch.

The Patch Type field for this report represents the object types used for patching. The user can extend the list of values available to the report by adding values to the OA - Patching Object Type Names validation. Two examples of extending the list of values available to the report are:

- Modifying the name of an object type
- Adding special-purpose object types

The Meaning field of the validation entry must exactly match the name of the object type to be reported on.

Field Definitions

Figure D-10 shows the Submit Report window you use to specify which patches and environments you want the Patches Applied to an Environment Report to report on. Table D-11 provides definitions for the fields.

Figure D-10. Patches Applied to an Environment Report

Table D-11. Patches Applied to an Environment Report field definitions

Field Name (*Required)	Definition
*Patch Type	Patch type for which to obtain migration information (by default, R10 Oracle Patch or R11 Oracle Patch).
*Environment	Environment to be reported on.
Transaction Date From	Transactions that occurred on this date or later.
Transaction Date To	Transactions that occurred on this date or earlier.
App Code	Transactions performed on package lines using this App Code.
Performed by	Transactions performed by this user.
*Order by	Criterion to use to sort the report: Patch Number or Last Update Date .
Since Patch	Only migrations that have occurred after this patch number was applied to the specified environment.

Pending Patches Report

This section provides information about the Pending Patches Report.

Description

This functionality is a subset of the Packages Pending Report in Mercury Change Management that is specific to Oracle patches. It lists package lines for Oracle patches that are waiting at a migration step (that is, lines that have been approved for a given environment to which the patch needs to be applied). This report also has Web links to execution logs and `readme.txt` files if the patch has been applied.

The Patch Type field for this report represents the object type used for patching. The user can extend the list of values available to the report by adding values to the OA - Patching Object Type Names validation. Two examples of extending the list of values available to the report are:

- Modifying the name of an object type
- Adding special-purpose object types

The Meaning field of the validation entry must exactly match the name of the object type to be reported on.

Field Definitions

Figure D-11 shows the Submit Report window you use to specify which patches and environment you want the Pending Patches Report to report on. *Table D-12* provides definitions for the fields.

Figure D-11. Pending Patches Report

Table D-12. Pending Patches Report field definitions

Field Name (*Required)	Definition
*Patch Type	Patch type for which to obtain migration information (by default, R10 Oracle Patch or R11 Oracle Patch).
Environment	Environment that has package lines for Oracle patches that are waiting at a migration step.
App Code	Transactions performed on package lines using this App Code.

Appendix **E** Tokens

In This Appendix:

- *Overview of Tokens*
 - *List of Tokens*
-

Overview of Tokens

While configuring certain features in Mercury IT Governance Center, it is often necessary to refer to information that is undefined until it is used in a specific context. Instead of generating entities that are valid only in specific contexts, variables can be used to create general entities that can be applied to a variety of contexts. These variables are called tokens.

Tokens can be used in (but are not limited to) the following Mercury IT Governance Center entities:

- Object types
- Request types
- Validations and SQL statements
- Report types
- Workflow executions and notifications
- Workflow steps

For More Information

For more information about tokens, see the *Commands, Tokens, and Validations Guide and Reference*.

List of Tokens

Table E-1 lists and describes some of the standard tokens included in the Extension.

Table E-1. Tokens included in the Extension

Prefix	Token	Description
ENV.AC	OA_APPLMGR_PASSWORD	Application Manager password for the Oracle installation on the server host. This value is encrypted.
ENV.AC	OA_APPLMGR_USERNAME	Application Manager user name for the Oracle installation on the server host.
ENV.AC	OA_APPS_DB_LINK	Name of the database link from the Mercury IT Governance schema to the APPS schema of the environment's database.
ENV.AC	OA_APPS_PASSWORD	Password for the APPLSYS schema of the environment's database. This value is encrypted.
ENV.AC	OA_APPS_USERNAME	User or schema name of the APPLSYS account of the environment's database.
ENV.AC	OA_APPS_VERSION	Version of the Oracle E-Business Suite installation (10 for Release 10, 11 for Release 11 or 11i).
ENV.AC	OA_CONC_REQUEST_PASSWORD	Password used by Mercury IT Governance Center to retrieve Oracle Concurrent Request logs from the Oracle E-Business Suite instance. This value is encrypted. Only required if this instance will execute Object Migrator requests.
ENV.AC	OA_CONC_REQUEST_USERNAME	User name used by Mercury IT Governance Center to retrieve Oracle Concurrent Request logs from the Oracle Server. Required only if this instance will execute Object Migrator requests.
ENV.AC	OA_MT_APPSFORMS	Flag indicating whether forms are installed in the APPL_TOP of the Oracle installation on the middle-tier (client) host.
ENV.AC	OA_MT_CONC_PROGRAMS	Flag indicating whether concurrent programs are installed in the APPL_TOP of the Oracle installation on the middle-tier (client) host.
ENV.AC	OA_MT_DATABASE_FILES	Flag indicating whether database files are installed in the APPL_TOP of the Oracle installation on the middle-tier (client) host.
ENV.AC	OA_MT_SELFSESV_APPS	Flag indicating whether self-service applications are installed in the APPL_TOP of the Oracle installation on the middle-tier (client) host.

