

Oracle
Real-Time Scheduler
Server Application User Guide
Release 2.3.0.0.0
E68097-01

March 2016

Oracle Real-Time Scheduler Server Application User Guide

Release 2.3.0.0.0

E68097-01

March 2016

Documentation build: 4.8.2016 7:11:26 [M1_1460124686000]

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

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

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

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

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

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

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

Contents

Business User Guide.....	18
Overview.....	18
User Interface Standard Features.....	19
System-Wide Standards.....	19
Page Components.....	19
Characteristics Common to All Pages.....	19
Maintenance Pages.....	39
The Unique Identifier Area.....	39
Confirmation & Error Messages.....	40
Warning If Unsaved Changes Exist.....	40
Query Pages.....	40
The Search Area.....	40
Portals.....	40
Portals Are Made Up Of Zones.....	41
Each User Can Customize Which Zones Appear.....	42
Users Must Be Granted Security Access To Each Zone.....	43
Zones May Appear Collapsed When A Page Opens.....	43
Dashboard Portal.....	43
User Preferences.....	46
User - Main.....	47
User - To Do Roles.....	48
User - Access Security.....	48
User - Bookmarks.....	49
User - Portal Preferences.....	49
User - Favorite Links.....	50
User - Favorite Scripts.....	51
User - Characteristics.....	51
User - Miscellaneous.....	52
Initiating Scripts.....	52
Accelerator Key Summary.....	53
Portal and Zone Features.....	55
Zone Types and Characteristics.....	55
Explorer Zones.....	55
Map Zones.....	61
Common Characteristics of Zones.....	62
Common Base Portals and Zones.....	62
All-In-One Portal.....	62
Separate Query and Maintenance Portals.....	63
Activities.....	64
About Activities.....	64
Activity Type Controls Everything.....	65
An Activity Is Associated With One Host System.....	65
Activity Workflow.....	65
Activity Identifiers.....	70
Location.....	71
Service Area.....	72
Service Class.....	72
Estimated Work Duration.....	72
Time Windows.....	73
Appointments.....	75
Emergency Activities.....	76
Activity Priority.....	77
Resource Capabilities.....	78
Resource Restrictions.....	79
Scheduling An Activity.....	81
Expired Activities.....	81
Activity Actions.....	82
Dispatching An Activity.....	85

Completion Information.....	86
Activity Cancellation By Host.....	86
Depot Related Activities.....	87
Handling Keys.....	87
About Complex Activities.....	87
A Complex Activity Is Scheduled Differently.....	88
Complex Activity Workflow.....	90
Visit Task Workflow.....	92
Use Case Example.....	94
Same Crew.....	95
Under Scheduled Alert.....	96
Time Tracking.....	96
Completion Information.....	96
About Activity Dependency.....	96
Job.....	96
Chain.....	98
About Assignments.....	99
Current Assignment.....	100
Assignment Workflow.....	100
Crew Assignment.....	105
Recall.....	107
Concurrent Assignments.....	108
About Activity Attachments.....	108
Attachments Related To The Type Of Work.....	108
Attachments Related To A Specific Activity.....	109
Maintaining Activities.....	110
Request An Assisting Crew.....	110
Maintaining Current Crew Assignment.....	110
Completing Activities with Concurrent Assignments.....	111
Maintaining Complex Activities.....	111
Maintaining Assignments.....	112
Activity Dependency.....	112
Override Time Windows.....	112
About Override Time Windows.....	112
Prohibit Work When Otherwise Allowed.....	112
Allow Work When Otherwise Not Allowed.....	112
Overrides Regenerate Arrival Time Windows.....	113
Enforcing Overrides.....	113
Use Case Examples.....	113
Maintaining Override Time Windows.....	114
Crews.....	114
About Crews.....	114
Work Is Scheduled To Crew Shifts.....	115
Mobile Workers.....	115
Vehicles.....	116
Fixed Crew Resources.....	116
Resource Capabilities May Change.....	117
Crew Locations.....	117
Planned Time Off.....	117
Crew Hierarchy.....	117
Depot Restrictions.....	117
Crew May Be Managed Externally.....	118
Maintaining Mobile Workers.....	118
Maintaining Vehicles.....	118
Maintaining Crews.....	119
My Calendar.....	119
Resource Route Replay.....	119
Crew Shifts.....	120
About Crew Shifts.....	120
Crew Shift Type Controls Everything.....	120
Creating Shifts.....	120
Default Crew Shift Details.....	120
Crew Shift Workflow.....	120
Controlling The Schedule.....	121

Crew Shift Actions.....	125
Planned Unavailability.....	126
Breaks.....	126
Non-Productive Tasks (NPTs).....	129
Meetings.....	130
Locking The Schedule.....	130
Working The Schedule.....	132
About Shift Weekly Templates.....	134
A Weekly Template Is Made Of Shift Templates.....	134
Crew Specific Weekly Template.....	135
Common Weekly Templates.....	136
One Effective Shift Weekly Template For A Crew.....	137
Shift Generation.....	137
Overridden Shift Details.....	138
Draft and Original Templates.....	138
Maintaining Crew Shifts.....	139
Recommended Crew Allocation.....	139
Maintaining Breaks / Non Productive Tasks.....	140
Maintaining Shift Weekly Templates.....	140
Comparing Draft and Original Templates.....	140
Navigating Between Original and Draft Templates.....	141
Maintaining Shift Templates.....	141
Periods of Unavailability (POUs).....	141
About Periods of Unavailability (POUs).....	141
Different Types of Events.....	141
Leave POUs.....	142
Crew Meeting POUs.....	142
Creating POUs.....	144
About POU Templates.....	144
Actual vs. Template POUs.....	144
POU Generation.....	144
Overridden POU Details.....	145
Maintaining Periods Of Unavailability.....	145
Maintaining Period Of Unavailability Tasks.....	146
Dispatchers.....	146
About Dispatchers.....	146
Managing Crews And Activities.....	146
Dispatcher Shifts.....	147
Handling Alerts.....	148
KPIs.....	148
To Dos In Scope.....	148
About CDI.....	149
CDI is for Dispatchers Only.....	149
CDI Data.....	149
CDI Map.....	149
CDI Is Customizable.....	150
Maintaining Dispatchers.....	150
Maintaining Dispatcher Shifts.....	151
CDI.....	151
Managing your Dispatcher Shift.....	152
Using the Scheduling Gantt.....	152
Gantt Views.....	152
Crews Pane.....	153
The Schedule Area Pane.....	153
The Toolbar.....	155
Customize Your Gantt Display.....	155
Filter The List of Crews.....	156
Taking Actions.....	156
Plan View.....	157
Tasks View.....	158
Using the Activity Search.....	158
Using the CDI Map.....	158
Monitoring KPIs.....	159
Monitoring Contracted Work.....	159

My Alert Queue.....	160
Managing To Do Entries.....	160
KPI Specific CDI.....	160
Dispatcher To Do List.....	160
Alerts.....	161
About Alerts.....	161
Alert Type Controls Everything.....	161
Alert Queue.....	162
Backup Alert Queue.....	162
Alert Actions.....	162
About Alert Areas.....	163
Hazard Areas.....	163
Vehicle Leave Areas.....	163
Excluded Resources.....	164
Maintaining Alerts.....	164
Maintaining Alert Areas.....	164
Procedures.....	164
About Procedures.....	164
Procedure Type Controls Everything.....	165
Shift Related Procedures.....	165
Activity Related Procedures.....	166
Maintaining Procedures.....	166
Timesheets.....	166
About Timesheets.....	167
Timesheet Period.....	167
Time Usage Summary.....	167
Generating Timesheets.....	168
Review and Corrections.....	168
Maintaining Timesheets.....	169
Mail.....	169
About Mail.....	169
Emergency Messages.....	169
Mail Summary.....	170
Maintaining Mail.....	170
Transfer of Goods.....	170
About Transfer of Goods.....	170
The Goods.....	171
The Transfer.....	171
Load Capacity.....	171
Vehicle Maximum Capacity.....	172
Depots.....	172
About Depot Runs.....	174
Depot Task Duration.....	174
Depot Task Workflow.....	174
Depot Task Actions.....	175
About Depot Cutoff.....	176
Processing A Cutoff Event.....	176
Initiating Cutoff Events.....	176
Scheduling Manifest.....	177
Close Entire Shift.....	177
Maintaining Depot Tasks.....	177
Maintaining Depot Time Windows.....	177
Maintaining Depot Cutoffs.....	178
Contractor Management.....	178
About Contractors.....	178
Shift Based Model.....	178
Capacity Model.....	179
Activity Contracting Restrictions.....	181
Contractor Eligibility Rules.....	182
Committing An Activity To A Contractor Company.....	182
Resource Groups.....	182
CDI and Contractors.....	183
About Capacities.....	183
Capacity Type Controls Everything.....	183

Creating Capacities.....	183
Capacity Workflow.....	183
Controlling The Schedule.....	184
Capacity Actions.....	185
About Capacity Weekly Templates.....	185
A Weekly Template Is Made Of Capacity Templates.....	186
Actual vs. Template Capacities.....	186
Defining and Applying A Template.....	186
Making Changes To An Applied Template.....	186
One Effective Weekly Template For A Contractor.....	187
Capacity Generation.....	187
Overridden Capacity Details.....	187
Draft and Original Templates.....	188
Maintaining Resource Groups.....	188
Maintaining Capacities.....	189
Maintaining Capacity Weekly Templates.....	189
Maintaining Capacity Templates.....	189
Maintaining Contractor Eligibility.....	190
Contractor Work Management.....	190
Managing Your Work.....	190
Mobile Application.....	190
About The Crew's Application.....	191
Overview.....	191
The Mobile Device Is Known At Start Time.....	191
Real-time Updates.....	191
Sending Current Location.....	192
Geographic Map.....	192
Enforce Procedures.....	192
Handling Attachments.....	193
Creating Activities in the Field.....	193
Switch To Offline Mode.....	193
Request For Help.....	193
Sending / Receiving Mail Messages.....	194
The End Of Shift Process.....	194
Reviewing Timesheets.....	194
Switching Devices Mid Shift.....	194
About The Contractor's Application.....	195
Overview.....	195
No Dispatch Process.....	195
Completion Only Updates.....	195
About Remote Messages.....	195
Maintaining Remote Messages.....	196
To Do Processing.....	196
The Big Picture of To Do Processing.....	196
To Do Role.....	196
To Do Priority.....	197
Work Distribution.....	197
A User's Current To Do.....	197
Related To Do Entries.....	198
Complete A To Do.....	198
Managing To Do Entries.....	199
To Do Supervisor Functions.....	199
Supervisor To Do Summary.....	199
Supervisor User Summary.....	199
Supervisor To Do Assignment.....	200
To Do List Functions.....	201
To Do Summary.....	201
To Do List.....	201
To Do Entry.....	203
To Do Entry - Add Dialog.....	203
To Do Entry - Main.....	204
To Do Entry - Keys/Values.....	205
To Do Entry Actions.....	205
To Do Search.....	207

To Do Search - Main.....	207
To Do Search Actions.....	209
How To.....	210
How To Assign a To Do to a User.....	210
How To Work on To Do Entries.....	210
How To Add a New To Do.....	211
How To Complete To Dos.....	211
How To Forward a To Do.....	212
How To Send Back a To Do.....	212
How To Add a Log Entry to a To Do.....	212
Reports.....	212
On-line Report Submission.....	212
Report Submission - Main.....	212
Viewing Historic Reports.....	213
Report History - Main.....	213
Administrative User Guides.....	214
Framework Administrative User Guide.....	214
Defining General Options.....	214
Defining Installation Options.....	214
Installation Options - Main.....	214
Installation Options - Messages.....	215
Installation Options - Algorithms.....	215
Installation Options - Accessible Modules.....	217
Installation Options - Installed Products.....	217
Support For Different Languages.....	218
User Language.....	218
Additional Topics.....	219
Defining Languages.....	219
Defining Countries.....	220
Country - Main.....	220
Country - States.....	220
Defining Currency Codes.....	221
Defining Time Zones.....	221
Designing Time Zones.....	221
Setting Up Time Zones.....	222
Setting Up Seasonal Time Shift.....	222
Defining Geographic Types.....	222
Defining Work Calendar.....	223
Defining Display Profiles.....	223
Additional Hijri Date Configuration.....	225
Defining Phone Types.....	225
Setting Up Characteristic Types & Values.....	226
There Are Four Types Of Characteristics.....	226
Searching By Characteristic Values.....	226
Characteristic Type - Main.....	227
Characteristic Type - Characteristic Entities.....	228
Setting Up Foreign Key Reference Information.....	229
Information Description Is Dynamically Derived.....	229
Navigation Information Is Dynamically Derived.....	230
Search Options.....	230
Foreign Key Reference - Main.....	230
Defining Feature Configurations.....	231
Feature Configuration - Main.....	232
Feature Configuration - Messages.....	232
Defining Master Configurations.....	232
Setting Up Master Configurations.....	233
Miscellaneous Topics.....	233
Module Configuration.....	233
Global Context Overview.....	234
System Data Naming Convention.....	235
Caching Overview.....	236
Debug Mode.....	237
System Override Date.....	238
Advanced Search Options.....	238

Defining Security & User Options.....	239
The Big Picture of Application Security.....	239
Application Security.....	239
Action Level Security.....	240
Field Level Security.....	240
Encryption and Masking.....	241
The Base Package Controls One User, One User Group, And Many Application Services.....	246
Importing Security Configuration from an External Source.....	247
The Big Picture of Row Security.....	247
Access Groups, Data Access Roles and Users.....	247
Defining Application Services.....	248
Application Service - Main.....	248
Application Service - Application Security.....	248
Defining Security Types.....	248
Security Type - Main.....	249
Defining User Groups.....	249
User Group - Main.....	249
User Group - Application Services.....	250
User Group - Users.....	250
Defining Access Groups.....	251
Defining Data Access Roles.....	251
Data Access Role - Main.....	251
Data Access Role - Access Group.....	252
Defining Users.....	252
User Interface Tools.....	252
Defining Menu Options.....	252
Menu - Main.....	253
Menu - Menu Items.....	254
The Big Picture of System Messages.....	255
Defining System Messages.....	255
The Big Picture of Portals and Zones.....	257
There Are Three Types of Portals.....	257
Common Characteristics of All Portals.....	257
Common Characteristics of Stand-Alone Portals.....	259
Custom Look and Feel Options.....	260
User Interface.....	260
UI Map Help.....	260
Setting Up Portals and Zones.....	261
Defining Zone Types.....	261
Defining Zones.....	262
Defining Context-Sensitive Zones.....	278
Defining Portals.....	279
Defining Display Icons.....	280
Defining Navigation Keys.....	281
Navigation Key Types.....	281
Navigation Key vs. Navigation Option.....	281
The Flexibility of Navigation Keys.....	281
Linking to External Locations.....	282
Overriding Navigation Keys.....	282
Maintaining Navigation Keys.....	282
Defining Navigation Options.....	283
Navigation Option - Main.....	284
Navigation Option - Tree.....	286
Database Tools.....	286
Defining Table Options.....	286
Table - Main.....	286
Table - Table Field.....	288
Table - Constraints.....	289
Table - Referred by Constraints.....	290
Defining Field Options.....	290
Field - Main.....	290
Field - Tables Using Field.....	291
Defining Maintenance Object Options.....	291
Maintenance Object - Main.....	292

Maintenance Object - Options.....	292
Maintenance Object - Algorithms.....	292
Maintenance Object - Maintenance Object Tree.....	294
Defining Valid Values.....	294
Defining Lookup Options.....	296
Defining Extendable Lookups.....	297
The Big Picture Of Audit Trails.....	299
Captured Information.....	299
How Auditing Works.....	299
The Audit Trail File.....	300
How To Enable Auditing.....	300
Audit Queries.....	301
Bundling.....	303
About Bundling.....	303
Configuring Maintenance Objects for Bundling.....	304
Working with Bundles.....	306
Revision Control.....	308
About Revision Control.....	308
Working with the Revision Control Zones.....	310
Working with the Revision Control Portal.....	312
Information Lifecycle Management	313
The Approach to Implementing Information Lifecycle Management.....	313
Enabling ILM for Supported Maintenance Objects.....	315
Ongoing ILM Tasks.....	317
Archived Foreign Keys.....	317
Configuration Tools.....	317
Business Objects.....	317
The Big Picture of Business Objects.....	317
Defining Business Objects.....	331
Defining Status Reasons.....	338
Business Services.....	338
Service Program.....	338
Defining Business Services.....	339
User Interface (UI) Maps.....	340
Defining UI Maps.....	342
Ensuring Unique Element IDs for UI Maps.....	344
Maintaining Managed Content.....	344
Managed Content - Main.....	344
Managed Content - Schema.....	345
Data Areas.....	345
Defining Data Areas.....	345
Schema Designer.....	346
Schema Viewer.....	348
Business Event Log.....	348
To Do Lists.....	349
The Big Picture of To Do Lists.....	349
To Do Entries Reference A To Do Type.....	349
To Do Entries Reference A Role.....	349
To Do Entries Can Be Rerouted (Or Suppressed) Based On Message Number.....	350
The Priority Of A To Do Entry.....	351
Working On A To Do Entry.....	351
Launching Scripts When A To Do Is Selected.....	351
To Do Entries Have Logs.....	352
How Are To Do Entries Created?.....	352
The Lifecycle Of A To Do Entry.....	355
Linking Additional Information To A To Do Entry.....	355
Implementing Additional To Do Entry Business Rules.....	356
To Do Entries May Be Routed Out Of The System.....	356
Periodically Purging To Do Entries.....	356
Setting Up To Do Options.....	356
Installation Options.....	356
Messages.....	357
Feature Configuration.....	357
Defining To Do Roles.....	358

Defining To Do Types.....	359
List of System To Do Types.....	363
Implementing The To Do Entries.....	363
Background Processes.....	363
The Big Picture of Background Processes.....	363
Background Processing Concepts.....	364
Parameters Supplied To Background Processes.....	364
Processing Errors.....	366
Parallel Background Processes.....	367
Timed Batch Processes.....	368
How to Re-extract Information.....	368
How to Submit Batch Jobs.....	369
How to Track Batch Jobs.....	369
How to Restart Failed Jobs and Processes.....	369
Assessing Level of Service.....	369
System Background Processes.....	369
Defining Batch Controls.....	370
Batch Control - Algorithms.....	372
On-Line Batch Submission.....	372
Batch Submission Creates a Batch Run.....	372
Jobs Submitted in the Background.....	373
Email Notification.....	373
Running Multi-Threaded Processes.....	373
Batch Jobs May End in Error.....	373
Submitting Jobs in the Future.....	374
Lifecycle of a Batch Job Submission.....	374
Batch Job Submission - Main.....	374
Tracking Batch Processes.....	376
Batch Run Tree - Main.....	376
Batch Run Tree - Run Control.....	377
Service Health Check.....	377
The Big Picture of Requests.....	378
Request Type Defines Parameters.....	378
Previewing and Submitting a Request	378
To Do Summary Email.....	379
Exploring Request Data Relationships.....	379
Defining a New Request.....	379
Setting Up Request Types.....	380
Maintaining Requests.....	380
Defining Algorithms.....	380
The Big Picture Of Algorithms.....	381
Algorithm Type Versus Algorithm.....	381
How To Add A New Algorithm.....	381
Minimizing The Impact Of Future Upgrades.....	382
Setting Up Algorithm Types.....	382
List of Algorithm Types.....	383
Setting Up Algorithms.....	383
Defining Script Options.....	384
The Big Picture Of Scripts.....	384
Scripts Are Business Process-Oriented.....	384
A Script Is Composed Of Steps.....	385
Designing And Developing Scripts.....	385
A Script May Declare Data Areas.....	385
Designing Generic Scripts.....	386
Securing Script Execution.....	386
The Big Picture Of BPA Scripts.....	386
How To Invoke Scripts.....	387
Developing and Debugging Your BPA Scripts.....	387
Launching A Script From A Menu.....	387
Launching A Script When Starting The System.....	387
Executing A Script When A To Do Entry Is Selected.....	388
The Big Picture Of Script Eligibility Rules.....	389
The Big Picture Of Server-Based Scripts.....	392
Plug-In Scripts.....	393

Service Scripts.....	394
Debugging Server-Based Scripts.....	395
Maintaining Scripts.....	395
Script - Main.....	395
Script - Step.....	396
Script - Data Area.....	414
Script - Schema.....	414
Script - Eligibility.....	415
Merging Scripts.....	416
Script Merge.....	417
Maintaining Functions.....	420
Function - Main.....	420
Function - Send Fields.....	421
Function - Receive Fields.....	422
Attachments.....	422
Attachment Overview.....	422
Configuring Your System for Attachments.....	423
Maintaining Attachments.....	424
Adding Attachments.....	425
Application Viewer.....	425
Application Viewer Toolbar.....	425
Data Dictionary Button.....	425
Physical and Logical Buttons.....	426
Collapse Button.....	426
Attributes and Schema Button.....	426
Maintenance Object Button.....	426
Algorithm Button.....	427
Batch Control Button.....	427
To Do Type Button.....	427
Description and Code Buttons.....	427
Service XML Button.....	427
Select Service Button.....	428
Java Docs Button.....	428
Classic Button.....	428
Preferences Button.....	428
Help Button.....	428
About Button.....	428
Slider Icon.....	429
Data Dictionary.....	429
Using the Data Dictionary List Panel.....	429
Using the Data Dictionary Detail Panel.....	430
Maintenance Object Viewer.....	431
Using the Maintenance Object List Panel.....	431
Using the Maintenance Object Detail Panel.....	431
Algorithm Viewer.....	432
Using the Algorithm Viewer List Panel.....	432
Using the Algorithm Plug-In Spot Detail Panel.....	432
Using the Algorithm Type Detail Panel.....	432
Using the Algorithm Detail Panel.....	432
Batch Control Viewer.....	432
Using the Batch Control Viewer List Panel.....	433
Using the Batch Control Detail Panel.....	433
To Do Type Viewer.....	433
Using the To Do Type Viewer List Panel.....	433
Using the To Do Type Detail Panel.....	433
Service XML Viewer.....	434
Using the Service XML Viewer Overview Panel.....	434
Using the Service XML Viewer Detail Panel.....	434
Java Docs Viewer.....	434
Using the Java Docs Viewer List Panel.....	435
Using the Java Package Detail Panel.....	435
Using the Java Interface / Class Detail Panel.....	435
Application Viewer Preferences.....	435
Application Viewer Stand-Alone Operation.....	435