Table E-1. Tokens included in the Extension [continued]

Prefix	Token	Description
ENV.AC	OA_NCA_PASSWORD	Application Manager password for the Oracle installation on the middle-tier (client) host. This value is encrypted.
ENV.AC	OA_NCA_PATCH_TOP	Directory path of the NCA Patch Top on the middle-tier (client) host.
ENV.AC	OA_NCA_USERNAME	Application Manager user name for the Oracle installation on the middle-tier (client) host.
ENV.AC	OA_OM_DB_VALUE	Value identifying this Oracle instance to Object Migrator as recorded in the Oracle value set CLM_DATABASES.
ENV.AC	OA_OM_INSTALLED	Flag to indicate whether Object Migrator is installed in this environment's database.
ENV.AC	OA_REMOTE_OM_ENV_ID	ID of the environment containing the Object Migrator instance to use for AOL Object Migrations to this environment.
ENV.AC	OA_SERVER_APPSFORMS	Flag indicating whether forms are installed in the APPL_TOP of the Oracle installation on the server host.
ENV.AC	OA_SERVER_CONC_PROGRAMS	Flag indicating whether concurrent programs are installed in the APPL_TOP of the Oracle installation on the server host.
ENV.AC	OA_SERVER_DATABASE_FILES	Flag indicating whether database files are installed in the APPL_TOP of the Oracle installation on the server host.
ENV.AC	OA_SERVER_PATCH_TOP	Directory path of the NCA Patch Top for server files in the Oracle installation on the server host.
ENV.AC	OA_SERVER_SELFSEV_APPS	Flag indicating whether self-service applications are installed in the APPL_TOP of the Oracle installation on the server host.
ENV.AC	OA_SERVER_UTL_DIR_PATH	UTL directory path.
ENV.AC	OA_SYSTEM_PASSWORD	Password for the SYSTEM schema of the environment's database. This value is encrypted.
USR.AC	OA_APPS_USERNAME	The Oracle Applications user name used by this Mercury IT Governance Center user.

Numerics

- 11.5.9 Maintenance Procedures Guide (Oracle) 211
- 11.5.9 release (Oracle) 91
- 5.0.0 release, upgrading from 94
- 5.5.0 release
 - product features introduced in 91
 - upgrading from 98
- 6.0.0 release, what's new in 29, 93

A

- ADADMIN utility 202, 204, 224, 258
- ADMGRPCH utility 92, 202, 206, 224
- ADPATCH utility 92, 202, 204, 210, 211, 222, 223, 258
- AKLOAD utility 80, 251
- analyst business role 40
- AOL
 - Conc Prog object type 177
 - Concurrent Request Log 179
 - Resp object type 93
 - User object type 93
- App Code Loader utility 65
- Application Object Library (AOL) 43
- Apps DBA submenu 116

APPSORA.env file 79

B

- Bugs Reported to Oracle portlet 130
- business area concept 40
- business lead business role 38
- business roles
 - analyst 40
 - business lead 38
 - concept 36
 - database administrator (DBA) 38
 - developer 39
 - project manager 36
 - project sponsor 37
 - team lead 39
 - track owner 37

C

- Change Management submenu 108
- changes
 - overview of 168
 - requesting 169
- CLM_DATABASES value set 68
- CLM_DB_LINKS value set 68
- cloning
 - instances 185

- requirements 184
- Cloning Oracle Applications Release 11i white paper 184
- commands
 - ksc_oa_capture_patch 230
 - ksc_oa_copy_clone_scripts 196
 - ksc_oa_purge_temp_patch 230
- concepts
 - business area 40
 - business roles 36
 - source 40
 - track 40
- CONCSUB requests 33
- concurrent request logs 43
- Concurrent Request WatchDog 86
- concurrent requests (Object Migrator), remote execution of 66
- configuring
 - database links 58
 - environments 59
 - instances 188
 - object types 77
 - Oracle Applications users 70
 - request types 72
 - security groups 71
 - server.conf parameters 57
 - the Extension 57
 - workflows 74
- Copy (C) driver 211

D

- Dashboard pages, adding 118
- Database (D) driver 211
- database administrator (DBA) business role 38
- database links, configuring 58
- ddlDriver.#####.log 55
- developer business role 39
- documents
 - prerequisite 26
 - related 26
- Download Center (Mercury IT Governance),

- URL for 50
- downloading the installation files 50
- drivers
 - Copy (C) 211
 - Database (D) 211
 - Generate (G) 211
 - Unified (U) 211, 222

E

- environments, configuring 59
- Extension overview 27

F

- fnd_apps_grants.#####.log 55
- fnd_apps_grants.sql script 83
- fnd_drop_dummy.#####.log 55
- fnd_drop_dummy.sql script 82, 84
- fnd_knta_synonyms.#####.log 55
- fnd_knta_synonyms.sql script 83
- fnd_linked_synonyms.#####.log 55
- fnd_linked_synonyms.sql script 85
- FTP connectivity 43, 44

G

- gap analysis 145
- General Ledger (Oracle) 43
- Generate (G) driver 211
- GL Concurrent Request Log 179

I

- init.ora file 44
- installing
 - collecting information for 49
 - deleting the files 81
 - downloading the files 50
 - installation summary report 52
 - logs generated during 55
 - preparing for 47
 - procedure for 50

Instance Management submenu **109**

instances

- cloning **185**
- overview of **182**

instances, configuring **188**

integrating with Oracle E-Business Suite **33, 82**

integration

- determining the method for **82**
- options **33**

integration method

- changing to linked **84**
- changing to local **82**

Issue Management submenu **107**

issues

- capturing and tracking **135**
- overview of **134**
- reports about **138**

IT Governance Server, setting fields in **57**

ITG_DEPLOYMENT_HOME environment variable **48**

ITG_Home directory **48**

ITG_Home/Deployment_Home directory **48**

J

jarxvf.#####.log **55**

K

kDeploy.sh script **51, 52**

ksc_oa_capture_patch command **230**

ksc_oa_copy_clone_scripts command **196**

ksc_oa_purge_temp_patch command **230**

L

license agreement for the Extension **50**

linked database integration with Oracle E-Business Suite **35**

local integration with Oracle E-Business Suite **34**

logs **55**

AOL Concurrent Request **179**

concurrent request **43**

ddlDriver.#####.log **55**

fnf_apps_grants.#####.log **55**

fnf_drop_dummy.#####.log **55**

fnf_knta_synonyms.#####.log **55**

fnf_linked_synonyms.#####.log **55**

GL Concurrent Request Log **179**

jarxvf.#####.log **55**

packageDriver.#####.log **55**

postXMLDriver.#####.log **55**

preXMLDriver.#####.lo **56**

recompile_invalid.#####.log **56**

M

menus **115**

- Apps DBA submenu **116**
- Change Management submenu **108**
- Instance Management submenu **109**
- Issue Management submenu **107**
- Oracle Applications **106**
- Oracle Applications Roles **106, 111**
- Patch Management submenu **110**
- Project Management submenu **108, 113**
- Project Sponsor submenu **112**