Stand-Alone Configuration Options.....	436
Example Application Viewer Configuration.....	436
Application Viewer Generation.....	437
Defining and Designing Reports.....	437
The Big Picture Of Reports.....	437
Integration with BI Publisher and Business Objects Enterprise.....	438
How To Request Reports.....	439
Viewing Reports.....	440
Configuring The System To Enable Reports.....	440
Configuring BI Publisher Reports.....	440
Configuring Business Objects Enterprise Reports.....	441
Defining Reporting Options.....	441
Defining Report Definitions.....	442
Sample Reports Supplied with the Product.....	443
How to Use a Sample Report Provided with the System.....	443
Subreports Used with Crystal Reports.....	444
How To Define A New Report.....	445
Use a Sample Report as a Starting Point.....	445
Publishing Reports in BI Publisher.....	445
Publishing Reports in Business Objects Enterprise.....	446
Designing Your Report Definition.....	447
External Messages.....	449
Incoming Messages.....	449
Inbound Web Services.....	449
Guaranteed Delivery.....	453
Outgoing Messages.....	453
Outbound Messages.....	454
Web Service Adapters.....	466
Sending Email.....	468
XML Application Integration.....	468
The Big Picture of XAI.....	468
Designing Your XAI Environment.....	485
Schema Editor.....	495
Setting Up Your XAI Environment.....	502
Running Your XAI Environment.....	511
Maintaining Your XAI Environment.....	516
Integrations.....	522
LDAP Integration.....	522
LDAP Integration Overview.....	522
Configuring LDAP Integration.....	523
Oracle Identity Manager Integration.....	526
Data Synchronization.....	528
Defining Sync Requests.....	528
Designing Your Sync Requests.....	530
Sync Requests.....	530
Configuration Migration Assistant (CMA).....	530
Understanding CMA.....	531
The CMA Process Flow.....	532
Migration Assumptions, Restrictions and Recommendations.....	533
CMA Configuration.....	534
Master Configuration - Migration Assistant.....	534
Migration Plans.....	534
Defining a Migration Request.....	537
Wholesale and Piecemeal Migrations.....	538
Identifying Tables to Exclude From Migrations.....	539
Configuring Custom Objects for Migration.....	539
The CMA Execution Process.....	540
Exporting a Migration.....	541
Importing and Applying a Migration.....	543
Running Batch Jobs.....	558
CMA Reference.....	558
Framework-Provided Migration Configuration.....	558
Configuring Facts.....	560
Fact Is A Generic Entity.....	561

Fact's Business Object Controls Everything.....	561
Fact Supports A Log.....	561
Real-Time Scheduler Administrative User Guide.....	561
General Options.....	561
Defining Installation Options.....	561
Base Time Zone.....	562
Formatted Address.....	562
Address Geocoding.....	562
Defining Locations.....	562
Configuring Mapping Options.....	563
Defining Geographic Areas.....	563
Setting Up Global Configuration.....	563
Service Areas.....	564
Understanding Service Areas.....	564
Different Service Areas For Different Service Types.....	564
Geographic Boundaries.....	564
Service Area Hierarchy.....	565
Configuring Service Areas.....	565
Defining Service Area Hierarchy Levels.....	565
Defining Service Area Hierarchy.....	565
Prepare Postal Code Boundary Data.....	566
Defining Service Areas.....	566
Postal Code to Service Area Mappings.....	566
Capabilities.....	567
Understanding Capabilities.....	567
Worker Capabilities.....	567
Vehicle Equipment.....	568
Limited Time.....	568
Defining Worker Capabilities.....	568
Defining Equipment.....	569
Tasks.....	569
Understanding Activity Types.....	569
Activity Business Object Controls Everything.....	569
The Activity's Host System.....	569
Service Type.....	570
Default Resource Capabilities.....	570
When Can Work Be Performed.....	570
Determine Activity Priority.....	571
Taking a Break While On Site.....	572
Advance Allocation.....	572
Activity Time Tracking.....	572
Handling Expired Activities.....	572
Assist Activity Types.....	573
Creating Activities By The Crew.....	573
Contracting Work.....	573
Understanding Service Classes.....	573
Designing Activity Types.....	574
The Lifecycle Business Objects.....	574
Activity And Assignment Business Objects Are Related.....	574
Extending Existing Business Objects.....	575
Custom Activity Business Objects.....	575
Configuring Activity Options.....	576
Defining Site ID Configuration.....	577
Host External System Configuration.....	577
Defining Remark Types.....	577
Defining Priority Profiles.....	577
Defining Work Profiles.....	578
Defining Appointment Booking Groups.....	578
Uploading Common Attachments.....	578
Defining Activity Types.....	578
Defining Service Classes.....	579
Defining Task Types.....	579
Crews.....	580
Understanding Crew Hierarchies.....	580

Configuring Crew Hierarchy.....	580
Defining Crew Hierarchy Levels.....	580
Defining Crew Hierarchy.....	580
Configuring Crew Options.....	581
Defining Mobile Worker Types.....	581
Defining Vehicle Types.....	581
Defining Crew Types.....	581
Configuring Crew Shift Options.....	582
Defining Shift Types.....	582
Defining Shift Cost Profiles.....	582
Defining Shift Related Task Types.....	583
Periods of Unavailability (POUs).....	583
Understanding POU Configuration.....	583
Defining Period of Unavailability Task Types.....	583
Defining Period of Unavailability Types.....	583
Dispatchers.....	584
Understanding Dispatch Areas.....	584
Defining Dispatcher Types.....	584
Defining Dispatch Areas.....	584
Customizing The CDI Portal.....	585
Gantt Customizations.....	585
CDI Map Customizations.....	587
Alerts.....	588
Understanding Alert Types.....	588
Alert Business Object Controls Everything.....	588
Alert Creation.....	589
Alert Creation Threshold.....	589
Valid Activity Types.....	589
Alert Distribution.....	589
Alert Priority.....	590
Alert Evaluation.....	590
Snoozing Options.....	590
Can A Dispatcher Manually Close Alerts?.....	591
Should The Alert Expire?.....	591
Should The Alert Recycle?.....	591
Designing Alert Types.....	591
When Should The Alert Be Raised?.....	591
When Should The Alert Be Closed?.....	592
Use The Base Business Objects.....	592
Defining Alert Types.....	592
Key Performance Indicators (KPIs).....	592
Understanding Key Performance Indicators (KPIs).....	592
Calculation Algorithm.....	592
Alert Based KPIs.....	593
Severity Thresholds.....	593
Designing Your KPIs.....	593
Defining KPIs.....	593
Procedures.....	594
Understanding Procedure Types.....	594
Related Entity.....	594
Questionnaire.....	594
Effective Dated Procedures.....	595
Attendance.....	595
Override Results.....	595
Designing Procedure Types.....	595
Defining Procedure Types.....	595
Timesheet.....	596
Understanding Timesheet Configuration.....	596
Timesheet Business Objects Control Everything.....	596
Work And Travel Time 'Buckets'.....	596
Designing Timesheet Types.....	596
Integration with Timekeeping Systems.....	596
Approval.....	597
Configuring Timesheet Options.....	597

Defining Timesheet Types.....	597
Defining Custom Timesheet Time Types.....	597
Enable Timesheet Functionality.....	597
Mail.....	598
Defining Mailing Lists.....	598
Defining Mail Templates.....	598
Transfer of Goods.....	598
Understanding Transfer of Goods.....	598
Depot Related Business Objects.....	599
Depot Task Types.....	599
Using Capacity Limits.....	599
Configuring Transfer of Goods.....	599
Defining Types Of Load Capacity.....	599
Defining Depot Related Task Types.....	600
Defining Depot Related Vehicle Types.....	600
Defining Depot Related Crew Types.....	600
Defining Depot Related Shift Types.....	601
Defining Depots.....	601
Contractor Management.....	601
Understanding Contractor Configuration.....	601
Contractor Users.....	601
Contracting Work.....	603
Configuring Shift Based Contractors.....	603
Configuring Capacity Based Contractors.....	604
Defining Resource Group Types.....	604
Configuring Capacity Model Options.....	604
Defining Capacity Types.....	604
Integration with Contractor Systems.....	605
Mobile Application.....	605
Understanding Mobile Devices.....	605
Mobile Platform.....	605
Mobile Device Terminals (MDT).....	606
Registration.....	606
Sending GPS Data.....	606
Managing Attachments.....	606
Data Encryption.....	607
Understanding Deployments.....	607
Deployment Types.....	607
Deployment Parts.....	607
Deployment.....	608
Out of Date Deployments.....	608
Downloading Deployments.....	608
Mobile Application Versioning.....	609
Setting Up Deployments.....	609
Understanding The Legacy Mobile Platform.....	610
Mobile Application Connection Modes.....	610
Configuration Tools.....	611
Navigation and Display.....	613
Remote Script Invocation.....	614
Device Capabilities.....	615
Mobile Data Cleanup.....	615
Configuring Mobile Devices.....	615
Defining MDT Types.....	615
Defining MDTs.....	616
Configuring Deployment Types.....	616
Defining Deployment Parts.....	617
Defining Deployment Types.....	617
Maintaining Deployments.....	617
Schedulers.....	618
Understanding Schedulers.....	618
The scheduling Process.....	618
Multiple Schedulers.....	619
Scheduler Process Types.....	619
Request Types.....	619

Schedules Are Continuously Running Batch Processes.....	620
Auto-Dispatching.....	620
The Scheduler Registry.....	620
Current Crew Location.....	622
Monitoring Schedulers.....	622
Understanding Scheduler Areas.....	623
Multiple Service Areas within a Scheduler Area.....	623
Using Optimization Areas.....	623
Scheduler Map Files.....	628
Understanding Scheduler Parameters.....	628
Parameter Types.....	628
Scheduler Configurations.....	628
Cost Control Parameters.....	629
Understanding Speed Profiles.....	633
Speed Profile Is Made Of Daily Templates.....	633
Speed Profile Area.....	633
Speed Profile Overrides.....	634
Designing Your Schedulers.....	634
Using Multiple Schedulers to Scale the Scheduling Process.....	634
Failover and Load Balancing.....	635
Business Drivers Related to Cost Control Settings.....	636
Configuring Schedulers.....	639
Configuring Scheduler Options.....	639
Defining Parameter Definitions.....	639
Defining Scheduler Areas.....	640
Defining Scheduler Configurations.....	640
Setting Up Scheduler Batch Controls.....	641
Defining Schedulers.....	641
Defining Speed Profiles.....	642
Viewing the Scheduler Registry.....	642
Integrations.....	642
Understanding Web Services.....	642
Appointment Booking.....	643
Activity Maintenance By Host.....	643
Activity Completion To Host.....	644
Activity Status Update To Host.....	645
New Activities to Host.....	645
Contractor Management.....	645
Crews Managed By Host.....	645
Depot Management.....	646
Miscellaneous.....	647
Configuring Outbound Messages.....	647
Configuration Options.....	647
Batch Control Global View.....	647
Configuration Migration Assistant (CMA).....	648
Executing Wholesale Migrations.....	648
Data that Cannot be Migrated Using CMA.....	648
BI Configuration.....	649
Table Setup Sequence.....	649
Control Data Setup.....	649
Resource Setup.....	650
Transfer of Goods Setup.....	650
Contractor Management Setup.....	650
BI Setup.....	651

Chapter 1

Business User Guide

Overview

OracleReal-Time Scheduler provides enterprise software designed to receive, schedule, assign, dispatch, and track work activities performed by your organization's resources and contractors.

The following lists the functions of each of the application's core components.

- Receive requests to perform activities at specific locations from a host application (i.e., other Oracle Utilities application, Customer Relationship Management [CRM], Enterprise Asset Management [EAM], Transportation Management System [TMS], etc.). Update the requesting host application on the status of work activities and return information collected about the work upon completion.
- Provide the ability for the host application to schedule customer appointments based on resource availability and costs.
- Schedule work activities to the available resources based on the type and the location of the work, and the required skills and equipment needed to complete the work. Manage third party contractors, either resources acting like internal resources or based on work capacity.
- Dispatch work activities to the available resources, including necessary customer / location information, work instructions, and driving instructions.
- Provide the administration function for service management to define their work, work calendars, and work resources, including skill sets, certifications, work schedules, and vacation schedules. Service management defines the scheduling rules, including cost parameters for overtime, and meeting appointments and other service level agreements. Service management also administers the users of the application and their roles.
- Provide the capability for service management and dispatchers to track work activities and manage exceptions.
- Provide a mobile application for the resources to use on mobile devices to work and complete the activities, receiving information about the work to be done (including driving directions), and collecting necessary information from the work location.

User Interface Standard Features

This section describes basic system concepts, features, and standards of the user interface.

System-Wide Standards

This document is the central reference to basic system concepts, features, and standards.

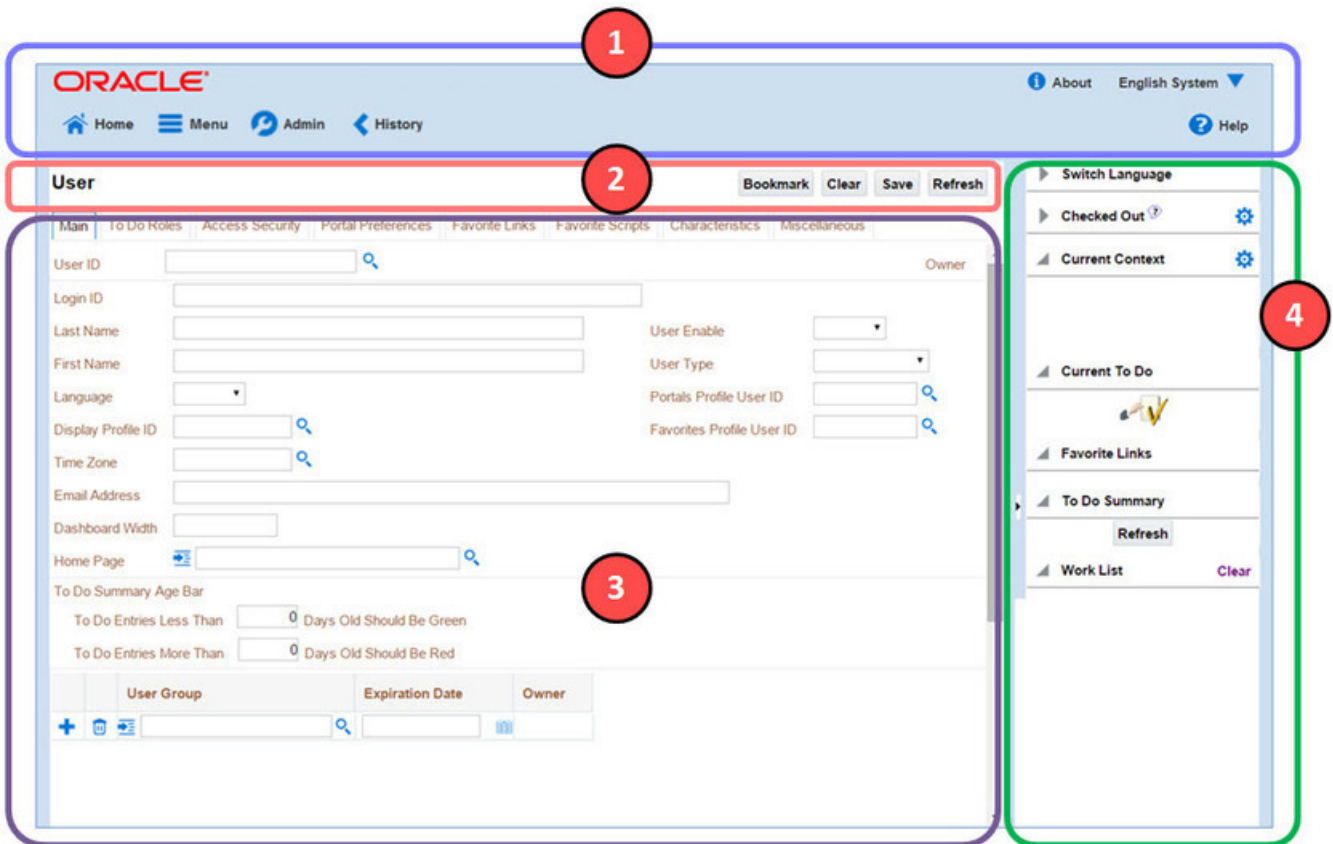
Page Components

This section serves as a central reference to the basic building blocks of all pages used in the system.

Characteristics Common to All Pages

Oracle Utilities Application Framework screens comprise the following main areas:

1. [The Application Toolbar](#)
2. [The Page Title Area](#)
3. [The Object Display Area](#)
4. [The Dashboard Area](#)




NOTE: The look and feel of the application may be modified after the product is installed. See [Custom Look and Feel Options](#) for customization information, including how to change colors, fonts, and other system features. The information provided in this document represents features and functionality available only in the delivered product.

The Application Toolbar

This section describes the features available on the application toolbar.

Home Button

Click the **Home** button,  **Home**, on the Application Toolbar to open your home page. Your home page is defined in [User Preferences](#).

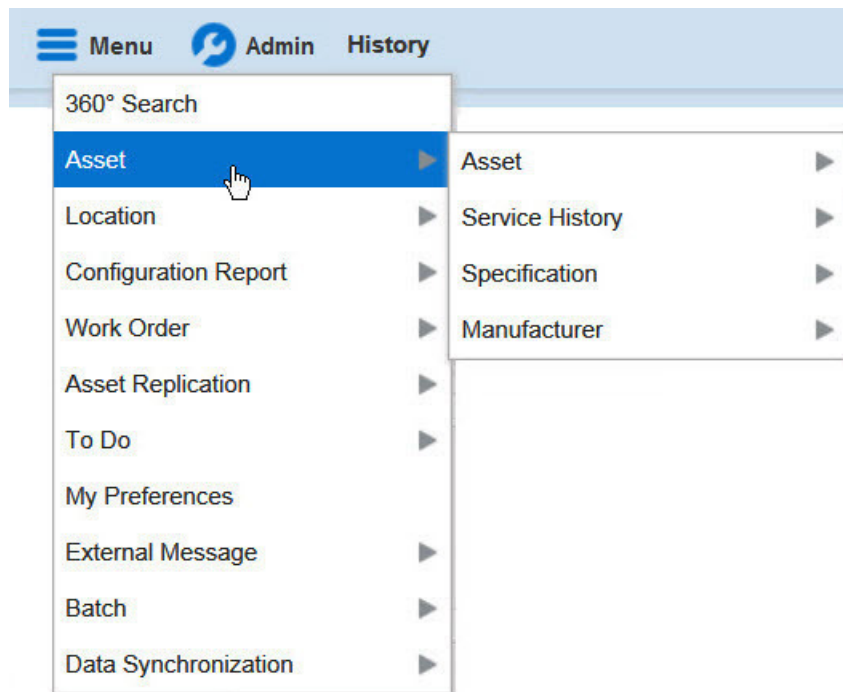
Keyboard shortcut: Alt+O

Menu

Menu is available in the application toolbar to help you navigate to the different pages of the system

The keyboard shortcut to open the Menu: Ctrl+Alt+M

The menu list is organized by functional area. Clicking the Menu button displays each functional area. Clicking a functional area, in turn, displays a submenu that contains pages within that area.



The pages within each functional area *typically* have two options: Add and Search. If the menu item does not have an Add or Search option, select the menu item itself.

How the Add and Search option behaves depends upon whether the maintenance page is fixed or portal-based.

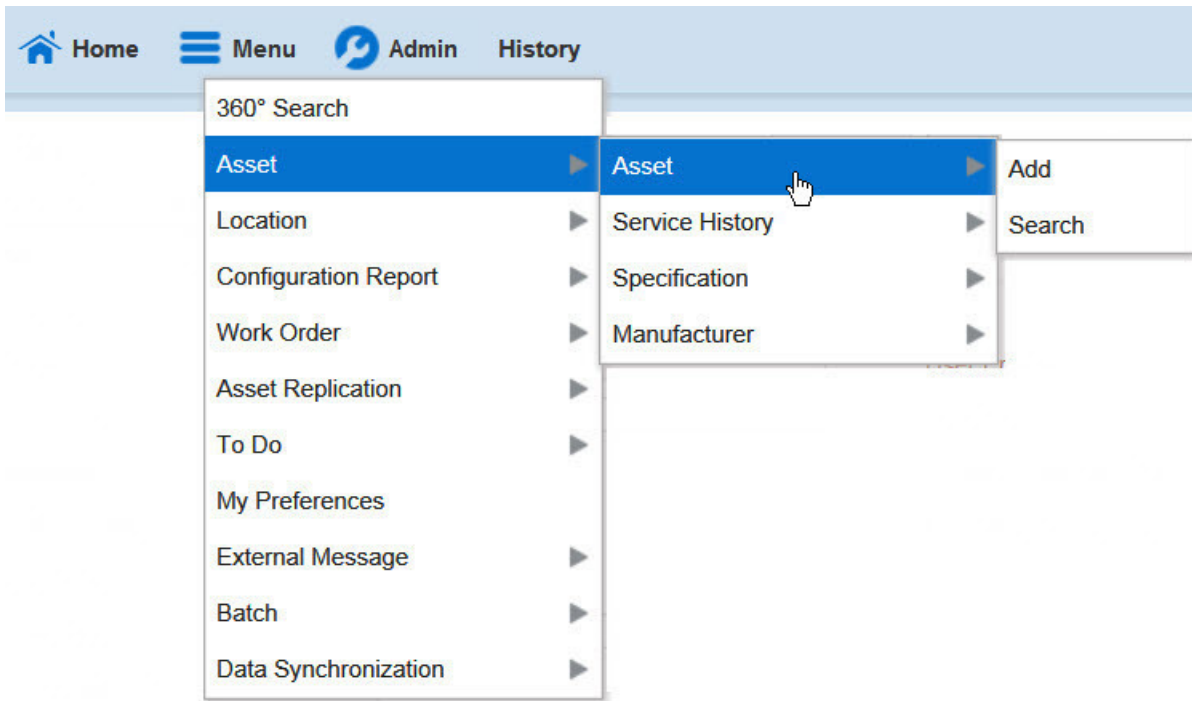
Fixed:

- **Add:** Displays the maintenance page with empty fields so that the user can complete the information and save the entity.

- **Search:** Displays a pop-up search window for the user to enter the search criteria and select the entity to display. Once the entity is selected, the user is taken to the maintenance map with the data populated in the fields.

Portal-Based:

- **Add:** Either navigates to a page where the user can select the entity type or business object, or directly to the input map where the user can enter and save the entity.
- **Search:** Displays a search portal where the user can enter the search criteria and select the desired entity. Once the entity is selected, the user is typically taken to a maintenance portal or the information is broadcast to other zones within the same portal.



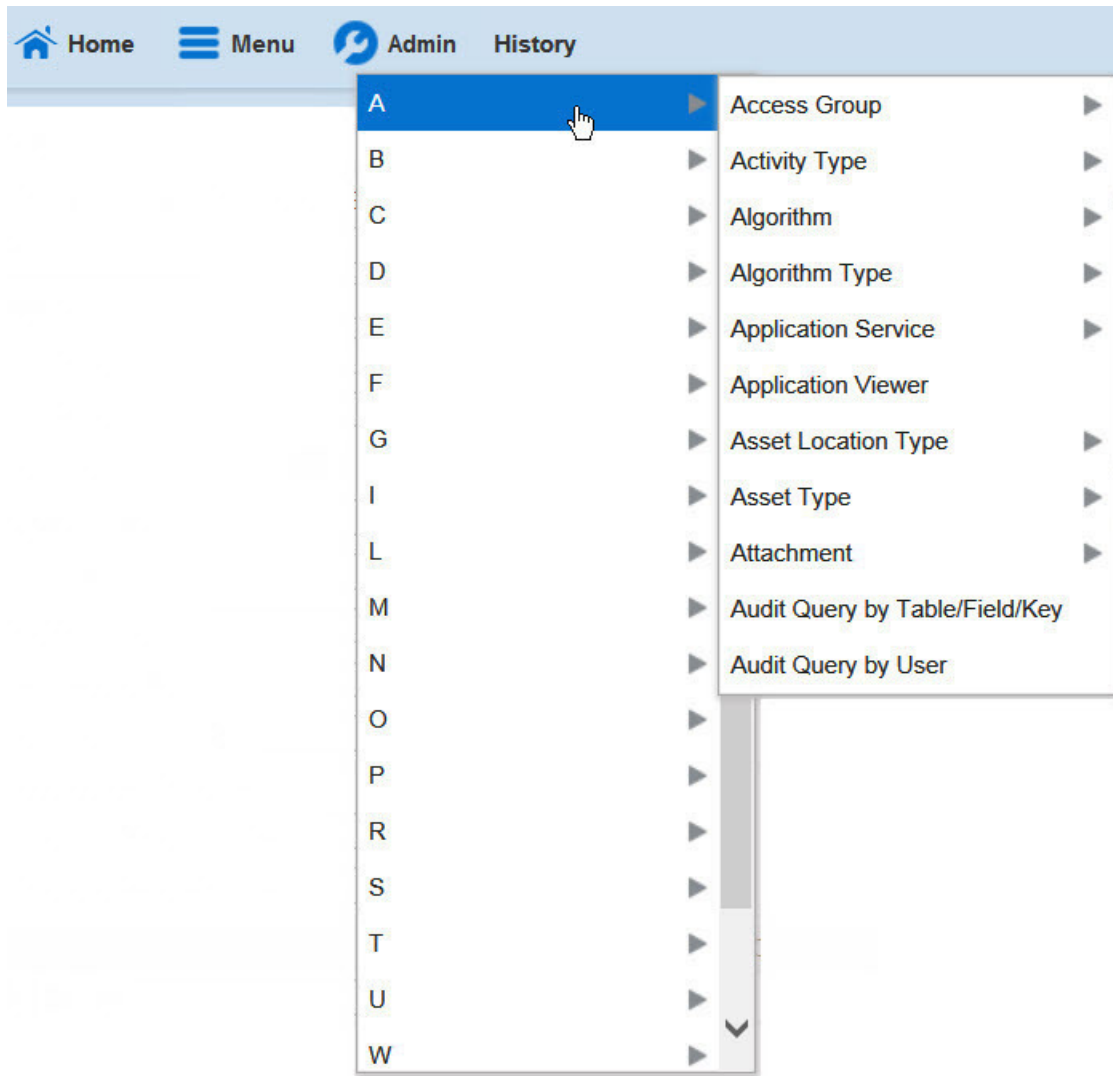
Admin Menu

The Admin dropdown is available in the application toolbar to help you navigate to the different pages of the system.

The keyboard shortcut to open Admin: `Ctrl+Alt+A`

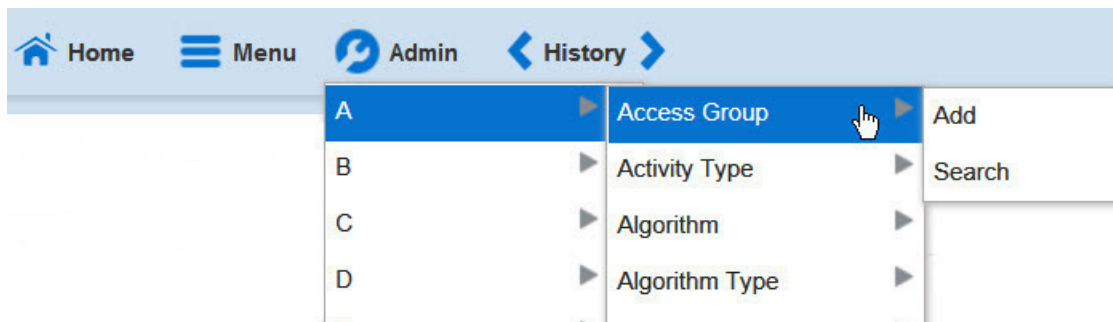
Depending on how your implementation has configured the Admin menu list, it may be organized either by functional area or alphabetically.

Clicking the Admin button displays each functional area or alphabetical list. Clicking one of the options, in turn, displays a submenu that contains pages within that area. The following is an example of the Admin menu organized alphabetically.



NOTE: Refer to [Installation Options](#) for information about the admin menu configuration options.

The menu lines displayed for each functional area *typically* have two options: Add and Search. If the menu item does not have an Add or Search option, select the menu item itself. See [Menu](#) for more information about these options.



Back Button

Click the **Back** button on the application toolbar to open the previously visited page.

Keyboard shortcut: Alt+B.



The **back** and **forward** navigation buttons, as well as the **History** list, appear in the application toolbar only when page use warrants their appearance.

NOTE: Under certain circumstances, such as revisiting a page that was left unsaved after being loaded with default data, a page may not appear as it did when you left it.

History Button

Click the **History** button to display a list of previously visited pages. When you click on an entry in the list to return to an earlier page, all items above the selected page are removed from the list.



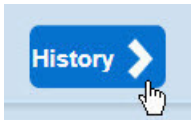
The **back** and **forward** navigation buttons, as well as the **History** list, appear in the application toolbar only when page use warrants their appearance.

Keyboard shortcut: Ctrl+Alt+H

Forward Button

Click the **Forward** button to return to the page that was most recently dismissed.

Keyboard shortcut: Alt+G.

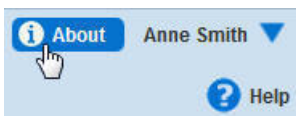


The **back** and **forward** navigation buttons, as well as the **History** list, appear in the application toolbar only when page use warrants their appearance.

NOTE: Under certain circumstances, such as revisiting a page that was left unsaved after being loaded with default data, a page may not appear as it did when you left it.

About Button

Click the **about** button to display a window that describes the current and release information for your version of your product.



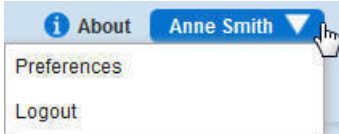
The about window shows a variety of information including:

- Who the system is licensed to

- The user ID that is logged in
- The release ID of the product
- The release ID of the framework used to construct the product
- The release ID of any customer specific modifications (this comes from the [installation record](#))

User Menu

The User menu is available in the application toolbar to provide access to user-specific options.

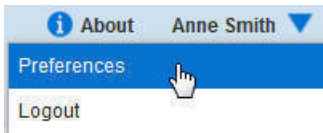


Through this dropdown, the user can access their [User Preferences](#) or [Logout](#) of their current session.

NOTE: Configurable Menu. Additional menu entries may be visible in your implementation as this menu configuration allows for additional options to be defined.

Preferences

Click the **Preferences** menu item in the User Menu to view or modify your settings in a form that appears in the **Main** tab in the object display area.



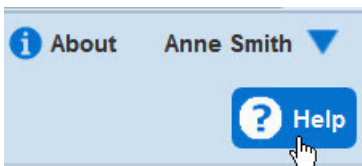
Logout

Click the **Logout** menu item in the User Menu to complete your current session of the application. Clicking this item logs the current user out and displays the Welcome page so that another user can login.



Help Button

Click the **Help** button in the Application Toolbar to open context-sensitive Help in a new browser window.



Keyboard shortcut: `Alt+F1`

Page Title Area

The Page Title Area displays the page title and the page actions toolbar.

For certain pages in the system, the page title may be followed by an information string. It is only applicable to portal-based pages that display a single record. This information is only visible if the portal has been configured to display the primary key's information string. Typically only "stand alone" maintenance portals are configured to show the information string, whereas "all-in-one" portals are not configured this way.

FASTPATH: Refer to [Common Base Portals and Zones](#) for more information about standard base product portals.

NOTE: If an information string is long and would interfere with a page actions button, the information string will automatically be truncated and suffixed by an ellipsis, "...", indicating that more information is available. In this situation, the missing text can be viewed if the user hovers their mouse over the string.

The Page Actions Toolbar displays standard control options, such as Save and Duplicate. More detail about the actions that may appear are described in the following section.

Page Actions Toolbar

In the Page Title Area, control functions appear in the Page Actions Toolbar; such as, Clear, Save, Duplicate, Delete and Refresh buttons.

NOTE:

Buttons in the Page Title Area appear as the function is available to the user; for example, the Delete button will not appear if the user has made any changes to the page's content, or does not have the necessary security rights for this action.

In addition, other functions that are specific to a portal-based page may appear in the toolbar, if configured. For example, the ability to navigate to the Search zone for a specific maintenance object may be visible on the maintenance portal.

This section describes the standard features that are available in the Page Actions Toolbar.

Bookmark Button

Click the Bookmark button on the Page Actions Toolbar to save the current page and context to the bookmarks list.



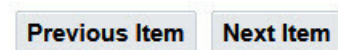
NOTE: Secured. This button is secured using the application service **F1-BOOKMARKS**. Users that do not have security for the application service will not see the Bookmark button (or the corresponding dashboard zone).

After clicking the Bookmark button, you are prompted to provide a name for the bookmark. After saving, it is visible in the [Bookmarks Zone](#), allowing you to return to this page with this context loaded with one click.

NOTE: Unsaved data will not be included in the bookmark. If you view a record and make changes to data without saving the changes and then bookmark the page, the unsaved changes are not included in the bookmark information.

You may change the name of the bookmark or change the sequence of the bookmarks by navigating to the [User — Bookmarks](#) in the user preferences.

Next/Previous Item Buttons



The (**Previous Item** and **Next Item** buttons appear in the Page Actions Toolbar under these conditions:

- You access a fixed page, meaning that it is not portal-based.

- You use a search page to display an item.
- There are multiple items in the search results list *below* or *above* the selected item.

If, for example, you enter search criteria of "Smith" on the User Search page, you might see the following results:

- Smith, Albert
- Smith, Peter
- Smith, Zane

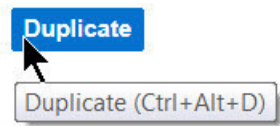
If you select **Smith, Peter**, Peter's user information is displayed on the User page. If you then click the **Next Item** button, **Smith, Zane** will be displayed. In addition, because **Smith, Zane** is the last entry in the search results, the **Next Item** button is no longer needed and will not appear in the Page Actions Toolbar.

If you then click the **Previous Item** button, **Smith, Peter** will be displayed; click again and the **Smith, Albert** record will appear, and, because **Smith, Albert** is the first entry in the search results, the **Next Item** button is no longer needed and will not appear in the Page Actions Toolbar.

Keyboard shortcuts: Alt+P, Alt+N.

Duplicate Button

Click the **Duplicate** button in the Page Actions Toolbar to create a duplicate of the currently selected object.



Before the new object is added to the database, a dialog appears in which you must enter the unique identifier for the duplicate object. When you click OK, the system creates the record by copying the original record. It may be helpful to think of the duplication function as the equivalent of a "save as" action.

Keyboard shortcut: Ctrl+Alt+D.

NOTE: Copying ancillary information for a record sometimes requires confirmation during a duplication request. If so, you may also be asked if you also want to duplicate ancillary information. If, for example, you duplicate a user group, besides supplying the ID of the new user group, you are asked if all users in the original user group should be linked to the new user group.

This button does not appear in the Page Actions Toolbar when:

- Duplication is not supported for the current object.
- You are on a portal based page. (For these types of pages, if duplication is supported, the actions are defined in a zone on the portal.)
- You have made changes to the object without committing the changes to the database by clicking Save. (After making changes, you must either discard them using the **Refresh** button, or save them.)

Delete Button

Not all transactions have the capability to be deleted. Deletion is related to the business impact of the action. In some transactions, Delete is present for some scenarios and unavailable for others; for example, you may not be able to delete a record at a certain status level.

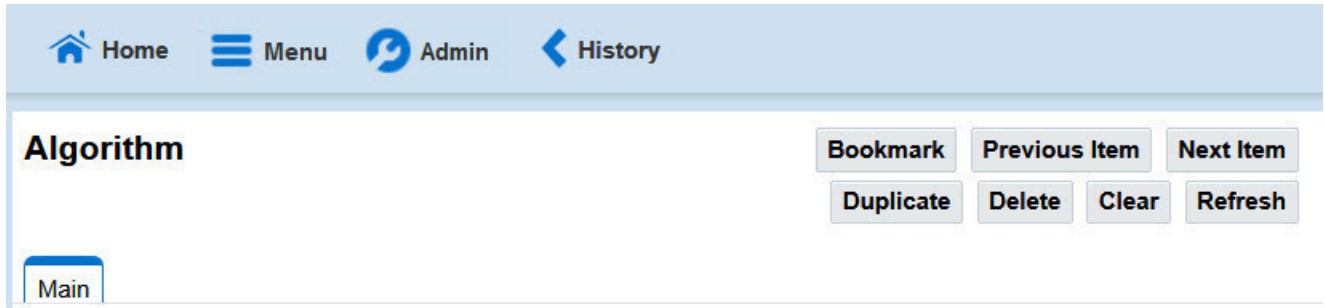
NOTE:

If you make a change to an object and do not save the changes to the database, you cannot delete the object until you either save the changes or refresh the object with the contents on the database.

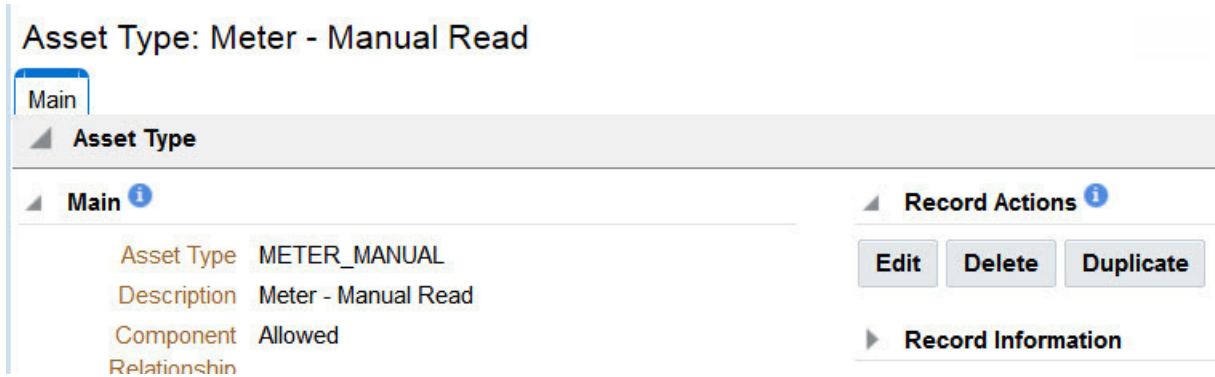
Before you delete an object, the system verifies that the object is not referenced anywhere in the system. If it is, you cannot delete it. If, for example, you attempt to delete a user, the system ensures this user has not been referenced in a user group or on a To Do.

Depending on the type of page that you are using, the option to delete may vary.

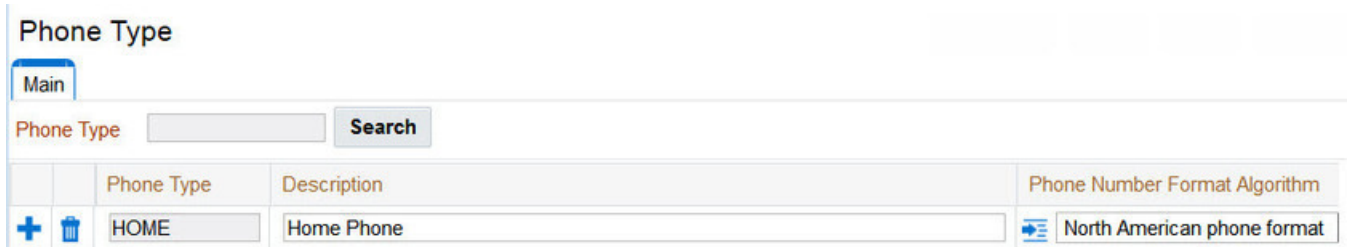
- Fixed pages display a Delete button in the page actions toolbar when the option is available.



- Portal-based pages display the Delete button in the object-related map.



- List zones on All in One portals commonly display the Delete button (garbage can icon) next to the entity in the search results.



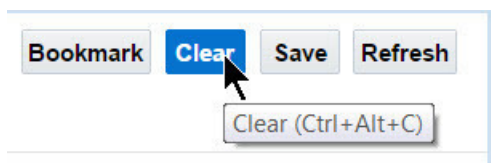
In all cases, before the object is deleted, a confirmation dialog is displayed.

CAUTION: A timeout situation could occur as the system attempts to verify the use of the deleted object if the object is referenced in many places *and* involves large volumes of production data. If this timeout situation occurs, you must work with your database administration group to delete the offending object.

Clear Button

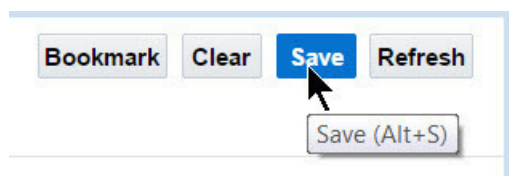
Click the **Clear** button to remove the values from the currently displayed object's fields. This action is typically used to clear input fields when adding a new object.

Keyboard shortcut: Ctrl+Alt+C.



Save Button

Clicking the **Save** button saves any changes you have made in the data on the current object.



Keyboard shortcut: **Alt+S**.

The button does not appear unless:

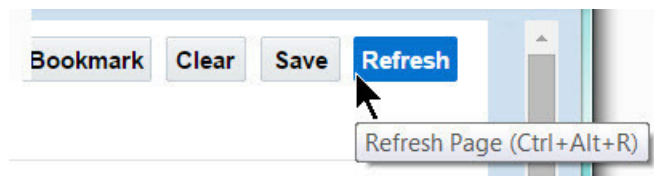
- You enter a page in add mode and default values are loaded.
- You make changes to an object after it is displayed (note that you must tab or click out of the field before the Save button appears).

Refresh Button

Click the Refresh button to reload the values on the page with those from the database.

CAUTION: All unsaved changes are discarded.

Keyboard shortcut: **Alt+R**.



Object Display Area

The Object Display Area contains the details of the object(s) currently being viewed or maintained. For example, if you display a portal, its zones are displayed in this area.

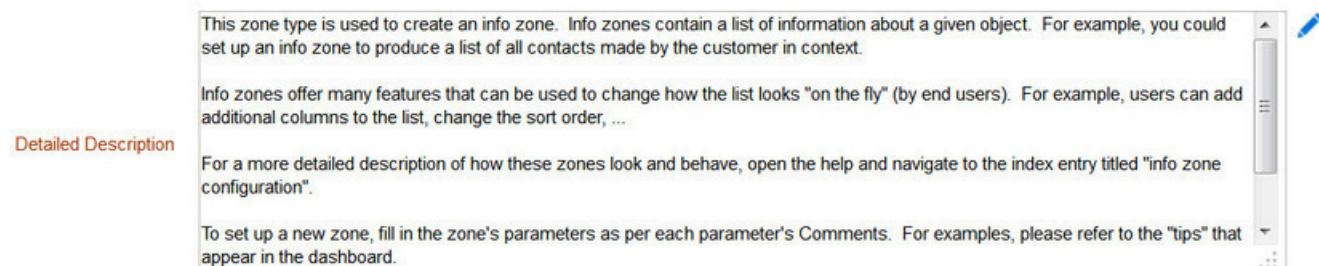
Because many of the page components follow Internet standards of operation, it seems redundant to describe their function. Instead, this section describes special features, or enhancements, that appear within the framework.

Text Box

The following is an example of a basic text box:

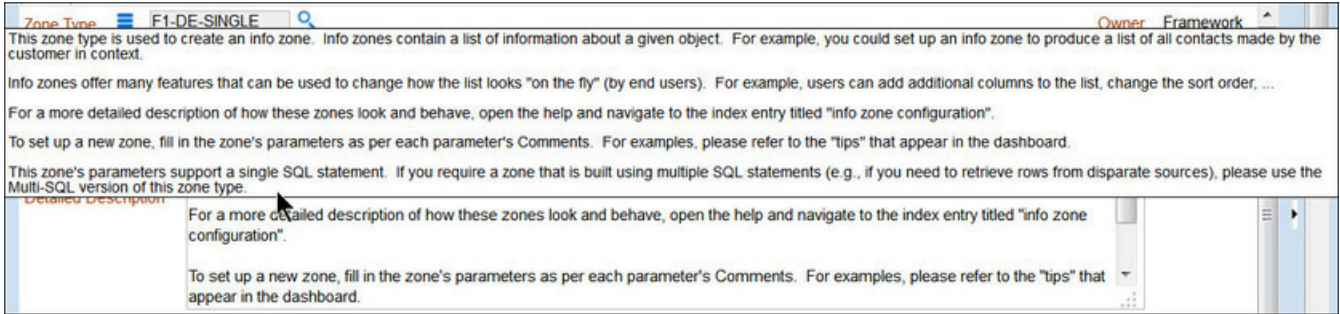


If the text box is a large field, the text box may be configured to enable scrolling. In this situation, an [Expanded Edit](#) icon, pencil, will appear adjacent to it.