Mercury GL Migrator, installing **47**

Mercury IT Governance Center database, maintaining **231**

Mercury IT Governance Center schema **44**

Mercury IT Governance Server

- maintaining **231**
- setting fields in **57**

Mercury Object Migrator, installing **47**

mitg_extensions.sh script **51**

mitg-600-OracleApps.jar file **48, 50, 51**

MKS Toolkit **211**

O

OA - OM Concurrent Managers Rel 10 Validation **244**

OA - Oracle Patch Admin Security Group **213**

- Object Migrator concurrent requests, remote execution of **66**
- object types
 - AOL Conc Prog **177**
 - AOL Resp **93**
 - AOL User **93**
 - configuring **77**
 - OraApps 11i Cloning **195, 196**
 - OraApps ADADMIN Compile APPS Schema **204**
 - OraApps ADADMIN Compile Flexfields **204**
 - OraApps ADADMIN Generate Jar Files **204**
 - OraApps ADADMIN Generate Messages **204**
 - OraApps ADADMIN-related **224**
 - OraApps AKLOAD Migrate AK Setups **80**
 - OraApps Oracle Workflow **79**
 - OraApps R11i Merge Patch **206, 224**
 - R10 Oracle Patch **222**
 - R11 Oracle Patch **223**
- open_links database parameter **45**
- OraApps 11i Cloning object type **195, 196**
- OraApps 11i Cloning workflow **187, 195, 373**
- OraApps ADADMIN Compile APPS Schema object type **204**
- OraApps ADADMIN Compile Flexfields object type **204**
- OraApps ADADMIN Generate Jar Files object type **204**
- OraApps ADADMIN Generate Messages object type **204**
- OraApps ADADMIN-related object types **224**
- OraApps AKLOAD Migrate AK Setups object type **80**
- OraApps Application Issue request type **135, 287**
- OraApps Application Issue workflow **135, 136, 324**
- OraApps Apps Issues Detail Report **96, 383**
- OraApps Apps Issues Summary Report **96, 387**
- OraApps Cloning Process workflow **185, 193, 194, 196, 329**
- OraApps Cloning Request request type **185, 193, 291**
- OraApps Config. Deployments portlet **131**
- OraApps Conversion Process workflow **171, 331**
- OraApps Conversion Request request type **171, 294, 331**
- OraApps Critical Requests Summary Report **138, 160, 166**
- OraApps Customization/Configuration Deployment workflow **73, 170, 175, 335**
- OraApps Design & Development request type **173, 297**
- OraApps Design & Development workflow **173, 341**
- OraApps Enhancement Process workflow **344**
- OraApps Enhancement Request request type **172, 300, 344**
- OraApps GAP Analysis Process workflow **145, 164, 348**
- OraApps GAP Analysis Request request type **145, 162, 303**
- OraApps Interfaces Process workflow **170, 352**
- OraApps Interfaces Request request type **170**
- OraApps Issues Detail Report **92, 138**
- OraApps Issues Summary Report **92, 139**
- OraApps IT Demand Summary Report **92, 96, 139, 160, 166, 393**
- OraApps Oracle Patch Deployment workflow **356**
- OraApps Oracle Workflow object type **79**
- OraApps Patch Analysis Report **200, 208, 229, 400**
- OraApps Patch Data Capture Subworkflow **201, 227**
- OraApps Patch Deployment workflow **204, 226, 227**

-
- OraApps Patch Detail Report **200, 208, 229, 403**
 - OraApps R11i Merge Patch object type **206, 224**
 - OraApps Ready Task List Report **161, 166, 396**
 - OraApps Release 11i Upgrade project template **143, 146, 153**
 - OraApps Report Request request type **169, 309**
 - OraApps Reports Process workflow **169, 362**
 - OraApps Setup Change Process workflow **174, 366**
 - OraApps Setup Change Request request type **174, 312**
 - OraApps Status Update Request request type **158, 163, 315**
 - OraApps Status Update Request workflow **159, 165, 368**
 - OraApps Unassigned Ready Tasks Report **161, 166, 397**
 - OraAppsStatus Update Request workflow **165**
 - Oracle Applications menu **106**
 - Oracle Applications Roles menu **106, 111**
 - Oracle Apps 11i All Phases portlet **125**
 - Oracle Apps 11i Overall Status portlet **120, 124**
 - Oracle Apps 11i Reports Track Status portlet **127**
 - Oracle Apps Activities Summary (Past 2 Weeks) portlet **122**
 - Oracle Apps Activity by Priority portlet **121**
 - Oracle Apps Critical Activities portlet **121, 124**
 - Oracle Apps DBA page **129**
 - Oracle Apps Instance Cloning Queue portlet **131**
 - Oracle Apps Patch Deployment Status portlet **130**
 - Oracle Apps Project Manager page **119**
 - Oracle Apps Project Sponsor page **123**
 - Oracle Apps Reports Track Owner page **126**
 - Oracle E-Business Suite, integrating with **33**
 - Oracle General Ledger **43**
 - Oracle patch #2594863 **80**
 - overview of the Extension **27**
- P**
- packageDriver.#####.log **55**
 - pages
 - Oracle Apps DBA **129**
 - Oracle Apps Project Manager **119**
 - Oracle Apps Reports Track Owner **126**
 - overview **117**
 - Project Sponsor **123**
 - Patch Application Comparison Report **209, 406**
 - patch archive, naming **207**
 - Patch Management submenu **110**
 - patch repository, setting up **212**
 - Patch Stage environment, creating **214**
 - patches
 - analyzing information about **199**
 - applying **202**
 - merging **206**
 - Oracle #2594863 **80**
 - Oracle release 10 versus release 11 and 11i **203**
 - overview of **198**
 - reporting on **207**
 - Patches Applied to an Environment Report **209, 408**
 - patch-related reports **400**
 - Pending Patches Report **209, 410**
 - portletlets
 - Bugs Reported to Oracle **130**
 - OraApps Config. Deployments **131**
 - Oracle Apps 11i All Phases **125**
 - Oracle Apps 11i Overall Status **120, 124**
 - Oracle Apps 11i reports Track Status **127**
 - Oracle Apps Activities Summary (Past 2 Weeks) **122**
 - Oracle Apps Activity by Priority **121**
 - Oracle Apps Critical Activities **121, 124**
 - Oracle Apps Instance Cloning Queue **131**
 - Oracle Apps Patch Deployment Status **130**
-