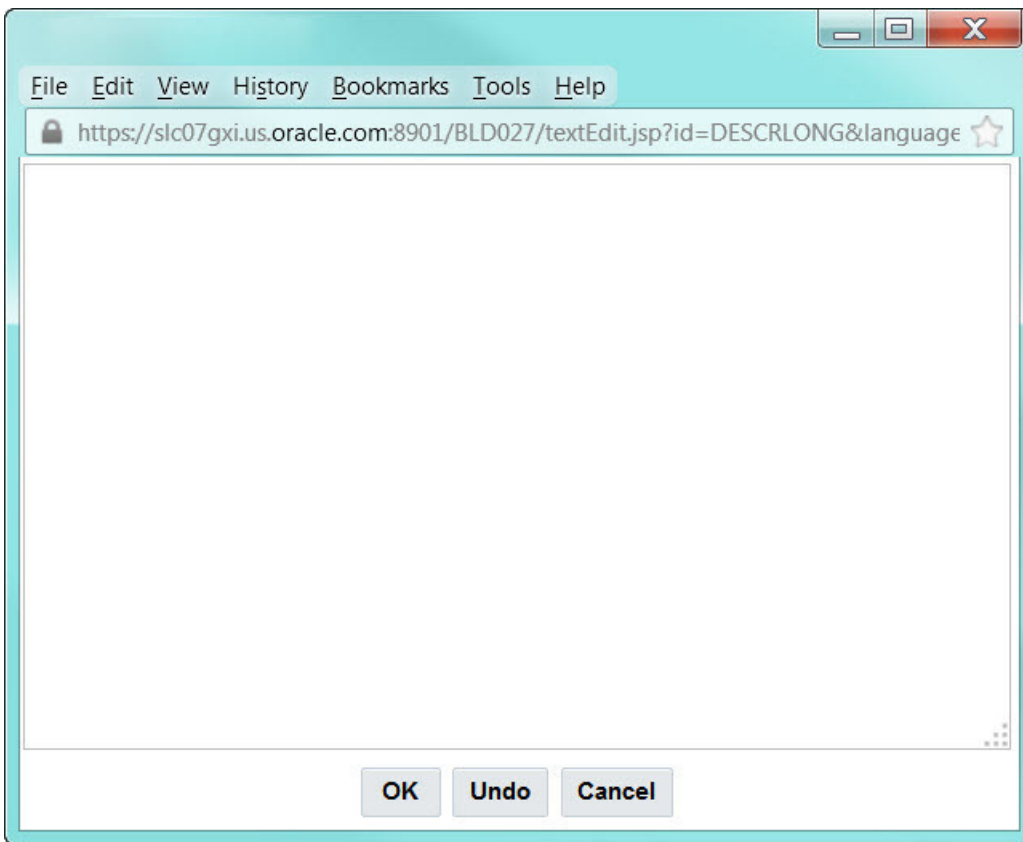


Additional capabilities are provided to display and edit data in these text boxes:

- If the field contains more text than is visible, a pop-up that displays the full text appears when you hover your mouse over the text box. The text in the pop-up can be selected and copied.



- If you click the Expanded Edit icon, an editable pop-up window appears. If the field is not protected as a result of business rules for the page, you may modify the text in this window.



NOTE:

Special characters are converted to equivalent standard characters when entered in a text box:

Entered Character Description	Unicode	Converted Character Description
Curly left double quote.	U+201D	Straight double quote.
Curly right double quote.	U+201C	Straight double quote.
Curly left single quote.	U+2019	Straight single quote.
Curly right single quote.	U+2018	Straight single quote.
En dash.	U+2013	Simple hyphen.

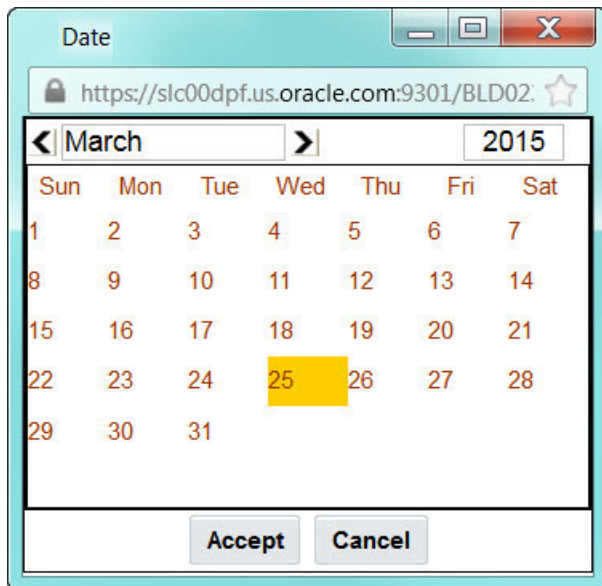
Em dash.	U+2014	Simple hyphen.
Ellipse	U+2026	Three dots.
Middle dot (bullet / centered dot)	U+00B7	Asterisk.
Fraction 1/4 (using 1 character).	U+00BC	1/4 (using three separate characters).
Fraction 1/2 (using 1 character).	U+00BD	1/2 (using three separate characters).
Fraction 3/4 (using 1 character).	U+00BE	3/4 (using three separate characters).

Date Only Field

When only the date is required, the Date field looks as follows:

The image shows a text input field with the label "Expiration Date" in orange text above it. To the right of the input field is a small blue calendar icon.

If you click the calendar icon next to the date field, or you double-click inside the date field, the following window appears:



If the date field was blank, the current date is highlighted when the window first appears. If the date field contained a date, it is displayed.

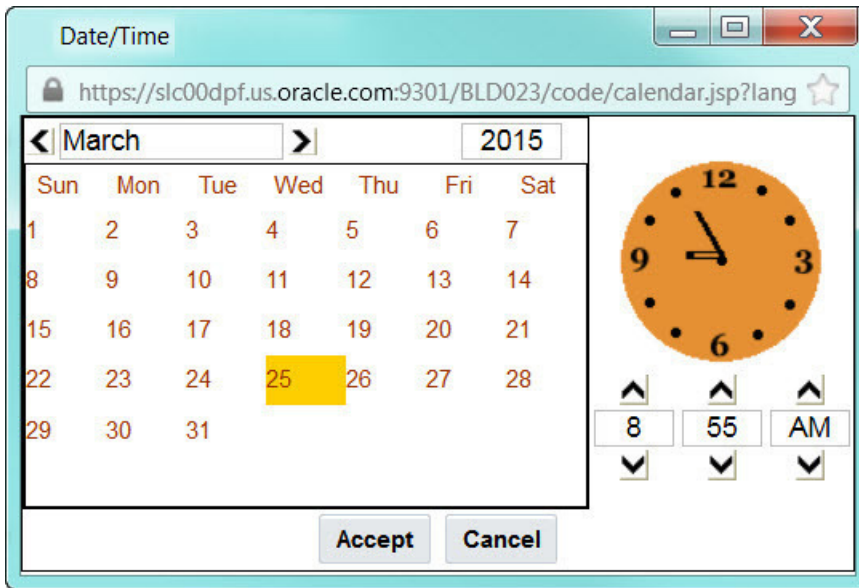
You can double-click the desired date, or you can select a date and click Accept to close the date window and populate the date field.

Date and Time Field

When both the date and time are required, the Date and Time fields typically look as follows:

The image shows a text input field with the label "Desired Execution Date/Time" in orange text above it. The field contains the text "03-25-2015 / 08:55AM".

If you double-click or press the Enter key while the cursor is in a Date and Time field, the following window appears:



Select the date and use the toggle buttons (or enter) the time and then click **Accept** to populate the original fields.

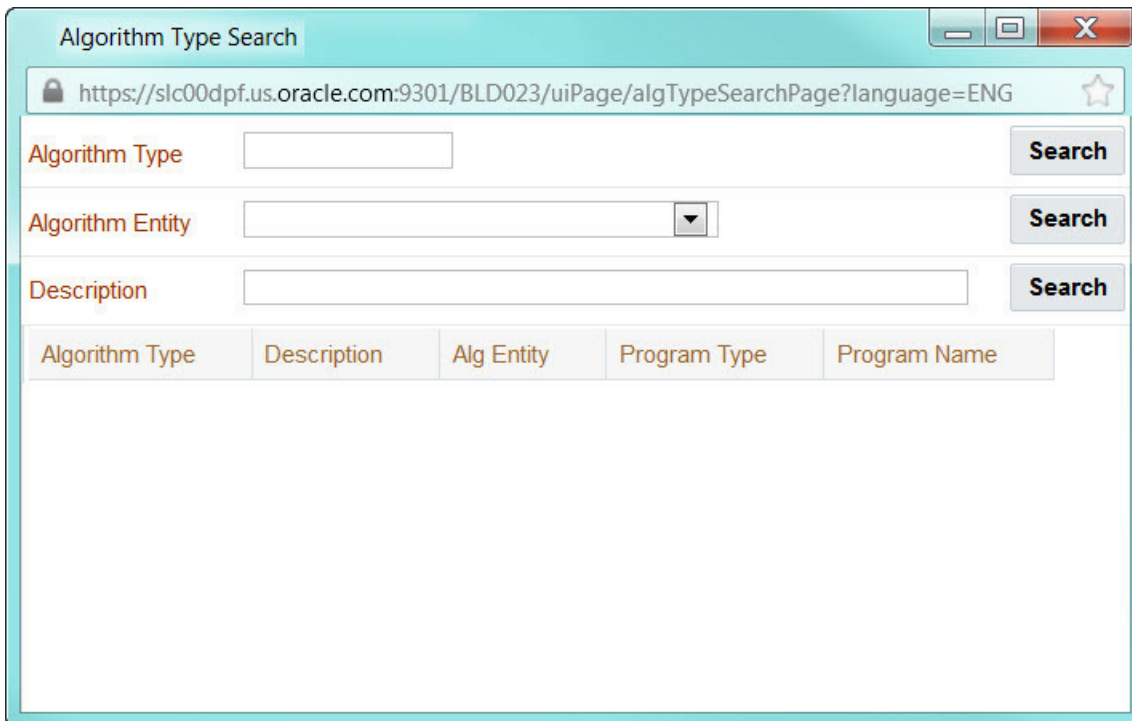
NOTE: The user cannot select and move the hands on the clock to set the time.

Search Button

The search button appears primarily as a magnifying glass.



When using a pop-up search window, the search button can appear as a button with its function labeled.



You can have the system search for an item that matches the data you entered using any of the following methods:

- Enter a value in the search field and press **Enter** while the cursor is in the search field. The system assumes you want to search for the value entered and pre-populates it in the search window's search criteria. If a unique match can be found, the system automatically populates the field with details corresponding to the record found. If a unique match cannot be found (either no matches or multiple matches), a search page opens containing selectable items.
- Click the Search button without entering a criteria. This will typically cause a pop-up search window to appear so that you can further refine your search.
- Enter a value and tab out of the search field. The system looks for a value that exactly matches what you entered. If it cannot find such a value, the search dialog will not be presented.

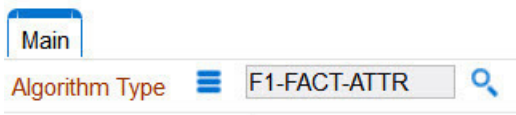
When an item is selected in the search page, the search page closes and the input field is populated with the value selected.

NOTE: Sorting columns within the search grid. You may click on the column heading of columns in a search grid to cause the contents of the grid to be sorted. Refer to [Sorting Functionality](#) for more information.

Context Menu Button

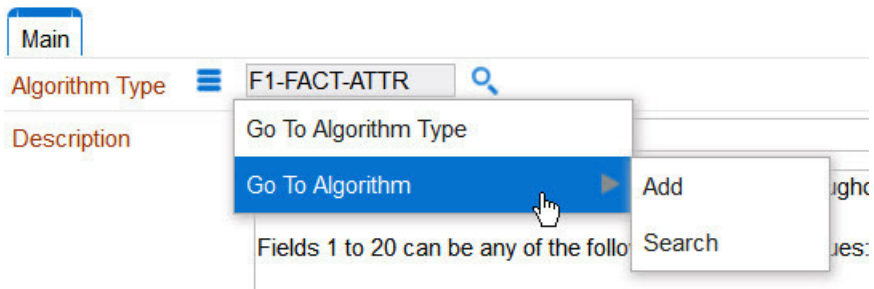
The Context Menu button is a small version of the Menu button from the Application Toolbar. An example is shown below:

Algorithm Type




Context menus exist to help you navigate to other pages and populating the page with the current context (by passing the ID of the related object). In addition, the menu items offered in a Context Menu differ depending on the entity that the menu is related to; for example, when you click the Algorithm Type context menu, a pop-up menu similar to this appears:

Algorithm Type




Different context menus exist for each of the major objects in the system; for example, in Customer Care and Billing, the context menu for Premise contains premise-oriented menu items, whereas the context menu for Account contains account-oriented menu items.

Go To Button


Clicking a Go To button, , opens the page for the related object - it functions like a hyperlink.

	User Group	Expiration Date	Owner
	<input type="text" value="ALL_SERVICES"/> System User Group	<input type="text" value="12-31-2100"/>	Framework
<input type="button" value="Go To User Group"/>			

Broadcast Button

Some zones have been configured to "broadcast" unique identifiers to other zones within the portal; if so, you'll see a broadcast tower icon, , in the first column. Clicking the icon causes specific zones to reload — these zones that have been configured to rebuild when the unique identifier is received. For example, you might have configured a zone to display the image of a bill when a bill ID is broadcast from an info zone.

Work Lists


The work list buttons can appear in a zone's column heading, .

Clicking the icon results in populating the dashboard [Work List Zone](#) with the entire column. The user may use the Work List zone to easily process entries one at a time without returning back to the source query zone. The work list stays populated until the user overlays the work list by clicking a different column's icon.


When a user clicks on a work list entry in the dashboard, the system behaves as if the user clicked on the hyperlink in the zone. In addition, the system adds a check mark next to the item showing that the user has "worked on" or visited the entry.

NOTE: The Work List zone is populated only upon user request.

Scroll Bars with Dynamic Height

Sometimes the height of a grid isn't sufficient to display a grid's rows. When this occurs, a vertical scroll bar appears and an "expand button" appears above this bar, . For example, the following grid is not high enough to show all of the available dashboard zones:


Dashboard - 10 of 10 zones selected for display.

Zone	Display	Initially Collapsed	Sequence	Refresh Seconds		Security Access		
Switch Language	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text" value="20"/>	<input type="text" value="0"/>	Yes
Bookmarks	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	Yes
Checked Out	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	Yes
Current Context	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	Yes

If you want to view all of the person's phone numbers without using the vertical scroll bar, you can click the expand button. When clicked, the height of the grid expands to show all rows in the grid:

Dashboard - 10 of 10 zones selected for display.

Zone	Display	Initially Collapsed	Sequence	Refresh Seconds		Security Access
Switch Language	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="text" value="20"/>	<input type="text" value="0"/>	Yes
Bookmarks	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="text" value="0"/>	<input type="text" value="0"/>	Yes
Checked Out	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="text" value="0"/>	<input type="text" value="0"/>	Yes
Current Context	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="text" value="0"/>	<input type="text" value="0"/>	Yes
Current To Do	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="text" value="0"/>	<input type="text" value="0"/>	Yes
Favorite Links	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="text" value="0"/>	<input type="text" value="0"/>	Yes
Favorite Scripts	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="text" value="0"/>	<input type="text" value="0"/>	Yes
Favorite Searches	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="text" value="0"/>	<input type="text" value="0"/>	Yes
To Do Summary	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="text" value="0"/>	<input type="text" value="0"/>	Yes
Work List	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="text" value="0"/>	<input type="text" value="0"/>	Yes

Notice, after the grid is expanded, the grid's vertical scroll bar disappears and the button in the upper corner becomes a Collapse Button, . If you click the collapse button, the grid will return to its original height.

NOTE: Search grids. Expand all functionality is not available in search grids.

Sorting Functionality

Grids have a heading row that contains labels for the columns. By clicking a column heading, you cause the rows to be sorted in the order of the selected column. By clicking on a column more than once, you cause the rows to be sorted in reverse order. Clicking the heading a third time removes the sorting function and returns the grid to its default display organization.








The following shows how information look after organizing the grid, clicking the column heading, by algorithm type (notice the icon in the column name).

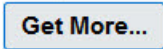
Algorithm Code	Description	Algorithm Type ▲
F1-ACTIVEWS	Web Service - Verify Active Operations	F1-ACTIVEWS
F1-ATCHM-INF	Attachment Information	F1-ATCHM-INF
F1-BAT-LSDEF	Batch Level of Service Algorithm	F1-BAT-LVSVC
F1-BNDL-INFO	Bundle Information	F1-BNDL-INFO
F1-BNEXVALID	Bundle Export Validation	F1-BNEXVALID
F1-BOSTATFS	Take BO Status/Reason Final Snapshot	F1-BOSTATFS
F1-BUNEXPBUN	Prepare bundle XML	F1-BUNEXPBUN
F1-BUNIMPAPP	Apply a bundle	F1-BUNIMPAPP
F1-BUNLOGUSR	Log user updating an export bundle	F1-BUNLOGUSR
F1-CMPSNAPSR	Compare Sync Request Snapshots, populate status reason if discarded.	F1-CMPSNAPSR
F1-COMPSNAPS	Compare Initial and Final Snapshots	F1-COMPSNAPS
F1-DECRT-INF	Decision Report Information	F1-DECRT-INF
F1-FINALSNAP	Take Final Snapshot	F1-FINALSNAP

NOTE: Sorting on dropdown values. If you attempt to sort a column that contains a dropdown value, the information is sorted in the order of the unique identifier associated with the dropdown value, not the description you see on the screen; for example, if you sort on a column containing unit of measure, the information will be sorted in the order of the unit of measure code, not the description that appears in the drop down. This means that the resultant sort will appear incorrect and therefore we discourage you from doing this.

Get More Functionality

Grids that can potentially contain a large number of rows have special processing that we refer to as Get More functionality. We'll use an example from Oracle Utilities Customer Care and Billing to explain - consider the grid that contains all payments that have been added to a cash drawer. By the end of the day, this grid could contain hundreds of rows (one for each payment remitted during the day). Rather than show every payment in the grid, the system retrieves the first X payments (where X varies from grid to grid, and from release to release) and appends a Get More button to the bottom of the grid as illustrated below:

	Tender Amount	Tender Type	Account	Check 
	\$123.90	Cash	4172150879 - Rodehaver, Gary - Residential - M-D1	
	\$69.76	Cash	5983360951 - Nicholson, Edvard Paige - Res - M-D1	
	\$10.00	Cash	4438465447 - Richie, Kate - Residential - - ALOP	
	\$10.00	Cash	2609586025 - Richie, Harry - Residential - - ALOP	
	\$10.00	Cash	8593641982 - Richie, Girlie - Residential - ALOP	
	\$10.00	Cash	0401455188 - Richie, Iris - Residential - - ALOP	





When you click the **Get More** button, the system appends the next X rows to the bottom of the grid.


NOTE: The absence of a "get more" button. If no button appears at the bottom of the grid, this means you are seeing all of the rows.

NOTE: Search grids. The Get More function is not available in search grids.

Adding and Removing Rows Using Arrows





In some editable grids, a new row can be added to another editable grid (the grid on the left) by clicking the Add item button, , of the desired row in the grid on the right:

	Sequence	Description		Sequence	Description
	10	Winter kWh charge, first 500		30	Winter subtotal
	20	Winter kWh charge, remaining		60	Summer subtotal
	40	Summer kWh charge, first 300			
	50	Summer kWh charge, remaining			
	70	Employee discount			

When the Remove item button is clicked, , the row is transferred from the grid on the right to the grid on the left. This change will not be committed to the database until you save the object being maintained

Reorganizing Items Using the Up and Down Arrows

Occasionally, rows can be repositioned in an editable grid by clicking up and down arrows. The following is an example of such a grid:

		Merge Type	Sequence	Step Type	Description
	 	Original	10	Edit data	Edit data - Perform State Transition
	 	Original	20	Edit data	Edit data - Perform Main Processing
	 	Original	30	Label	Label - End of script

Notice the following about such a grid:

- The up arrow is dim in the first row. This is because it can't be moved further to the top.
- The down arrow is dim in the last row. This is because it can't be moved further down.
- The up and down arrows are usable in all other rows. Each time you click one of these arrows, the respective row is moved up or down one row.

NOTE: Reminder. When you move rows into a grid, the system does not apply the change to the database until you click the Save button (which commits the entire object to the database).

Trees

Trees are used to illustrate complex relationships between objects. The following is an example from Oracle Utilities Customer Care and Billing of a page with a tree that is expanded:



The following points describe how to use trees:

You can expand and contract the nodes in a tree by clicking on the icon that prefixes the node (you can also click the + and - button to expand and contract a node).

NOTE: Expand all. If you press `Ctrl` while clicking on an icon that prefixes a node, all child nodes (and their children) expand.

You can click the context menu button to cause a context menu to appear. You can navigate to any of the pages that are presented on this menu. Refer to [Context Menu Button](#) for more information.

You can click on the information that appears in an underlined node to drill into the page in which the object is maintained. For example, if you click on the account information, you will be transferred to the Account - Main page.

NOTE: Important! If you make changes after expanding a tree's nodes (e.g., by drilling down on a Premise and changing the premise's address), the tree will contract to the first node if you return to the page.

Nodes in trees fit into two broad categories:

- **Information nodes.** Information nodes highlight the existence of other data. For example, on the Control Central - Bill Payment Tree, a node exists to highlight the existence of new financial transactions that were created since the last bill was completed.
- **Data nodes.** Data nodes show information about objects in the system. For example, data nodes exist on the Control Central - Bill Payment Tree that show bills and payments.

Required Field Indicator

An asterisk (*) is used to indicate that a field is required. The system determines this based on the field's configuration. When a field becomes required because of an algorithm and not the field's configuration, however, the system will not add an asterisk.

NOTE: The field indicator only appears on portal-based pages. Older, fixed pages will not include this designation.

NOTE:

If a page includes a grid, fields within a row may be marked as required.

Operations 1			
	Operation Name	Schema Type	Schema Name
+ *	<input type="text"/>	<input type="text"/>	<input type="text"/>

In this situation, the field(s) within the grid are **only** required if the row is being populated.

Accordion Windows

Accordion windows are used when the data held in an object's rows does not comfortably fit into a grid.

Each row in an accordion has a header that summarizes a row's details. When you click a header, the row expands to show the details. The following is an example of an accordion where all headers are collapsed (meaning that only summaries of the rows' details are shown):

Steps	All 3 displayed	[Expand All]	[Collapse All]	Copy
<input type="checkbox"/> + (10)	Edit data - Get the Physical BO for the MO			
<input type="checkbox"/> + (20)	Edit data - Get the prime keys of the BO			
<input type="checkbox"/> + (30)	Edit data - Read the BO			

When you expand a header, its details are exposed. For example, the following shows the same accordion after the third row is expanded:

Steps	All 3 displayed	[Expand All]	[Collapse All]	Copy
<input type="checkbox"/> + (10)	Edit data - Get the Physical BO for the MO			
Step Sequence	<input type="text" value="10"/>			
Step Type	<input type="text" value="Edit data"/>			
Text	<input type="text" value="Get the Physical BO for the MO"/>			
Edit Data Text	<pre> move "parm/mo" to "F1-RetrieveMOOption/input/mo"; move 'F1PB' to "F1-RetrieveMOOption/input/moOptionType"; move 'false' to "F1-RetrieveMOOption/input/isMultiple"; invokeBS 'F1-RetrieveMOOption' using "F1-RetrieveMOOption"; </pre>			
<input type="checkbox"/> + (20)	Edit data - Get the prime keys of the BO			
<input type="checkbox"/> + (30)	Edit data - Read the BO			

NOTE: You must expand to update. In order to update a row in an accordion, you must expand it.

Row Summary and Chunk Management

At the top of an accordion you'll find a summary of its rows. The format of this information differs depending on the number of rows and the size of the accordion:

- If the accordion contains every row, a summary of the total number of rows is displayed.
- If the accordion contains a subset of rows, you are presented with a summary of the rows AND you are provided with options to display different "chunks" of rows.

- Click **First** to display the first "chunk" of rows (the size of a chunk differs depending on the accordion). This option is dimmed if the first chunk is in the accordion.
- Click **Previous** to display the previous chunk of rows. This option is dimmed if the first chunk is in the accordion.
- Click **Next** to display the next chunk of rows. This option is dimmed if the last chunk is in the accordion.
- Click **Last** to display the last chunk of rows. This option is dimmed if the last chunk is in the accordion.

Dashboard Area

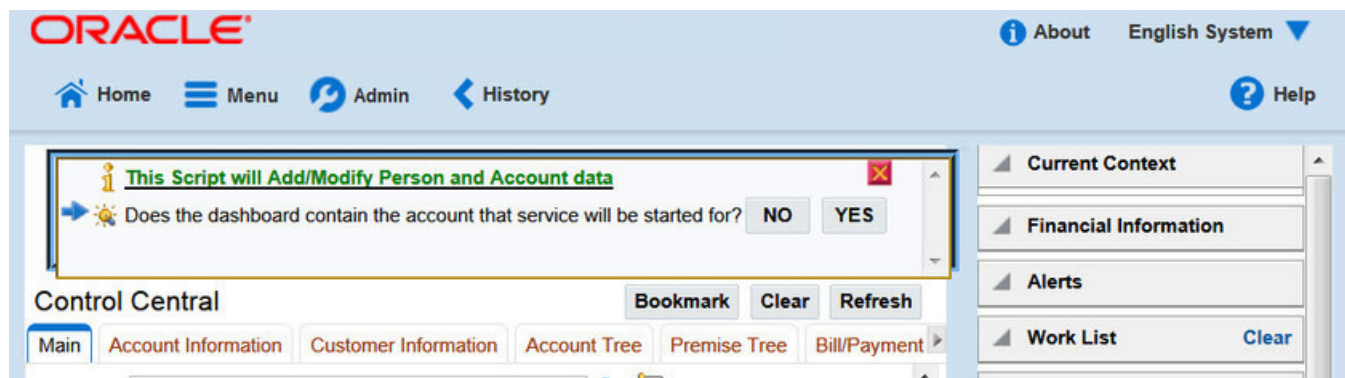
The dashboard is a [portal](#) that contains zones. These zones appear regardless of the object being displayed in the [Object Display Area](#). Refer to [Dashboard Portal](#) for a description of these zones.

NOTE: User configurable. Every user has control over the zones that appear in their dashboard. Refer to [Portal Preferences](#) for more information about how a user can customize the contents of the dashboard.

Script Area

When a [script is initiated](#), the [object display area](#) shifts down to make room for the script area.

The information in the script area takes you through the steps in the selected business process (provided by your implementation team).



Maintenance Pages

Maintenance pages are used to maintain (add, change, etc.) the information in the system using a fixed page metaphor. All user interfaces implemented in this way function in a similar manner. This section focuses on functionality common to all maintenance pages using this metaphor.

NOTE: Not all maintenance pages in the application use this style of maintenance page. The preferred metaphor for maintenance pages, however, is to use portal-based pages. For more information see [Common Base Portals and Zones](#).

The Unique Identifier Area

Every object in the system must have a unique identifier. For some objects, typically administrative or control table data, the unique identifier is defined by the user creating the record. Other objects, typically master or transaction entities, the unique identifier is assigned by the system after the record is saved.

Where the identifier is displayed depends upon whether the ID is user-defined or system generated.

Pages displaying objects with a user defined identifier (such as To Do Type) typically display the identifier on the left along with a search button. For master and transaction tables, (such as To Do Entry) the unique identifier is typically found in the upper right corner of the screen along with a search button and is labeled as the entity's ID. In this case, if there is an information string associated with the object, it is displayed on the left.

If the object being maintained has an associated context menu, a [Context Menu Button](#) is displayed appear.

Confirmation & Error Messages

Whenever a database modification action is requested, the system first validates the data to be put on the database.

First, the system validates the information. If there are errors, the first error encountered is displayed in a window. In addition to the error message, the error window also contains the error number and the identities of the programs that detected the error. If you don't understand an error and you have to call product support, please supply them with this information.

If there are no errors, the system is modified.

Warning If Unsaved Changes Exist

If you make any changes to an object and you don't commit these changes to the database, the following warning appears: "You have unsaved changes that will be lost if you continue. Press OK to continue anyway."

If you want to commit the changes, click **Cancel**. You will then return to the page and can make further changes or press Save. If you want to discard your changes and transfer to the new page, click **OK**.

CAUTION: This warning is not displayed if you are working on a *new* object on a fixed page and you navigate to a different page. Information about new objects is preserved in memory in anticipation of your return to the page. For example, if you are in the middle of adding a new user and you want to quickly reference information on a different page, you will not receive the above warning because your information will be saved. If you use the back button to return to the user page, all of the information you were adding will be present and you can enter the rest of the new user's data.

Query Pages

Query pages are used to display record sets in the system. Most query pages function in a similar manner. This section focuses on functionality common to all of the system's query pages.

The Search Area

The upper right corner of the screen is almost always the identifier of the object associated with the search. In Oracle Utilities Customer Care and Billing for example, the unique identifier on the Account Financial History page is the Account ID.

To the right of the unique identifier is a [Search button](#). This button is used to search for a different object than the one whose data is being displayed in the page.

If the object being displayed has an associated context menu, a [Context Menu Button](#) is displayed.

Portals

This section describes certain types of portals provided in the system where a user may control the content, for example the dashboard.

Portals used for base product query and maintenance pages are typically configured to not allow user customization. Refer to [Common Base Portals and Zones](#).

In most base products, there are several pages whose content can be customized by individual users. These portals exist because different users have different data requirements. For example, in Oracle Utilities Customer Care and Billing a billing clerk might prefer the customer information portal to show a customer's bills and payments. However, a credit and collection supervisor might prefer the same page to show an account's collection activity. Portals allow each user to define what they want to see.

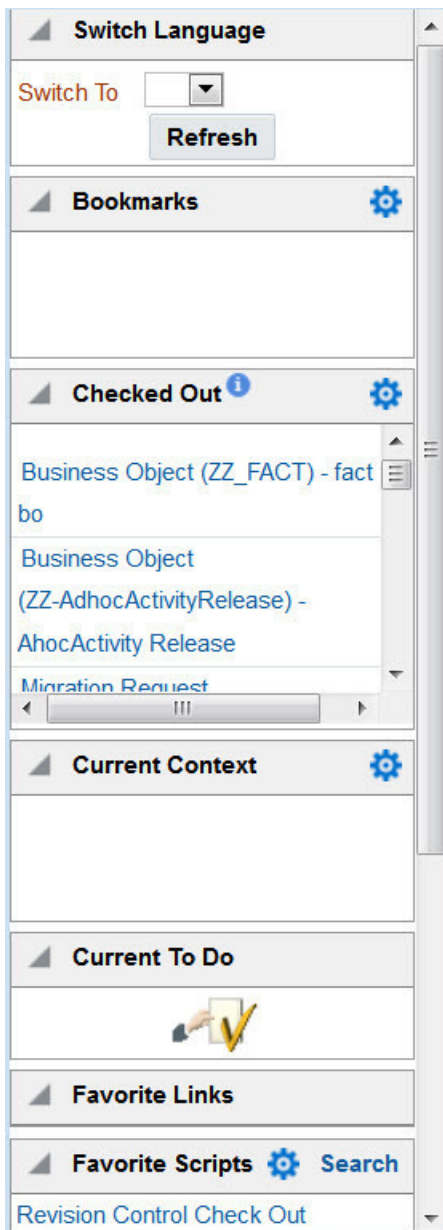
The contents of this section provide background information about portals.

Portals Are Made Up Of Zones

The term "zone" is used to describe a section on a portal. For example, the following is a portal with seven zones:

- Switch Language
- Bookmarks
- Checked Out
- Current Context
- Current To Do
- Favorite Links
- Favorite Scripts

NOTE: This example may not be available in your application; it is used for illustration purposes only.



Each User Can Customize Which Zones Appear

For any portal that is configured to show on portal preferences, each user has control over the following:

- Which zones appear on their portal
- The order in which the zones appear
- Whether the zones should be "collapsed" (i.e., minimized) when the portal opens.

Users define their preferences by clicking the [User Preferences](#) button. Pressing this button transfers the user to the [User Preferences](#) page on which their [Portal Preferences](#) can be defined.


CAUTION: Recommendation. We strongly recommend that you set your preferences to collapse zones that you don't use often. Why? Because we don't perform the processing necessary to build collapsed zones until you ask to expand a zone. Opening a page with zones collapsed accelerates the response times of portal.

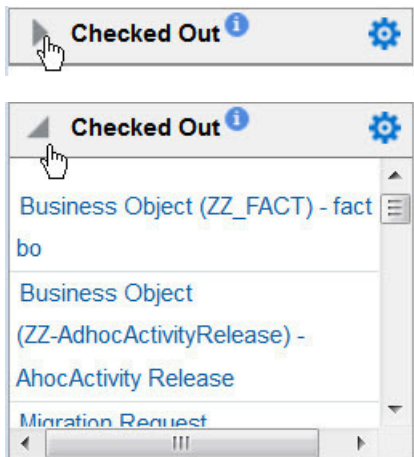
Users Must Be Granted Security Access To Each Zone

A user must be security granted access rights in order to see a zone on a portal.

FASTPATH: Refer to [Granting Access to Zones](#) for information .

Zones May Appear Collapsed When A Page Opens

[Portal Preferences](#) allows the user to collapse a zone (or minimized) when a portal appears to reduce space for the zone while keeping the data easily available. To view the information in a collapsed zone, the user can press the zone's expansion button, , to either expand or collapse the zone:



CAUTION: Recommendation. We strongly recommend that preferences be set up to collapse zones that may contain data that is only needed upon a user's request. The system doesn't perform the processing necessary to build collapsed zones until a user asks to expand a zone. As a result, indicating a zone is collapsed improves response times.

Dashboard Portal

The dashboard is a portal that always appears on the desktop. Its zones contain tools and data that are useful regardless of the object being displayed.

NOTE: Minimize the dashboard. You can minimize the dashboard by clicking on the vertical bar on the left side of the dashboard. This causes the dashboard portal to disappear allowing more of the page area to display. The vertical dashboard bar is then displayed on the right side of the page. Clicking the dashboard bar restores the display of the dashboard portal. Rather than clicking, you may press `Alt+J` to toggle the display of the dashboard.

The contents of this section describe the zones that are provided by the framework product for this portal. Please note the following additional information about the dashboard portal that may impact how it appears.

- **Additional zones may be available in other products.** The zones described below are available in every base product. Please refer to the product-specific documentation for additional dashboard zones available in your product (open the help and navigate to the index entry labeled **Dashboard**).

- **Context Sensitive Zones may appear.** Some pages may define additional zones that should appear in the dashboard only if that page is displayed. These are referred to as Context Sensitive Zones.
- **Dashboard zones are configurable.** The dashboard portal is configured to allow a user to define preferences for it. Refer to [Portal Preferences](#) for more information.

NOTE: Security. As a general rule, dashboard zones each refer to a unique application service so that a security administrator may configure which users are allowed to have access to which zones. Note that the user portal preferences indicates which zones a user has security access to.

Bookmarks Zone

After clicking the [Bookmark](#) button and saving a bookmark, it is visible in the **Bookmarks** zone. Clicking one of the links in the zone navigates to the page where the bookmark was saved and loads the information related to the record that was displayed when the bookmark was saved.

Click the trash can icon to delete the bookmark.

Note that a user may change the name of the bookmark or change the sequence of the bookmarks by navigating to the [User — Bookmarks](#) in the user preferences.


Current To Do Zone

The **Current To Do** zone is designed to streamline the process of working on To Do entries from assignment to completion. It displays key information about the user's current To Do entry and provides functions that help minimize the number of steps it takes to resolve the entry and potentially other entries related to it.

The following sections provide more detail about the zone.

Next To Do Assignment



Users may have the option to click the  To Do icon to request the next To Do entry they should work on. This functionality is optional and can be configured to follow your organization's business rules in determining the next task for a user. For example, if your organization practices work distribution "on demand", this feature can be configured to find the highest priority entry a user can work on and assign it to the user. Refer to [work distribution](#) for more information on this business practice.



NOTE: Next Assignment Rules Are In A Plug-in. Refer to [Next Assignment Algorithm](#) for more information on how to configure the system to take advantage of this function.

If the system is successful in determining the next entry the user should work on, this entry becomes the user's current To Do and its information is displayed on the zone.

User's Current To Do

The user's current To Do may be set using the Next Assignment algorithm, or by drilling in on a To Do from the To Do List or To Do Search page as described in [A User's Current To Do](#). Once a current To Do has been set, the zone displays details related to the To Do.

The **Message** describes why the entry exists. The message description allows the user to drill into the page on which the entry's respective object is displayed. For example, if the entry is associated with an asset that needs review, the user is brought to the asset page to view the information and potentially take action. If the [To Do type](#) has been configured to associate a [script](#) with this To Do entry's message number, then instead of being taken to the respective page, the associated script is launched.

Info displays the standard To Do information. An Info icon  appears if there are comments on the To Do and a Description icon  appears if there is a log entry of type **Forwarded**, **Sent Back** or **User Details** on the To Do. The description drills into the [To Do entry](#) page allowing the user to view more information about the entry or make changes to it as needed.

The user may use the **Complete** button in the zone to complete the current entry.

Next / Previous To Do Buttons

The **Previous To Do** and **Next To Do** buttons are enabled on this zone under the following conditions:

- The user selected an entry either on the [To Do List](#) page or the [To Do Search](#) page.
- At least one other entry exists *below* (for Next) or *above* the entry selected (for Previous).

Clicking **Next To Do** or **Previous To Do** will bring the user to the transaction indicated on the To Do Entry's message, where its respective object is displayed. It also makes that To Do the Current To Do displayed in the zone.

The keyboard shortcut to simulate Previous To Do: `Alt+Y`

The keyboard shortcut to simulate Next To Do: `Alt+Z`

NOTE: When a user drills down on a To Do entry from the To Do List page, the system automatically changes the status of the entry to **Being Worked**. If **Next To Do** or **Previous To Do** is subsequently used to scroll through this list, the system does *not* change the To Do entry's status. To change the status of these entries, the user must navigate to the [To Do List](#). On this page, multiple entries may be selected and their status changed with a single user-interface action.

Related To Do Entries

If the To Do entry has non-complete [related entries](#) a summary **Related To Do** description appears. The description states the total number of related entries, how many are still in **Open**, how many are assigned to current user and how many assigned to others. This hyperlink navigates to the [To Do Search](#) query allowing the user to review the related entries. At this point the user may decide to assign entries that can be positively identified as being caused by the same problem as the current To Do also to himself.

If the user determines that all related To Do entries have been resolved with the appropriate action taken, the **Complete All** button may be used to complete in one step the current entry and all its truly related entries assigned to the current user.

If the related entries are not resolved with one action, the user may opt to use the **Complete** button to complete the current entry alone and work on related entries one at a time. Using the related To Do information hyperlink the user may navigate to the To Do search page and drill into the first related To Do to work on using the message hyperlink. This entry now becomes the user's current To Do and the **Next To Do** and **Previous To Do** action buttons are set to scroll through the related entries list.

Favorite Links Zone

Each line in the **Favorite Links Zone** corresponds with one of the user's favorite links (e.g., transactions). Clicking on a line invokes the transaction. A number appears adjacent to the first nine entries to highlight that these links can be invoked by pressing an accelerator key (`Ctrl + a number`). Note that the numeric keypad on your keyboard cannot be used to implement this function.

Users define their favorite links by clicking the [User Preferences](#) button. Clicking this button transfers the user to the [User Preferences](#) page on which their [Favorite Links](#) can be defined.

Favorite Scripts Zone

The Favorite Scripts zone lists the business process assistant (BPA) scripts that you have defined on your [user preferences](#). You can click on a script to [execute](#) it. You can click the **Search** button in the zone to search for other scripts.

NOTE: If you do not see the Favorite Scripts zone in your dashboard, navigate to User Menu and click the **Preferences** menu item to display the [User - Portal Preferences](#) page. Under the Dashboard Portal row, select the **Display** option for the Favorite Scripts zone.

Favorite Searches Zone

If you marked a saved search as a favorite using [Saved Searches](#) functionality, the search is visible in the **Favorite Searches** zone. Clicking one of the links in the zone navigates to the page where the search was saved and populates the filter values and executes the search.

Switch Language Zone

The **Switch Language** zone is used to switch the application's language. Only languages that are supported by your implementation will appear.

To switch the application's language, choose the desired language and press the **Refresh** button.

NOTE: If your implementation uses a single language. If your implementation uses a single language, you should suppress this zone by [not granting security rights](#) to the zone's application service.

To Do Summary Zone

The **To Do Summary Zone** has several purposes:

- It presents a summary of the [To Do entries](#) that have been assigned to you and are **Being Worked On**.
- The colored "age bars" highlight the age of the To Do entries. The colors red, yellow and green are used to highlight the relative number of old, middle-aged and new To Do entries. The age associated with each of these colors is [user-configurable](#). Note, you can hover the cursor over an "age bar" to see the number of To Do entries.
- If you click on the To Do Type description or on an "age bar", you'll be transferred to the [To Do List](#) transaction where these To Do entries will be displayed. From this page, you can drill down to the specific entries.

NOTE: Time saver! After you drill down on entry from the [To Do List](#) page, if you have the [Current To Do](#) zone configured in the dashboard, you can use the **Previous To Do** and **Next To Do** buttons in the zone to scroll through additional To Do list entries without returning to the **To Do List** page.

Work List Zone

The **Work List** zone provides quick access to an entire column of an info or query zone. When a user clicks on a work list entry, the system will do whatever it would have done if they'd clicked on the hyperlink in the zone. In addition, the system ticks off that they've worked on the entry (this feature is mostly so users know that they've at least looked at something). Refer to [Work Lists](#) to better understand how to configure query columns to support work lists.

User Preferences

The Preferences (or My Preferences) transaction and the User Maintenance transactions are the same except that with My Preferences, only the following items can be modified:

- Main Tab
 - Language
 - Display Profile

- Time Zone
- Email Address
- Dashboard Width
- Home Page
- To Do Summary Age Bar Parameters
- Bookmarks
 - All information can be modified
- Portal Preferences
 - All information can be modified
- Favorite Links
 - All information can be modified
- Favorite Scripts
 - All information can be modified

User - Main

This page lets you view or define user information and access settings.

To open the page, either:

- Choose the **Admin > Users > Search** menu item.
- Choose the **Menu > My Preferences** menu item; or,
- Choose the **User Menu > Preferences** menu item in the application toolbar;

Note that some fields are only modifiable when the page is opened from the from the **Admin** menu.

Description of Page

Basic user identification information is viewed or set in the **First Name**, **Last Name**, **User ID**, and **Login ID** fields, all of which are modifiable only when the page is opened from the **Admin** menu. User and logon IDs must be unique. The Login ID is the id that the user uses to log in to the system. The User ID is the ID that is used throughout the application when users are assigned to tasks or when the user ID is stamped on records when logging add or update actions.

Language Code defines the language in which the user's screen prompts and messages appear.

Set **User Enabled** to allow the user to access the system.

NOTE: This field, available only when the page is accessed through the **Admin** menu, provides special field-level security. A user can change the field only if the **Enable/Disable** access mode is set.

Display Profile ID defines the [display options](#) for the user.

The next group of fields is used to view or set user preferences. For example, you can set up [portal preferences](#) on a "template user" and indicate that specified users will inherit their portal preferences from the template. Doing so provides consistency in the application of account information and dashboard zones. This fields are modifiable only when the page is accessed through the **Admin** menu.

- Set **User Type** to **Template User** if the user's preferences are to match those of other users.
- If a user's [portal preferences](#) are to be inherited from a "template user", use the **Portals Profile User ID** to define the user ID of the template user. If applied, the specified user will inherit portal preferences from the **Portals Profile User ID** and will be unable to customize those preferences.

- If a user's [favorite links](#) are to be inherited from a "template user", use the **Favorites Profile User ID** to define the user ID of the template user. If applied, the specified user will inherit favorites preferences from the **Favorites Profile User ID** and will be unable to customize those preferences.

The **Time Zone** associated with the user may be defined. Implementations may use this information for features and functions where the user's time zone impacts display of data.

Email ID captures the user's email address.

Dashboard Width defines the number of pixels to use for the user's [dashboard](#) area. A setting of 200 is recommended as a minimum. Depending on the width of a user's monitor, a higher setting (such as 250) will display more information without compromising the space on the main area of the application.

Home Page defines the page that appears when the user starts a session or clicks the [Home](#) link.

The To Do entry fields control how To Do entries are aged in respect of the "age bars" that appear on [To Do Summary](#), [To Do Supervisor Summary](#) and the [To Do Summary Dashboard Zone](#). Specifically:

- Use **To Do Entries Less Than ___ Days Old Should be Green** to define the number of days old a To Do entry must be less than to be considered "green" in the age bar.
- Use **To Do Entries More Than ___ Days Old Should be Red** to define the cutoff days after which a To Do entry is considered "red" in the age bar.
- Any To Do entry whose age isn't "red" or "green" is considered "yellow" in the age bar.

The **User Group** collection defines the group(s) with which the user is associated and the date (if any) when the user's association expires. This feature is modifiable only when the page is accessed through the **Admin** menu. For more information, see [The Big Picture Of Application Security](#). Assignments of users in a user group can also be established or maintained through the [User Group - Users](#) page.

NOTE: The ability to maintain the user groups on this page requires special field-level security. A user can change the field only if the **System Administration** access mode is set.

User - To Do Roles

Open the User page and then navigate to the **To Do Roles** tab to define the To Do roles assigned to the user.

The information on this page cannot be modified if accessed through **My Preferences**.

FASTPATH: A user's To Do roles control the types of **To Do** entries the user can access. For additional information on To Do roles, see [To Do Entries Reference A Role](#).

Description of Page

The grid contains the user's To Do roles. To modify a To Do role linked to the user, simply move to a field and change its value. To remove a To Do role, click the - (minus) button. To add a new To Do role, click the + (plus) button and enter the role.

Where Used

Every **To Do** entry references a role. A To Do role has one or more users (and a user may belong to many To Do roles). Users who are part of the To Do role assigned to a To Do entry may work on the To Do entry. For more information on this feature, see [The Big Picture Of To Do Lists](#).

User - Access Security

Open the User page and then navigate to the **Access Security** tab to define a user's security rights. You cannot modify this page if you accessed it through **My Preferences**.

FASTPATH: Refer to [The Big Picture of Application Security](#) for more information about data access roles and access groups.

Description of Page

Use the **Default Access Group** to define the access group that is defaulted on new records added by this user that are subject to [row security](#).

The scroll area contains the **Data Access Roles** to which this user belongs. A user's data access roles play a part in determining the accounts whose data they can access. Refer to [Access Groups, Data Access Roles and Users](#) for more information.

To add additional data access roles to this user, click the + button and specify the following:

Enter the **Data Access Role**. Keep in mind that when you add a **Data Access Role** to a **User**, you are granting this user access to all of the accounts linked to the data access role's access groups.

Use **Expiration Date** to define when the user's membership in this data access role expires.

NOTE: You can also use [Data Access Role - Main](#) to maintain a data access role's users.

The tree highlights the **Access Groups** to which the above **Data Access Roles** provide access.

User - Bookmarks

This page is used to manage a user's bookmarks that are created when a user clicks the [Bookmark](#) button and saves a bookmark. From this page the user can delete a bookmark, change the name of the bookmark or change the sort sequence.

Open the User page and then navigate to the **Bookmarks** tab to modify or delete a bookmark.

Description of Page

Sequence is the sort sequence associated with the bookmark. Bookmarks with the same sequence are then sorted alphabetically.

Name is the user-defined name assigned to the bookmark. This is the text that is visible in the [Bookmarks](#) dashboard zone.

NOTE: Favorite Links vs. Bookmarks. Favorite links are configured through the user page and define pages that a user can navigate to. They do not define context (although some pages may use information currently in global context when navigating). Some users may not be allowed to configure their own favorite links and would see the favorite links associated with a profile user. Bookmarks are created using a bookmark button available on each page. The bookmark saves navigation information, context information and also captures some information about the state of a page. Bookmarks are not governed by any user profile configuration.

User - Portal Preferences

The base product contains several [portals](#) that allow users to customize them via portal preferences. Portal preferences allow users to control:

- Which zones appear on the portal
- The order in which the zones appear
- Whether the zones should be "collapsed" (i.e., minimized) when the portal opens.

NOTE: You may not be able to change your portal preferences. If a note appears immediately before the list of portals, a system administrator has configured your user ID to reference a **Portals Profile User ID** (this is defined on the

Main tab). Preferences set in this way cannot be modified. System administrators do this in order to enforce a common look-and-feel throughout the user community.

Open the User page and then navigate to the **Portal Preferences** tab to modify a user's preferences.

Description of Page

The [accordion](#) contains a row for every portal configured to show on user preferences and to which you have access. To change how a portal's zones appear, [expand](#) the respective row and change the elements accordingly. The remainder of this section describes how you can configure how a portal's zones appear.

Zone lists all of the [zones](#) in the portal.

Place a check under **Display** if the zone should be displayed on the portal.

Place a check under **Initially Collapsed** if the zone should be minimized when the portal is initially opened. Setting zones as initially collapsed is a good idea to save space and to help pages to load more quickly. Refer to [Zones May Appear Collapsed When A Page Opens](#) for more information.

Use **Sequence** to control the position of the zone in respect of the other zones. This field is disabled if **Display** is not checked. Duplicate **Sequence** numbers are allowed.

For zones displaying data that is changing often, use **Refresh Seconds** to define in seconds how often the zone is refreshed. The minimum valid value is 15. The maximum valid value is 3600 (1 hour). A value of 0 indicates no automatic refresh.

Security Access indicates whether the user has access rights to a zone. Refer to [Granting Access To Zones](#) for more information.

User - Favorite Links

This page is used to define a user's favorite transactions and URLs. Each such link appears as an entry in the [Favorite Links Zone](#) in the [Dashboard](#) area.

NOTE: Ctrl + a number can be used to execute a favorite link. Rather than clicking a button in the favorite links zone, you can invoke your first nine favorite links by pressing `Ctrl + a number` (where the number corresponds with the relative position of the transaction in your list of favorites). Note that the numeric keypad on your keyboard cannot be used to implement this function.

Open the User page and then navigate to the **Favorite Links** tab to modify a user's favorite links.

Description of Page

NOTE: You may not be able to change your favorite links. If a **Note** appears beneath your **User ID**, your user id has been setup to reference a specific **Favorites Profile User ID** as defined on the **Main** tab. If a system administrator has setup your user ID in this way, your preferences may not be changed. System administrators may apply this feature in order to enforce a common look-and-feel throughout the user community.

Each row references a [Navigation Option](#). The navigation options supplied with the base package correspond with the system's menu items. This means you can setup a favorite link to open any transaction in the system (in either **add** or **update** mode).

NOTE: Favorite links can point to other places. The navigation options supplied with the base package always open a transaction's **Main** tab. If you find yourself continually accessing a different tab page, you can have your implementation team create a new navigation option that references the desired tab page. Once this navigation option is set up, you can reference it as one of your favorite links. In addition, it's also possible for your implementation team to setup navigation options that open a URL outside of the system. For example, your implementation team could setup a

navigation option to a credit rating company's website. Once this navigation option is set up, you can reference it as one of your favorite links.

Use a **Sequence** to control the order in which the navigation option appears in the **Favorite Links Zone**.

Security Access indicates if you have been given security rights to the respective transaction. Refer to [The Big Picture Of Application Security](#) for more information.

NOTE: Favorite Links vs. Bookmarks. Favorite links are configured through the user page and define pages that a user can navigate to. They do not define context (although some pages may use information currently in global context when navigating). Some users may not be allowed to configure their own favorite links and would see the favorite links associated with a profile user. Bookmarks are created using a bookmark button available on each page. The bookmark saves navigation information, context information and also captures some information about the state of a page. Bookmarks are not governed by any user profile configuration.

User - Favorite Scripts

This page is used to define a user's favorite scripts. Each script is displayed in the **Favorite Scripts** zone on the **Dashboard** portal where it can be used to [initiate a script](#).

NOTE: Ctrl + Shift + a number can be used to execute a favorite script. Rather than selecting a script from the business process assistant, you can invoke your first nine favorite scripts by pressing Ctrl + Shift + a number (where the number corresponds with the relative position of the script in your list of favorites). Note that the numeric keypad on your keyboard cannot be used to implement this function.

Open the User page and then navigate to the **Favorite Scripts** tab to modify a user's Favorite Scripts.

Description of Page

NOTE: You may not be able to change your favorite scripts. If a **Note** is displayed on top of the grid, your user id has been setup to reference a specific **Favorites Profile User ID** as defined on the **Main** tab. If a system administrator has setup your user ID in this way, your preferences may not be changed. System administrators may apply this feature in order to enforce a common look-and-feel throughout the user community.

Each row in the grid contains one of your favorite **Scripts**.

Use the **Sort Sequence** to control the order in which the scripts appear in the [business process assistant menu](#).

Security Access indicates whether you have access to the script's application service. Refer to [The Big Picture Of Application Security](#) for more information.

User - Characteristics

Select **Admin > User > Search** and navigate to the **Characteristics** tab to define characteristic values for the user. You cannot modify this page if you accessed it through My Preferences.

Description of Page

Use the characteristics grid to capture additional information about the user.

You can only choose **Characteristic Types** defined as permissible for the user record. Refer to [Setting Up Characteristic Types & Their Values](#) for more information.

Enter a valid **Characteristic Value** for the selected characteristic type. You may enter more than one characteristic row for the same characteristic type, each associated with a unique **Sequence** number. If not specified, the system defaults it to the next sequence number for the characteristic type.

User - Miscellaneous

Select **Admin > User > Search** and navigate to the **Miscellaneous** tab to view additional information associated with a user.

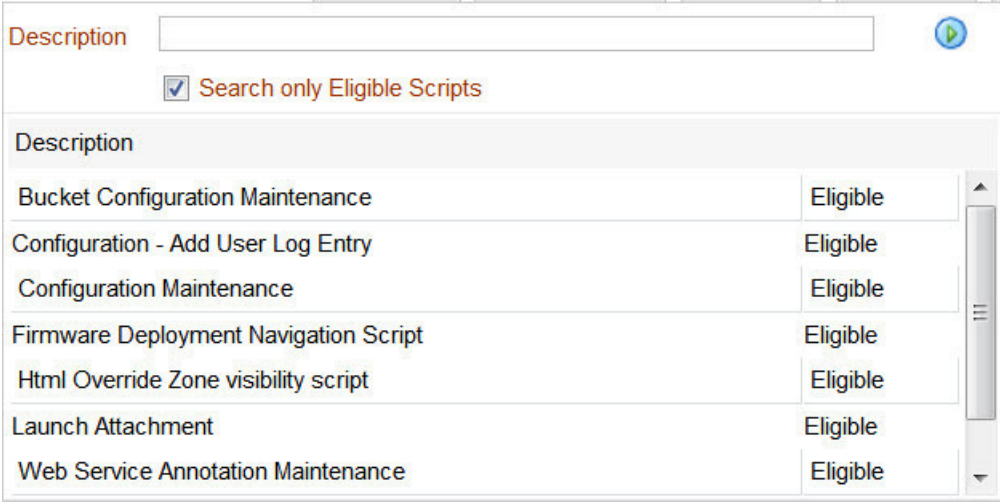
User Id, First Name and Last Name are displayed. Depending on your implementation, additional zones may appear on the User Portal

If additional zones appear, use the inline help for information about the fields managed on the zone.

Initiating Scripts

Your implementation team can set up business process assistant (BPA) [scripts](#) to walk you through business processes. Once scripts have been set up, you can initiate a script using any of the following methods:

- You can define a list of your "favorite scripts" on your [user preferences](#). You can then invoke any of the scripts from the Favorite Scripts zone on the dashboard portal. You can also invoke any of your first 9 favorites by pressing `Ctrl + Shift + a number` (where the number corresponds with the relative position of the script in your list of favorites). For example, if you press `Ctrl+Shift+2` and you've setup your "favorite scripts" to be **Start Autopay** and **Stop sending marketing information**; the **Stop receiving marketing information** script is invoked. Note that the numeric keypad on your keyboard cannot be used to implement this function.
- You can configure a menu item to launch a script by referencing a [navigation option](#) that is a **script** navigation option type.
- The system creates [To Do entries](#) to highlight tasks that require attention (e.g., bills in errors, accounts without bill cycles, etc.). Users can complete many of these tasks without assistance. However, you can set up the system to automatically launch a script when a user selects a To Do entry. For example, consider a To Do entry that highlights a bill that's in error due to an invalid mailing address. You can set up the system to execute a script when this To Do entry is selected. This script might prompt the user to first correct the customer's default mailing address and then re-complete the bill. Refer to [Executing A Script When A To Do Is Launched](#) for more information.
- You can indicate that a script should be [launched upon entering the system](#). This functionality enables an external system to launch the system and immediately launch a script.
- If you press `Ctrl+Shift+S`, an interactive menu is displayed:



The screenshot shows a search interface for scripts. At the top, there is a search bar labeled "Description" with a play button icon to its right. Below the search bar is a checkbox labeled "Search only Eligible Scripts" which is checked. Below this is a table with the following data:

Description	Eligible
Bucket Configuration Maintenance	Eligible
Configuration - Add User Log Entry	Eligible
Configuration Maintenance	Eligible
Firmware Deployment Navigation Script	Eligible
Html Override Zone visibility script	Eligible
Launch Attachment	Eligible
Web Service Annotation Maintenance	Eligible

The menu allows you to search for a script using any word in its description. Note, if you've turned on **Search only Eligible Scripts**, your search is limited to [eligible scripts](#). Note, the system displays an indication if each script in the grid is **Eligible** or **Ineligible** in this section.

You can initiate a script from this area by clicking on it.

When a script is initiated, this menu closes and the [object display area](#) shifts down to make room for the script area.

NOTE: Securing scripts. Only scripts the user is allowed to execute are displayed. Refer to [Securing Script Execution](#) for more information.

NOTE: Customize the script area. Using the [Height](#) step type, an individual script may be configured to use more or less of the script area than the standard script area. It is also possible to configure a script to hide the script area altogether.

The information in the script area takes you through the steps in the selected business process (note, your implementation team controls the information that appears in the script area).

The following points highlight useful information common to all scripts:

- **Return focus to script area.** Often during a script the focus moves from the script area to another section of the page. For example, you might be required to fill in data for a new record being added. When you are ready to return to the script area, rather than using the mouse to return to the area, you can press `Ctrl+Shift+F`. The focus returns you to either the first available input field or to a "default" button if no input field is available.
- **The Continue button.** Many steps in a script ask you to click the **Continue** button when you're ready to restart a script. For example, a step may ask you to confirm a customer's automatic payment information and then click **Continue** when you're ready for the script to restart. Rather than using the mouse to click this button, you can press `Ctrl+Shift+C`.
- **Selected button changes color.** When you click a button in the script area, the button selected changes color so that you can easily see which option you selected.

FASTPATH: Refer to [The Big Picture Of Scripts](#) for more information about scripts.


NOTE: You can abort a script at any time by clicking the red X button in the upper right corner of the script area (or by pressing `Ctrl+Shift+X`).

Accelerator Key Summary

The following table summarizes the keyboard shortcuts (hot keys) available in the system:

NOTE: It is possible that some of the following system accelerator keys may duplicate shortcuts, or hot keys, that are predefined within your browser. If this is the case, the browser's functionality commonly takes precedence.

Accelerator Key	Function
Alt+B	Returns to the previous page on which you were working. This is the equivalent of clicking the Back button.
Ctrl+Alt+C	Clears the currently displayed object from the page. This is the equivalent of clicking the Clear button.
Ctrl+Alt+D	Creates a duplicate of the object being displayed. Equivalent to clicking the Duplicate button.
Alt+G	Returns to the most recent page that was displayed when the Back button was clicked. Equivalent to clicking the Forward button.
Ctrl+Alt+H	Opens a dropdown list of recently-visited pages, the equivalent of clicking the History button.
Alt+J	Minimizes the dashboard . It also expands the dashboard, if it was minimized.

Accelerator Key	Function
Alt+K	Transfers you to To Do Entry - Main for the current item in the To Do list. Pressing Alt+K while on this page takes you to the first To Do entry on the list.
Alt+N	Displays the next entry in the search results. Equivalent to clicking Next Item .
Ctrl+Alt+P	Opens your User Preferences page. Equivalent to clicking the Preferences menu item.
Alt+P	Displays the previous entry in the search results. Equivalent to clicking Previous Item .
Ctrl+Alt+Q	Opens the business process assistant , where you can select a script that walks you through a business process.
Alt+R	Refreshes the page with the last saved version of the object being displayed in the page. Equivalent to clicking the Refresh button.
Alt+S	Saves any changes that you've made on the database. Equivalent to clicking the Save button.
Alt+X	Opens the To Do Summary .
Alt+Y	Displays the previous item in the To Do list. Equivalent to clicking the Previous To Do button in the Current To Do zone.
Alt+Z	Displays the next item in the To Do list. Equivalent to clicking the Next To Do button in the Current To Do zone.
Alt+Insert	Inserts a new row into an Editable Grid if the insertion point is in the grid. Equivalent to clicking the Insert button.
Alt+Delete	Removes a row from an Editable Grid if the insertion point is in the grid. Equivalent to clicking the Delete button.
Alt+F1	Invokes online Help for the application. Equivalent to clicking the Help button.
Alt+1 through Alt+9	These keys are only applicable when a Notebook is open. The shortcuts open the tab page whose relative position corresponds with the shortcut number. For example, if you press Alt+3 on a page with multiple tabs, the third tab is displayed. If there are more than nine tabs on a page, you must use the mouse or F2/Shift+F2 to display a tab. Note that you cannot use the numeric keypad for these shortcuts.
Ctrl+1 through Ctrl+9	Invokes the favorite link whose relative position corresponds with the shortcut number. For example, if you press Ctrl+3, the transaction for your third "favorite link" is invoked. Note that you cannot use the numeric keypad for these shortcuts.
Ctrl+Shift+C	If the current step in the Script area has a Continue button, this shortcut is equivalent to clicking Continue .
Ctrl+Shift+F	If the Script area is open and you are working on a different area of the page, this key returns focus to the script area. Focus returns to either the first available input field or to a "default" button if no input field is available.
Ctrl+Shift+S	Opens the business process assistant , where you can select a script that walks you through a business process.
Ctrl+Shift+X	If the Script area is open, equivalent to clicking the Close button in the script area.
Ctrl+Shift+1 through Ctrl+Shift+9	Invokes the favorite script whose relative position corresponds with the shortcut number. For example, pressing Ctrl+Shift+2 selects the second of your "favorite scripts". The numeric keypad cannot be used for these shortcuts.
Alt+left arrow	Populates an scroll area with the previous entry. Equivalent to clicking the scroll control  .
Alt+right arrow	Populates an scroll area with the next entry. Equivalent to clicking the scroll control  .
Enter when in a field with an adjacent Go Look button	Invokes the search. Equivalent to clicking the Search button .
Enter when in a field with an adjacent Search button	Invokes the search. Equivalent to clicking the Search button .

Accelerator Key	Function
Enter when a row in a search result grid is highlighted	Selects the item and closes the search. Note that after selecting an item from the search results, you can press Alt+N or Alt+P to scroll up or down through other items in the result grid without returning to the search results.
Enter when a menu item is highlighted	Selects the menu item.
Enter when the insertion point is within a tree node	Equivalent to clicking on the item in the tree: <ul style="list-style-type: none"> • If the insertion point is on a node image, the node is either expanded or collapsed, dependant on its current state. • If the insertion point is on a context menu, the context menu is opened and normal menu processing applies. • If the insertion point is on a "link" element, normal navigation processing occurs.
Tab when a row in a search result is highlighted	Highlights the next row.
Space when the insertion point is on a button	Equivalent to pressing the button.
Space when the insertion point is in a date/time field	opens the date/time selection window.
Space when the insertion point is in a check box	Toggles the check box between checked and unchecked.
Page Up while in a list grid	Scrolls the contents of the list grid up approximately one page.
Page Down while in a list grid	Scrolls the contents of the list grid down approximately one page.
up/down Arrow when in a dropdown list	Moves to the next or previous value in the list. You can also press a letter key to locate the next value starting with that letter.
up/down Arrow keys when in search results	Highlights the next/previous row in search results.
up/down/left/right Arrow when using a menu	Highlights other menu items.
F2	Opens the next tab page when a Notebook is open.
Shift+F2	Opens the previous tab page when a Notebook is open.

Portal and Zone Features

This section describes using [portals and zones](#) to find and display information. This is the preferred technique used by the base product for query and maintenance dialog.

Additionally, portals and zones may be configured by your implementation to supplement the base package provided functionality.

Zone Types and Characteristics

The topics in this section describe common zone types, function, and common characteristics.

FASTPATH: Refer to [The Big Picture of Portals and Zones](#) for a description of general portal and zone functionality.

Explorer Zones

The product provide several zone types that are used to find or display a set of information. These are called explorer zones. There are two types of explorer zones:

- **Info zones.** An info zone displays a list of information about an object. These types of zones often display information without requiring any input from the user.
- **Query zones.** A query zone allows users to find objects matching a given search criteria.

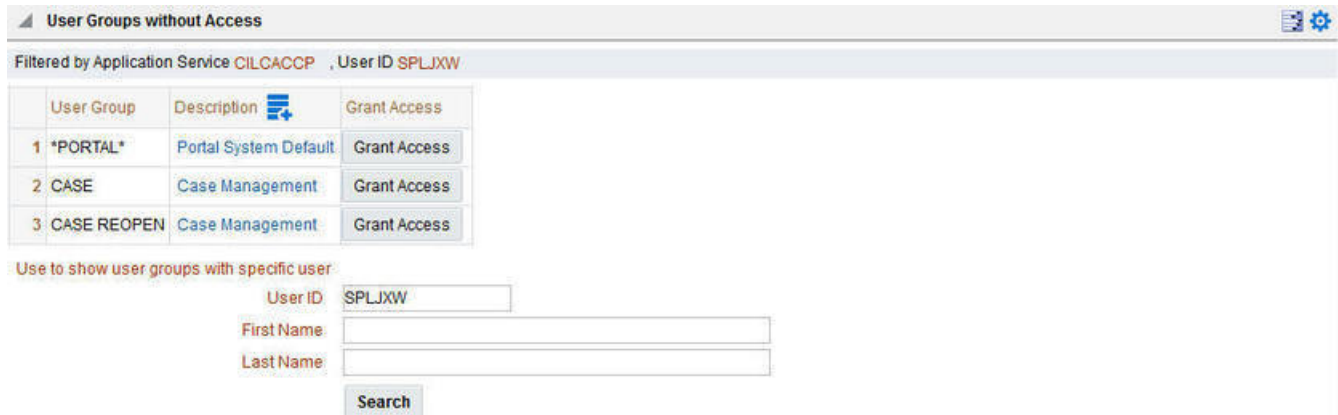
Virtually every aspect of an info or query zone is controlled by its zone [configuration](#). However, most share basic common components described in the subsequent sections.

Search Criteria / Filter Area

For query zones, the Search Criteria Area (also called Filter Area) contains the criteria used to find objects. The design of the zone controls the number and type of filters that appear in this area. After search criteria is entered, clicking the **Search** button causes the system to look for records that match the criteria. The results are shown in the Results Area.

For info zones, the zone's configuration indicates whether or not a Filter area is applicable and if so, whether or not it is open or closed by default. If the filter area is configured, the [Show Filter](#) icon is visible in the zone header. Click this icon to expose the Filter Area, if it is not visible and click the icon to hide the Filter Area if desired. Use this area to filter the information that appears in the zone by different values. Click the **Search** button to rebuild the Results Area using the specified filters.

For example, the following shows an info zone listing user groups that do not have access to a given Application Service where a filter area has been configured to allow users to show user groups specific to a given user by ID or name.



The screenshot shows a web interface titled "User Groups without Access". Below the title, it indicates the filter: "Filtered by Application Service CILCACCP, User ID SPLJXW". A table lists three user groups:

User Group	Description	Grant Access
1 *PORTAL*	Portal System Default	Grant Access
2 CASE	Case Management	Grant Access
3 CASE REOPEN	Case Management	Grant Access

Below the table, there is a section titled "Use to show user groups with specific user" containing three input fields: "User ID" (with the value "SPLJXW"), "First Name", and "Last Name". A "Search" button is located below these fields.

The info zone includes configuration to indicate whether the filter area, when exposed, should be shown at the top of the zone (above the results) or at the bottom of the zone (below the results). For query zones, the search criteria is always displayed at the top.












NOTE: Press Enter. Rather than clicking the **Search** button, you can press **Enter** to trigger the search / info zone refresh.

Description Bar

The Description Bar describes the search criteria that were used to build the Search Results Area.


Results Area

The Results Area displays the appropriate list of objects. For an info zone, this is the list of appropriate objects with any input filter criteria considered. The following is an example of the results for an info zone. For a query zone, this is the list that matches the search criteria defined in the Filter Area. The following is an example of the results set of an Info Zone.

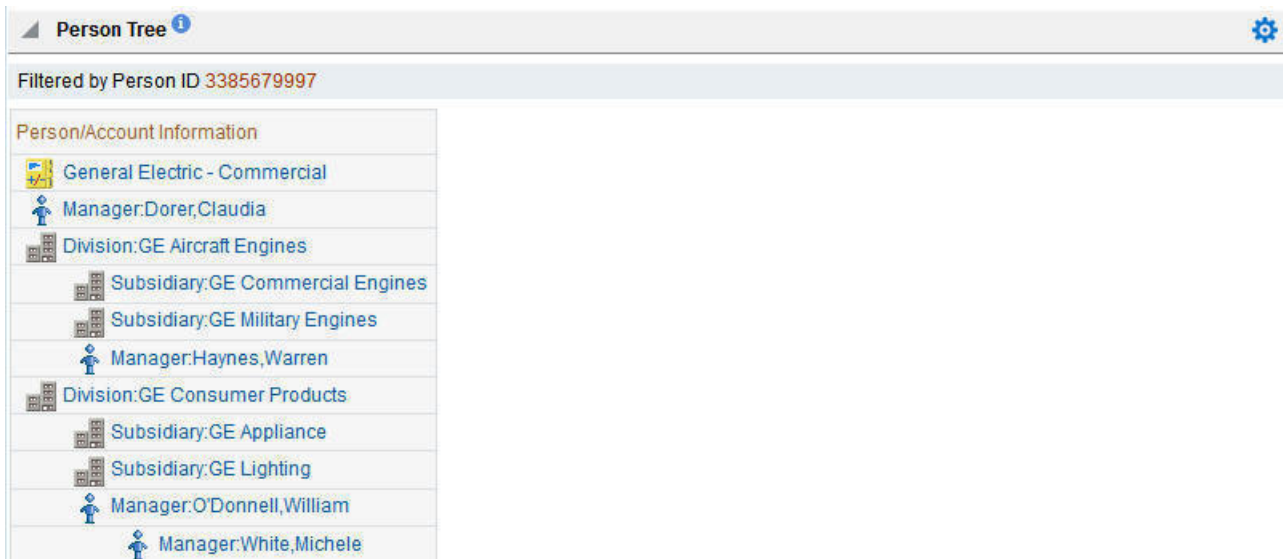
Business Objects with Status Reason		
Business Object	Description	Statuses
 1 F1-GenericBISyncRequest	Generic BI Sync Request	Discarded, Cancel
 2 W1-ActivityEquipRequirement	Equipment for Activity	Fulfilled, Canceled
 3 W1-ActivityGenerator	Activity Generator Lifecycle	Canceled
 4 W1-ActivityLaborRequirement	Labor for Activity	Fulfilled, Canceled
 5 W1-ActivityMaterialRequirement	Material for Activity	Fulfilled, Canceled
 6 W1-ActivityOtherRequirement	Other Resource for Activity	Fulfilled, Canceled
 7 W1-AdHocActivityCore	Ad Hoc Activity Core	Completed, Discarded
 8 W1-ApprovalRequest	Approval Request	Determine Approvers, Approval In Progress, Approved, No Appro
 9 W1-BCHeader	Blanket Contract Header	Created, Awaiting Activation, Active, Closed, Suspended
 10 W1-BCLine	Blanket Contract Line	Created, Ready for Revision, Revised, Complete
 11 W1-CompletionEventRoot	Completion Event Root	Executed, Discarded

For a query zone, the columns that appear may differ depending on the search criteria. For example, if you search for a customer by their driver's license number, the resulting columns may be different than the ones displayed when you search by customer name.

The following points highlight details related to viewing and working with the result set:

- **Broadcast.** If the results set is configured to display a broadcast icon, , it means that at least one other zone on the portal is configured to display data related to the record selected in the results list. Clicking this icon will cause the other zone or zones to be built or refreshed to display related data.
- **Hypertext.** One or more columns in the zone may be configured as hypertext. Clicking the hypertext will result in navigating to another page to display data related to that column based on the configuration for that column.
- **Maximum height.** The maximum height of the report area is controlled by the zone's configuration. A vertical scroll will appear when more rows exist than fit into the height.
- **Maximum rows.** The maximum number of rows that can appear is controlled by the zone's configuration. Excess rows are truncated before display. Please note, you can configure the zone to [download more rows to Excel](#) than appear in the Results Area.
- **Control column sorting.** By default zones are designed to allow a user to sort the results list by clicking any column heading. Clicking once sorts ascending. Clicking again sorts descending. However, a zone may be configured to not allow this type of sorting.
- **Rearranging columns.** A zone may be configured to allow a user to rearrange columns and add or remove columns visible in the Result Area. Refer to [Column Configuration](#) for the details.

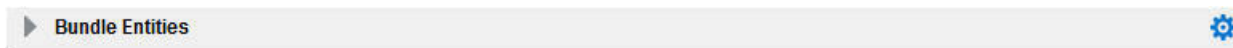
The type and variety of columns that appear in the Results Area is controlled by the zone's configuration. You can even set up an info zone to display information in hierarchical trees. For example, the following info zone from Oracle Utilities Customer Care and Billing has been configured to show a person's related persons.



Title Bar

The Title Bar contains the zone's description along with other icons and functions described in this section.

In the corner of the Title Bar, you'll see an indicator that tells you whether the zone is collapsed (e.g., minimized) or expanded. Each time you click the indicator, the zone toggles between expanded and collapsed. The following image shows a collapsed zone.




NOTE: Recommendation: For zones that are configured to show on portal preferences, set up [Portal Preferences](#) to collapse zones that aren't needed every time a portal is displayed. Rationale: The system doesn't build collapsed zones when a portal is displayed. Instead, collapsed zones are only built when a user expands them. Thus, indicating a zone is collapsed improves response times.

If the zone is configured with help text, a help icon appears to the right of the zone title. Click this icon to view the help text.

If a [saved search](#) is open for a zone, the name of the saved search is displayed after the zone title.

The following sections describe additional icons and tools visible on the title bar. Note that in addition to the information described in this section, additional actions that are specific to a zone may appear based on configuration. The actions may appear in the form of hypertext, an icon or a button.

Explorer Zone Menu

Click the Explorer Zone Menu icon  to list available tools. The following topics describe each tool.

Clear Filters

Click the **Clear Filters** link in the Explorer Zone menu to clear the filter values and zone data.

Export to Excel

Click the **Export to Excel** link in the Explorer Zone menu to download the zone's data to Excel. This link is hidden if the zone has been configured to not download rows to Excel.

Print

Click the **Print Zone** link in the Explorer Zone menu to print the zone.

Reset Zone

Click the **Reset Zone** link in the Explorer Zone menu to reload previously saved zone preferences from the database.

Save Search Options

Query zones and info zones that include filter areas, allow a user to capture and save filter criteria so that the same search can be more quickly executed in the future. Users may save many different search options for the same zone. Users also have the option to nominate one search as the default so that this search is automatically loaded upon entering the zone. In addition, users may choose to set one or more searches as a Favorite, which causes the search to appear in the [Favorite Searches](#) zone in the dashboard. This allows a user to navigate to the portal for that zone and load the saved search in one click.

The Explorer Menu includes various options for supporting Saved Searches that are described below.

Note the following additional points about this functionality:

- **Click Search before Saving.** When a user wants to save a search, the search criteria should be entered and the **Search** button should be clicked prior to saving the search.
- **Criteria is saved, not results.** Although a user must click **Search** prior to saving the search, the system saves the criteria, but not the results. When returning to the query or info zone in the future and opening a given saved search, the results displayed will be the current records in the system that match the saved search criteria.
- **Multi-Query Zones.** When a user is on a multi-query zone, the searches are saved at the multi-query zone level so that all saved searches for the various search dropdowns for a given multi-query zone are visible in the Open Saved Searches menu option.
- **Functionality is Secured.** If for any reason an implementation wishes to restrict which users are allowed to save searches, the functionality is securable. Only users with **Inquire** access to the **F1_USRFAVSCH** application service will see these options in the explorer menu. This is the same application service that is used to provide access to the Favorite Searches dashboard zone.

Save As

The **Save As...** option in the Explorer Zone menu allows a user to save the current entered search criteria. This should be clicked after executing the search.

The user is prompted to provide a name for the search. This search is now available from the Open Saved Searches option. If desired, a user can also choose to make this search the Default or may choose to Add to Favorites.

This option is visible whether or not an existing saved search is loaded. When a saved search is loaded, this option allows a user to make changes to the search criteria and save a new search with that criteria.

Open Saved Search

The **Open Saved Search...** option in the Explorer Zone menu is visible if there are any saved searches for the zone that is currently visible. Select this menu entry to view the current list of saved searches for the zone. For multi-query zones, any saved search for any of the query zones that are part of the multi-query zone are visible. Choose the desired saved search. The criteria is updated with this information and the search is performed.

Save

The **Save** option in the Explorer Zone menu is visible if a Saved Search is already loaded. This option allows a user to save changes to the criteria of an existing saved search (overwriting the previously saved criteria).

Set As Default

The **Set As Default** option in the Explorer Zone menu is only visible if a saved search is loaded and it is not currently the default. Click this option to nominate this search to be the one that is loaded when navigating to this zone from the menu.

Remove As Default

The **Remove As Default** option in the Explorer Zone menu is only visible if a saved search is loaded and it is currently the default. Click this option to no longer load this search automatically when navigating to this zone from the menu.

Add To Favorites

The **Add To Favorites** option in the Explorer Zone menu is only visible if a saved search is loaded and it is not currently included in the favorite searches zone. Click this option to include this search in the [Favorite Searches](#) zone. Once a search is in the Favorite Searches zone, you can use the link to the search in the dashboard to navigate to the portal for this zone and load the saved search in one click.

NOTE: Not available for dependent zones There are some portals that include a zone that is not the main query or info zone on the portal, but is rather a zone providing additional information for a record that is displayed. If a zone like this has filter options or configurable columns, a user is able to save searches and mark one as a default, but will not be able to add the saved search to favorites because the zone is dependent on other data on the page that is not captured for the individual zone.

NOTE: Dashboard Zone Preferences. Be sure that your [portal preferences](#) are configured to include the Favorite Searches dashboard zone.

Remove From Favorites


The **Remove From Favorites** option in the Explorer Zone menu is only visible if a saved search is loaded and it is currently included in the [Favorite Searches](#) zone. Click this option when you want to keep the saved search but you no longer want it visible in your Favorite Searches zone. (Deleting a saved search that is also marked as a favorite will automatically remove the search from the favorite searches zone.)

Delete

The **Delete** option in the Explorer Zone menu is only visible if a saved search is loaded. Click this option to remove the currently opened saved search from the list of saved searches.

Note that if the saved search had been configured as favorite, deleting the saved search remove it from the favorite searches zone.

Show Filters

Click the **Show Filters** icon  to show the [Filter Area](#). This button only appears if the zone has been configured to have filters.

NOTE: If the zone is configured to have filters and the drag and drop (column configuration) area, only one or the other is shown at any given time. It means that a user can either change the filter criteria and refresh the search results or change the column configuration and refresh the results.

Show Column Configuration Area

Click the **Column Configuration** icon  to show the [Column Configuration Area](#). This button only appears if the zone has been configured to allow columns to be dragged and dropped into the Results Area.

NOTE: If the zone is configured to have filters and the drag and drop (column configuration) area, only one or the other is shown at any given time. It means that a user can either change the filter criteria and refresh the search results or change the column configuration and refresh the results.

The Open Zone Maintenance Icon

Click the "wrench" icon  to switch to the zone's [zone maintenance](#) page. On this page, you can change the parameters that control the zone's behavior.

NOTE: Debug mode only. This icon only appears when the system operates in [Debug](#) mode.

Configurable Columns

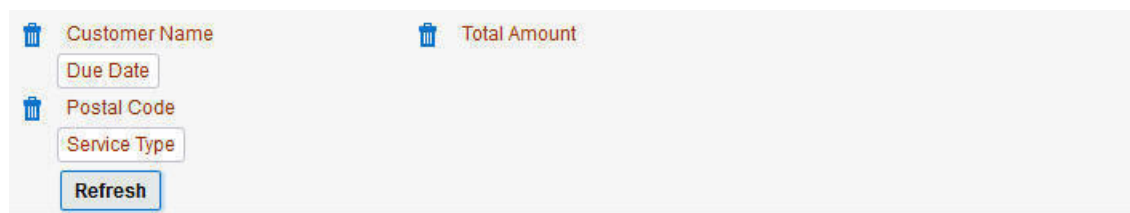
Info zones and query zones may be configured to allow users to rearrange columns and to remove or add columns to the output. This is also referred to as “drag and drop” of columns.

When a zones is configured to support this functionality, the [Show Column Configuration Area](#) icon is visible in the zone title area. Click this icon to expose the Column Configuration Area, if it is not visible and click the icon to hide the Column Configuration Area if desired.

If configurable columns are allowed, a user may rearrange the order of columns currently displayed in the Results Area by clicking the column heading and “dragging” it to the desired location.

In addition, the column configuration area may display columns that are not currently in the Results Area. These are indicated by a cream background. A user can drag one of these columns to the results area and then click **Refresh** to rebuild the zone. To remove a column that is currently in the results area, click the trash can icon next to the column name in the column configuration area and then click **Refresh** to rebuild the zone.

The following is an example of how the Column Configuration Area looks:



Multi-Query Zones

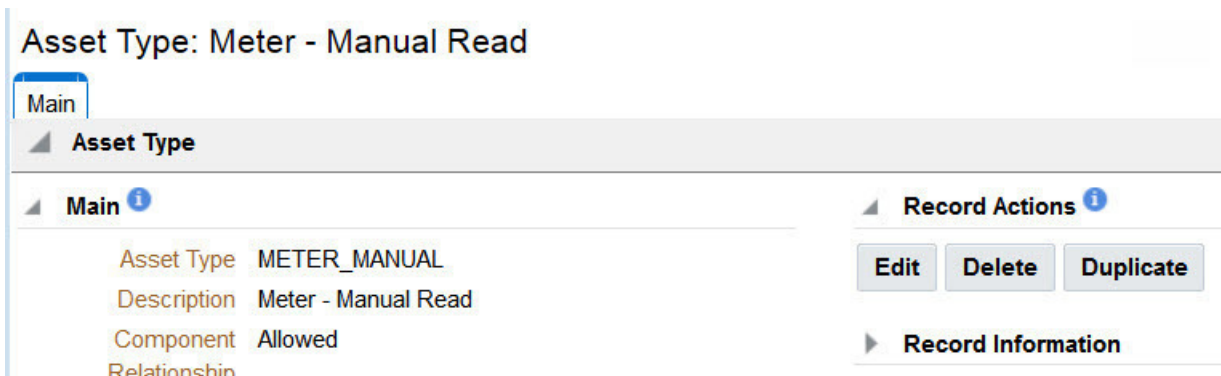
This type of zone combines individual Query Zones into a single query zone. A **Search By** dropdown, built from the zone descriptions of the individual Query Zones, allows you to select the desired query to execute. Selecting a specific Query Zone display the filters of that query in the filter area. At this point the zone simply executes the selected Query Zone, as if it was executed on a standalone zone. By default, the first query zone option is selected.



Map Zones

A map zone displays information related to one or more objects. For objects that are maintained using portal-based pages, the standard display of the object is using a map zone. Map zones may be configured to display an explicit map or may dynamically display the map based on the configuration of the record being displayed.


Because the information displayed in a map zone is configured using HTML, virtually any format supported by HTML is possible. The following is an example of a typical map zone used to display information about an object.



A [tool tip](#) can be used to display additional help information for the details shown in a map zone. The tool tip may appear adjacent to the section header or adjacent to individual elements on a map.

NOTE: You can customize the help icon or disable map help using the [Custom Look and Feel](#) feature configuration.

Open Map Maintenance Icon

Click the "hammer" icon  to switch to the [UI map](#) maintenance page. On this page, you can view the HTML configuration for the zone. Note that for map zones that display a business object configured without a map, the elements displayed are automatically rendered. In this case, no “hammer” icon is visible.

NOTE: **Debug mode only.** This icon only appears when the system operates in [Debug](#) mode.

Common Characteristics of Zones

The following topics describe common characteristics of all zone types supported by the base package.

Zone Visibility

A zone may be configured to dynamically be visible or completely hidden depending on the information being displayed. When hidden not event the header is displayed. Refer to the common **Zone Visibility Service Script** parameter for further details.

Control Zone Headers

A zone may reference a UI Map to define and control its header. This feature provides the means to achieve a uniform look and feel for the application as well as add functionality to the zone header beyond being just a title bar to open and close the zone. Refer to [UI Map](#) and to its context sensitive tips zone for more information on **Fragment** UI Maps and how they are used to construct zone headers.

Common Base Portals and Zones

The following sections describe common standards followed by the product for base query and maintenance dialogue.

All-In-One Portal

The all-in-one portal is used for low volume objects, where a small number of records are expected.

For example, many control tables use this style of portal. In this portal, rather than separate query and maintenance portals, a single portal allows a user to find a record and maintain the record. A zone listing all existing records is displayed when first entering the portal. A user can perform actions on existing records or add new records from there. More details about the list zone are provided below.

Refer to [Maintenance Portal](#) for information about zones commonly found on maintenance portals.

The all-in-one portal will typically provide the ability to add a new record. This may be a button in the Page Action area or a hyperlink defined directly on the list zone's header area.

List Zone

This info zone appears when first entering the portal and displays a row for each record.

The following functions are commonly available in a list zone:

- A broadcast icon is included in each row. Clicking this icon opens other zones that contain more information about the adjacent record.
- An **Edit** icon may be included in each row. Clicking this icon launches the maintenance BPA script to modify the adjacent record.
- A **Duplicate** icon may be included in each row allowing the user to duplicate the adjacent record without having to first display the record.
- A **Delete** icon may be included in each row allowing the user to delete the adjacent record without having to first display the record.
- If the record has a simple BO lifecycle of **Active** and **Inactive**, the list zone may include a button to **Activate** or **Deactivate** the adjacent record.

Separate Query and Maintenance Portals

For high volume objects where more sophisticated search criteria is needed to find a particular record, separate portals are provided to search for the record and to maintain it.

The query portal contains a zone to search of the object that you wish to maintain. Additional information about the query portal's search zone is provided below. Clicking an appropriate record in the search results navigates to the object's stand-alone maintenance portal.

The query portal will typically provide the ability to add a new record. This may be a button in the Page Action area or a hyperlink defined directly on the list zone's header area.

Query Portal Search Zone

The query portal contains one zone used to search for the object.

The zone may be a standard query zone or may be a multi-query zone providing one or more ways to search for your record.

Once the search results are displayed, click on the hypertext for the desired record's information string to navigate to the stand-alone maintenance portal.

Maintenance Portal

A maintenance portal refers to a page that is used to view and maintain a specific record. These types of records are governed by a business object, which defines the user interface layout for both the display of the record and the add or update of the record.

An all-in-one portal is a type of maintenance portal, once a record has been broadcast from the list zone. For the separate query and maintenance portal dialogue, the maintenance portal includes only zones relevant for the specific record selected.

Maintenance portals may have multiple tabs to display information relevant for the object. Often, if the object has a Log, the log is on a separate tab. (Although there may be some maintenance portals that include the log zone on the Main tab.)

The stand-alone maintenance portal is typically configured with the ability to navigate back to the query portal using either a **Search** button in the Page Actions area or using a **Go To Search** hyperlink in the main zone's header area. The stand-alone maintenance portal may also include the ability to add a new record using an Add button in the Page Actions area. (If not, a new record can be added by navigating through the menu or by navigating to the query portal where an Add hyperlink may be found.)

The following zones are common to portals used to maintain a record.

Note that additional zones may exist for specific all-in-one portals or stand-alone maintenance portals based on the particular functionality. Refer to the documentation for each portal to understand the specific zones that are provided.

Object Display Zone

This is a zone that displays the object being maintained.

An object being displayed may or may not allow for actions or updates to be made. Typically there is a Record Actions section in the display area that presents the valid actions. Depending on the functionality, the actions may prompt the users to input values required for that action.

Log Zone

If the maintenance object for the record includes a status, it typically has a log that records important events in its lifecycle.

This log is shown in an info zone on the portal, either on the Main tab or on a separate Log tab. A link appears in the zone's title bar to allow the user to add a new manual log entry.

Activities

About Activities

An activity is a task to be performed by a crew at a particular geographic location within a time period.

The system prioritizes the activity relative to other activities, schedules it to appropriate resources in a timely manner, and reports back completion information to the requesting host system.

Details provided with the activity control key aspects of how it is managed, such as

- The type of work to be performed
- Where work is to be performed?
- When work should be performed?
- How long is it estimated to take?
- Who may or may not be assigned to perform it. The activity may pose restrictions around the crew that may be considered to perform the work. For example the assigned crew may need to possess certain skills or equipment in order to carry it out.
- And much more

This section describes concepts related to managing activities.

Activity Type Controls Everything

Each activity references an activity type that describes the type of work to be performed as well as many settings that control how activities of that type should be scheduled, prioritized, dispatched, worked and a lot more.

Refer to Understanding Activity Types for more information.

An Activity Is Associated With One Host System

An activity is requested by a host system. All communications related to an activity are exchanged solely with the host system that requested it, which is defined on the activity's type.

Refer to The Activity's Host System for more information.

The host system sends in the activity along with the activity's [unique identifier](#) in that system. The host identifier of an activity is required for tracking purposes as well as for all communications from and to the host system related to the requested activity.

There are situations where activities are initiated within the system. For example, crews may use their mobile device to [initiate an activity in the field](#) and dispatchers may create an activity to [assist](#) another crew with their activity in the field. The host system has to be aware of all work in the service territory, including activities it did not initiate. In these situations, the system notifies the host system defined on the [activity type](#) of the activity that is being created within the system (the assist or activity created in the field). In response, the host system creates a corresponding activity in that system and reports back its unique host identifier for further tracking and communications. This ensures that regardless of where an activity was initiated, all messages such as status and completion information are sent to the appropriate host system.

Activity Workflow

Activity states are depicted in the following diagram:

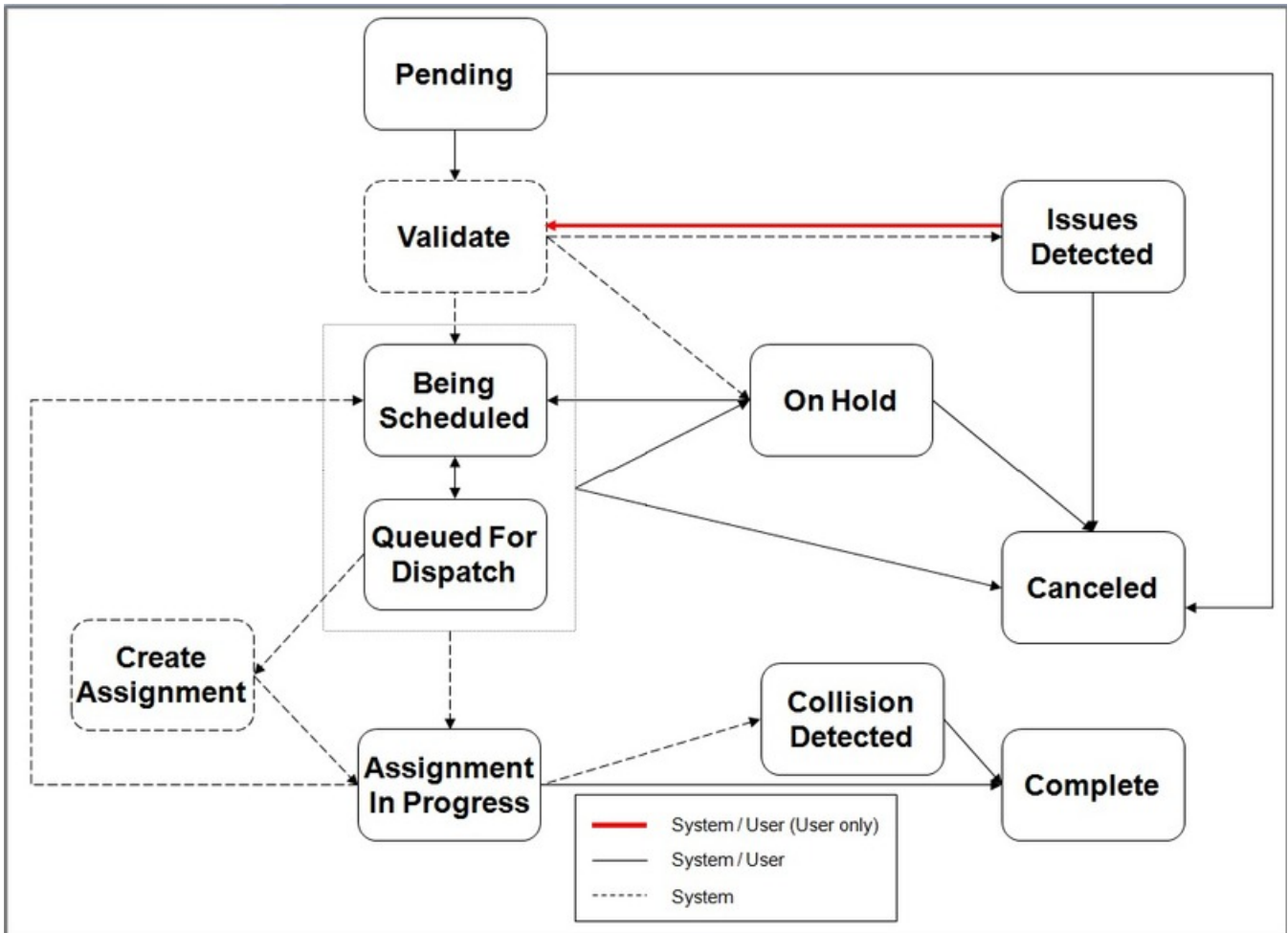


Figure 1: Activity Lifecycle

This section describes key stages in the overall lifecycle of an activity.

Refer to [About Complex Activities](#) for more information about complex activities and their unique workflow.

Validation

When an activity is received from the host system (or created internally), it is validated to ensure it has all the information needed for properly handling it. Basic validation ensures the system accepts and sends to the scheduler well formed activities.

If the activity is valid, the activity is automatically set to the **Being Scheduled** state and made available for [scheduling](#). If critical errors are found, the activity is rejected and not saved at all.

The system will accept the activity if all the errors are such that they can be corrected by a user within the application. In this situation, the activity is saved in an **Issues Detected** state. The activity stays in this state until all errors are corrected. Refer to the [Issues Detected](#) state for more information.

Once all errors are fixed by a user, the activity can now be [scheduled](#) and is automatically set to the **Being Scheduled** state.

Custom rules may be implemented to automatically place a valid activity [on hold](#) for various business reasons instead of directing it to be scheduled.

Issues Detected

When an activity is added and does not have all the necessary details to be properly scheduled, it is set to this state along with a list of the issues detected.

NOTE: If critical errors are found, the activity is rejected and not added at all. Only if the validation algorithms determine that the issues can be resolved by a user within the application, the activity is allowed to be added in this state.

A To Do Entry is raised to notify a user of the situation. When the user corrects the information, the activity is [validated](#) again.

Being Scheduled

An activity in this state is available for scheduling but its schedule is not finalized yet.

The activity typically starts in this state when it is initially received and dispatched to a specific crew (or contractor company) to be worked and then proceeds along its lifecycle towards completion. However, there are times when the assigned resource declines the work and the activity is set back to this state to be rescheduled.

Refer to [Scheduling An Activity](#) for more information about this process.

While the activity is in the **Being Scheduled** state

- The scheduler keeps improving on its scheduling decisions. During this continuous optimization phase, the activity's schedule may frequently change between being assigned, reassigned to another shift (or contracted capacity), its order changed within its currently assigned shift's route, as well as become unassigned again. This goes on until it is time to [queue the activity for dispatch](#) so it can be communicated and dispatched to the assigned crew (or contracting company).
- The activity's time windows are checked and if the activity has expired, it may be automatically extended or [canceled](#) as per configuration on the activity type. Refer to [Expired Activities](#) for more information.
- The activity's time windows are recalculated to take into account new or canceled applicable override time window events, if any exist. Refer to [About Override Time Windows](#) for more information.
- The activity's preference ranking of contractor companies are kept up to date. Refer to [Contractor Eligibility Rules](#) for more information.
- If the activity type is configured to advance allocate work of this type, then such activity may be allocated to the currently assigned crew shift some time prior to the shift's planned start. Refer to [Advance Allocation](#) for more information.
- Business rules may put the activity [on hold](#).

NOTE: Many of these business rules rely on the task being periodically monitored by the **Task Monitor** batch process. Refer to [M1-TSKTR](#) batch control for more information.

Queued For Dispatch

Depending on shift configuration, when the crew starts their shift, they may be dispatched their entire schedule for the day or only the first few tasks. Shift configuration may also allow the crew to preview a few of these tasks in advance, prior to starting the shift.

Refer to [Locking The Schedule](#) for more information.

When it is time to make the activity available for preview or dispatch it to the crew's mobile device, the system queues it for dispatch so that its schedule is locked and can no longer change once communicated to the crew.

A crew's work schedule is communicated to them in the order they should work their tasks. Therefore activities are queued for dispatch in that order.

Setting the activity to the **Queued For Dispatch** state is the first step in the dispatch process. The scheduler adds the activity to the end of the dispatch queue and stops further optimization of its schedule. The activity retains its current schedule from this point on, unless the activity is explicitly [rescheduled](#).

The following business events queue activities for dispatch:

- If a crew's shift is configured to support [Advance Dispatching](#), the configured number of first tasks they may preview are queued for dispatch at a predefined time prior to the shift's planned start time.
- When a crew starts their shift, the scheduler automatically queues for dispatch (if not already queued in advance) their first few scheduled tasks, all of them or none, as per the shift's drip mode configuration. If a drip horizon is used, each time the crew starts or completes a task the scheduler proactively queues the next one for dispatch. Refer to [Controlling Auto-Dispatching](#) for more information.
- All of the activities assigned to a contractor's capacity are queued for dispatch and committed to the contractor company at the predefined notification time before the capacity's start time. Refer to [Advance Notification](#) for more information.
- A dispatcher may manually mark an activity as ready for [dispatch](#). For example, the dispatcher may decide that the crew should perform the activity earlier than originally planned by the scheduler for some reason.

NOTE: Queuing an activity for dispatch should not be confused with the actual process of dispatching the activity to the crew's mobile device (or communicating it to the contractor company). These are consecutive but separate steps.

Certain conditions have to be met before the activity is actually dispatched. The activity remains in the queued for dispatch state until the following conditions are met to proceed with the actual dispatch.

- If the activity is assigned to a crew's shift the following conditions apply:
 - The activity cannot be dispatched until the scheduler locks the activity's sequence within the route, i.e. stops further optimization of the activity's schedule, so that it can be communicated to the crew.
 - This condition is only applicable in situations when a user has [manually dispatched](#) an activity and the schedule has not yet processed that request. Once the scheduler processes the request it locks the activity's schedule.
 - If the activity was auto dispatched, i.e. placed in the queue by the scheduler itself, this condition is always met since the scheduler always locks the activity's schedule at the same time.
- Actual dispatching can only happen once the crew has started their shift and is ready to take on work.
- If the activity was placed in the queue in advance, prior to the crew starting their shift, then it would wait in this state until the shift starts.
- If the activity is queued after the shift has already started, then this condition is always met.
- If the activity is assigned to a contractor capacity, it is communicated to the contractor (dispatched) right away as it enters the queue with no additional wait time. Refer to [Committing An Activity To A Contractor Company](#) for more information.

NOTE: When conditions are met, the activity would be automatically dispatched without relying on any background process to move it along. Even when the activity is advance dispatched, it would automatically be dispatched when the shift starts.

When ready to be dispatched, the system creates an assignment for the activity and its currently assigned crew shift (or contracted capacity), and sets the activity to the **Assignment In Progress** state. At this point the crew (or contractor company) is in charge of [working the activity](#).

Refer to [The Dispatch Process](#) to better understand the big picture of the dispatch process.

Assignment In Progress

When it is time for an activity to be dispatched to the assigned crew (or capacity contractor), the system creates an assignment for the activity and the assigned shift (or contracted capacity), and sets the activity to the **Assignment In Progress** state.

An assignment is a copy of the activity information created by the system to track the assigned resource's progress in accomplishing the work. The assignment is created on the server and a copy of it is dispatched (or sent) to the assigned resource. All updates provided by the assigned resource are communicated via their copy of the assignment.

Refer to [About Assignments](#) for more information.

The activity is considered dispatched when the assigned resource acknowledges receiving their copy of the assignment and thus accepting the work.

Refer to [The Dispatch Process](#) to better understand the big picture of this process.

While the activity is in this state the ownership of the work is with the assigned resource and the system waits for their resolution. Either the resource performs the work and completes their assignment, or declines it and returns their assignment.

NOTE: If due to dispatcher interventions and [recall](#) attempts the activity was sent to multiple crews the system would wait for the resolution from all these assignments before it can assess the activity's state.

Possible resolutions may be as follows"

- If the resource (crew or contractor company) performs the work and completes their assignment, the activity is automatically [completed](#). The activity is transitioned to the **Completed** state and the completion information is sent to the requesting host system.
- If the resource declines the work and returns their assignment, the activity automatically transitions back to the **Being Scheduled** state to be [rescheduled](#).
- If due to dispatcher interventions and recall attempts more than one crew have actually indicated that they completed their assignment, the system has no way of knowing which crew completion information to use. The activity is set to a **Collision Detected** state and a To Do Entry is raised to notify a user of the [collision](#) situation. A dispatcher should review the information and manually choose one of the completed assignments to be used to complete the activity.

Completed

The activity is set to the **Completed** state when:

- The crew (or contractor company) performs the work and completes their activity assignment along with all the necessary completion information. Refer to [Completion Information](#) for more information.
- The crew completes their activity assignment but marks it as incomplete because actual work could not be performed; for example, when details on the activity are invalid or there was some problem getting access to the work site. A completion message is sent to the host indicating that the system will no longer attempt to perform the work. This allows the host system to cancel their activity, or correct and resend it as needed. Refer to [Completion Information](#) for more information.
- The host notifies the system that the activity has been completed in the host system. This is not a typical situation but there are business practices where the crew may prefer to key in the completion information directly in the host system application. This may happen when all communication with the crew is handled by the host system and not by this system. Refer to [Crew May Be Managed Externally](#) for more information. In either situation, since the host has the completion information already, there is no need to send them a completion message.

Collision Detected

Typically only one crew completes the work requested by an activity and the completion information they have entered on their assignment is sent to the host system. However, due to user intervention and recalls, the activity may be assigned concurrently to more than one crew.

If for some reason more than one of the concurrent crews assigned to work the activity claim they have worked it then the system has now multiple sources for completion information and has no way to determine which to forward to the host system. This collision situation requires user intervention.

While in the [Assignment In Progress](#) state waiting for open assignments to be finalized, if the system detects more than one completed assignment for the activity, it sets the activity to this state and raises a To Do entry for a user to review the assignments and manually choose the one to use to complete the activity.

Refer to [Concurrent Assignments](#) for more information on this situation.

Canceled

Canceling an activity is typically initiated by the host system for the purpose of preventing the work from being performed for some reason. The system can only confirm a cancellation when the activity is not yet dispatched. Once dispatched, the system attempts to cancel the activity but only the crew has the latest information as to its current state.

Refer to [Activity Cancellation By Host](#) for more information about how such a request is handled.

[Expired activities](#) may also be automatically canceled based on the activity type configuration.

On Hold

There are situations where the activity is not yet ready to be scheduled as it waits for some business event to occur or for proper timing, etc. An activity that is placed on hold is not scheduled; it is not sent to the scheduler and as such, would not be automatically assigned a resource to perform it.

An activity is automatically placed on hold when:

- The activity is [created in the field](#) and the crew indicated they intend to work it themselves. In this situation there is no need to request the scheduler to schedule it as the crew intends to complete the work themselves. If the crew eventually does not perform the work, the activity no longer stays on hold and is sent back for [scheduling](#).
- The activity is [created in the field](#) and the crew does not plan to work it but the activity requires an appointment to be made. The system sends the new activity to the host system and waits for the host system to arrange an appointment with the customer. When the host system sends in the confirmed appointment, the activity no longer stays on hold and is sent back for [scheduling](#).
- Custom business rules are added to prevent the activity from being scheduled.
- A dispatcher may manually place an activity on hold as needed. When the hold is no longer needed, the user can manually transition the activity to be [scheduled](#).

Activity Identifiers

This section describes the various identifiers an activity may be associated with, including its internal unique activity identifier.

Host External Identifier

The host external identifier of an activity uniquely identifies a non canceled activity in the system by its identifier in the host system.

The identifier is provided by the host system when the activity is interfaced into the system and is used thereafter to identify subsequent host requests to update the activity's details or query its current status. Similarly, the system includes this identifier in all interim and final completion messages related to the requested activity that are reported back to the host system.

If the host system cancels an activity they have previously requested, the same host external identifier may be used to create a new activity in the system.

Host Alternate Identifier

The host may associate an activity with an additional non unique identifier for reporting purposes only. The system does not use this alternate identifier for any processing but also includes it on messages sent back to the host system.

Alternate Identifier

A short identifier provided by a dispatcher to easily communicate an activity to a crew in situations where that activity failed to be sent to the mobile device.

Refer to [Field-Referenced Activities \(FRAs\)](#) for more information.

Chain Related Identifiers

When the host sends in a set of activities in a [chain](#) dependency, each activity is associated with the host identifier of the entire chain. The system assigns a corresponding internal system generated identifier of the chain and posts that identifier on all activities in the chain.

Refer to [Chain](#) for more information.

Job Related Identifiers

When the host sends in a set of activities in a job dependency, each activity is associated with the host identifier of the entire job. The system assigns a corresponding internal system generated job identifier for each host job identifier and posts that identifier on all the activities in the job.

Refer to [Job](#) for more information.

Location

This section describes various aspects related to the location of where the work is to be performed.

Site Address

An activity must specify the address of the actual entrance to the site of service. This may be different than the [service address](#), where the work is performed.

The scheduler uses the site address to plan for the crew's travel time to and from the activity's location on the scheduled shift. It may use more or less travel time based on the [relative speed](#) of the vehicle(s) used by the crew on that shift.

The scheduler does not use the address itself but rather its geocoded value. Without the geocode of its site address an activity cannot be scheduled. If a geocode is not provided when the activity is created or when the address is changed the system attempts to geocode the address using the [geocoding service](#) defined on the installation options expecting an exact match.

NOTE: Logic to geocode the site address resides in algorithms associated with the activity's business object. Review these algorithms and adjust their match quality parameter as needed.

Site Identifier

The scheduler is encouraged to assign activities at the [same site](#) to the same crew shift. An activity is considered as being in the same site as another activity if they share the same site identifier.

In some cases, it is desirable for activities to be scheduled to the same crew when the activities are in the same area (site), but have different addresses (locations). For example, a campus, an apartment complex, or an office park would have various units with different addresses but are in close proximity.

The host system may send in a site identifier along with the activity details to group these activities into one site. If not explicitly provided, the system composes an activity's site identifier by concatenating a predefined list of the activity's site address constituents. By default all site address constituents are used.

Refer to [Site ID Configuration](#) for more information.

Site Delay

Site delay instructs the scheduler to factor in additional time (the amount of the site delay) immediately after the activity is planned to start, when planning the route of the assigned crew shift. For example, a crew may need time to park or gain access to a locked building after they travel to the site, but before they actually start work. Site delay is factored once for all activities [scheduled at the same site](#).

NOTE: Site delay cannot be used for [a complex activity](#).

Scheduling Activities At Same Site

When scheduled to a crew shift, same site activities are scheduled one after the other, with the scheduler considering the arrival, start, and completion times for each individual activity, but incorporates the site delay time, if specified, only once. Travel time between locations is calculated as usual for same site activities in different locations. Breaks can be scheduled between activities at the same site.

Service Address

By default the address for which service is provided is assumed to be the same as the site address. However there are situations where they may differ. For example, a large apartment building may have all meters located in a single location in the building and not inside the individual apartments. The crew would be directed to the meters' site address whereas the address of the individual apartment for which the work was requested would be provided as the service address for information only purposes.

Time Zone

The activity's time zone is the time zone of its geographic location, i.e. where work needs to be performed. All date and time information on the activity and its assignment portals is entered and displayed in that time zone.

Service Area

An activity is associated with a single service area that contains its location. The host system may send in the service or let the system default it based on the activity's address and type of service.

NOTE: Logic to determine the service area of an activity resides in a "Determine Service Area" algorithm on the activity's business object. Click [here](#) to see the algorithm types available for this system event.

Refer to [Understanding Service Areas](#) for more information.

Service Class

An activity's type may be associated with one or more service classes. A service class represents a broad category for work that may be used for various purposes.

Refer to [Designing Activity Types](#) for more information.

Estimated Work Duration

An activity specifies the estimated amount of time it should take a crew to perform the requested work. If not explicitly specified, this information is defaulted from the activity type.

The scheduler uses the estimated duration to book enough time on the assigned crew's shift for them to complete the work. The scheduler may allot more or less time based on the [relative efficiency](#) of the crew on that shift. Skill efficiency may also affect the scheduled time when applicable. Refer to [Skill Efficiency](#) for more information.

The nature of the work requested by a standard activity is such that its duration is short enough to be completed by a crew on a single shift. Longer work that requires multiple visits by one or more crews over several shifts is referred to as a complex activity.

Refer to [About Complex Activities](#) to better understand how complex activities differ from standard activities.

Time Windows

This section describes the various time windows associated with an activity.

At Least One Effective Time Window (ETW) Is Required

The host system requesting the activity must specify at least one time window during which the activity is in effect and may be scheduled. More than one ETW may be specified to state additional time periods during which the activity can be worked.

The earliest time window's start time defines the earliest time the activity can be scheduled and the latest time window's end time represents its [due date](#).

If the activity is an appointment it may have one ETW only reflecting the booked [appointment](#) time window.

If the activity is not worked by the end of the latest ETW it is considered [expired](#).

Due Date

There are times when the host system only states the earliest time an activity may be worked and relies on the system to automatically default when it is due. In this situation a single open ended ETW is sent with the activity and the system defaults its due date.

Activity type configuration controls the amount of time which activities of this type should typically be completed. If the activity type lacks the proper set up to support such defaulting, the activity is rejected.

Refer to [Default Due Date](#) for more information.

Multiple ETWs

By default all [ETWs](#) provided for an activity are equally preferable for scheduling. The host may specify a window cost with each ETW to apply a cost of using the time window relative to other time windows, thus establishing preference between the time windows.

Plan To Arrive Or To Complete?

By default an activity is scheduled such that the crew is planned to arrive at the activity location and start the work during an [ETW](#). The system does not necessarily account for the duration of the work to also be within the time window.

When it is critical that an activity is planned to start and complete within its [ETW](#) the host should send the activity's ETWs stating that completion time should be used as their Time Window Mode.

Preferred Time Windows (PTW)

The host system requesting the activity may optionally provide one or more preferred time periods when it is ideal to schedule an activity.

A preferred time window may not start earlier than the earliest [ETW](#) nor end after the latest. If specified, the scheduler is encouraged to prefer scheduling the activity during those times, and applies added cost when scheduling an activity outside of its PTW. The cost is defined on the activity type as the Preferred Time Factor.

A **PTW** may specify a service level agreement(SLA) flexibility time in minutes to define time before and after the time window within which an activity avoids incurring added cost.

For example, if the SLA Flexibility is set to 20 minutes, then scheduled arrival must be no more than 20 minutes before the start of this preferred time window or 20 minutes after the end of this window; otherwise, the Preferred Time Factor cost is applied.

When Can The Activity Be Actually Scheduled?

The host system provides the overall time frames during which the activity may be scheduled in the form of one or more **Effective Time Windows (ETW)**.

The following restrictions may prevent the activity from being scheduled during some part of these periods:

- Certain types of work may not be scheduled on **public holidays**.
- Certain types of work may only be performed on specific days of the week and within certain business hours. Refer to **work profile** for more information.
- Due to some upcoming event, for example an extreme temperature forecast, the business is required to prevent certain type of work within certain areas from being performed throughout the duration of the event. For example, there may be a government mandate that electricity meters cannot be cut off during days of extreme heat or extreme cold. Refer to **Override Time Windows** for more information on such events.

Due to such restrictions, the system has to identify portions of each ETW during which the crew can actually arrive to work the activity. These are known as **Arrival Time Windows (ATW)**. When no restrictions apply, an ATW covers the entire ETW.

In the same way the system has to identify portions of each ATW that overlap with any **Preferred Time Windows(PTW)** requested. These are known as **