- overview 117
- Reports Activity by Priority 128
- Reports Critical Activities 127
- postXMLDriver.#####.log 55
- preconfigured Dashboard pages, adding 118
- prerequisites
 - documents 26
 - knowledge and experience 31
- preXMLDriver.#####.log 56
- primary Object Migrator host 33, 66
- Project Management submenu 108, 113
- project manager business role 36
- project sponsor business role 37
- Project Sponsor submenu 112
- project templates, OraApps Release 11i Upgrade 143, 146, 153
- projects, overview of 142

R

- R10 Oracle Patch object type 222
- R11 Oracle Patch object type 223
- recompile_invalid.#####.log 56
- release 11.5.9 (Oracle) 91
- release 5.0.0, upgrading from 94
- release 5.5.0
 - product features introduced in 91
 - upgrading from 98
- release 6.0.0, what's new in 29, 93
- remote execution of Object Migrator concurrent requests 66
- remote Object Migrator host 66
- report types
 - OraApps Apps Issues Detail 96, 383
 - OraApps Apps Issues Summary 96, 387
 - OraApps Critical Requests Summary Report 138, 160, 166
 - OraApps Issues Detail 92, 138
 - OraApps Issues Summary 92, 139
 - OraApps IT Demand Summary 92, 96, 139, 160, 166, 393

- OraApps Patch Analysis 200, 208, 229, 400
- OraApps Patch Detail 200, 208, 229, 403
- OraApps Ready Task List 161, 166, 396
- OraApps Unassigned Ready Tasks 161, 166, 397
- Patch Application Comparison 209, 406
- Patches Applied to an Environment 209, 408
- Pending Patches 209, 410
- Reports Activity by Priority portlet 128
- Reports Critical Activities portlet 127
- reports, patch-related 400
- request types
 - configuring 72
 - OraApps Application Issue 135, 287
 - OraApps Cloning Request 185, 193, 291
 - OraApps Conversion Request 171, 294, 331
 - OraApps Design & Development 173, 297
 - OraApps Enhancement Request 172, 300, 344
 - OraApps GAP Analysis Request 145, 162, 303
 - OraApps Interfaces Request 170
 - OraApps Report Request 169, 309
 - OraApps Setup Change Request 174, 312
 - OraApps Status Update Request 158, 163, 315

S

- schema, Mercury IT Governance Center 44
- SCP connectivity 43, 44
- scripts
 - fnd_apps_grants.sql 83
 - fnd_knta_synonyms.sql 83
 - fnd_linked_synonyms.sql 85
 - kDeploy.sh 51, 52
 - mitg_extensions.sh 51
 - setServerMode.sh 49, 53
- security groups, configuring 71
- server.conf parameters, configuring 57
- SERVER_ENV_NAME parameter 59, 221

setServerMode.sh script 49, 53
source concept 40
SQL*Net 35
system requirements 42

T

team lead business role 39
tokens
 list of 414
 overview of 414
track concept 40
track owner business role 37
Track Owner submenu 115

U

Unified (U) drivers 92, 211, 222
upgrading
 deleting upgrade files 103
 impacts and guidelines 94
 overview 90
 planning for 143
 process for 102
users (Oracle Applications), configuring 70
utilities
 ADADMIN 202, 204, 224
 ADMRGPCH 202, 206, 224
 ADPATCH 202, 204, 210, 211, 222, 223
 AKLOAD 80, 251
 App Code Loader 65
UTL_FILE_DIR parameter, init.ora file 44

W

WatchDog functionality 86
workflows
 configuring 74
 OraApps 11i Cloning 187, 195, 373
 OraApps Application Issue 135, 136, 324
 OraApps Cloning Process 185, 193, 194, 196, 329
 OraApps Conversion Process 171, 331

OraApps Customization/Configuration
Deployment 73, 170, 175, 335
OraApps Design & Development 173, 341
OraApps Enhancement Process 344
OraApps GAP Analysis Process 145, 164, 348
OraApps Interfaces Process 170, 352
OraApps Oracle Patch Deployment 356
OraApps Patch Data Capture Subworkflow 201, 227
OraApps Patch Deployment 204, 226, 227
OraApps Reports Process 169, 362
OraApps Setup Change Process 174, 366
OraApps Status Update Request 159, 165, 368

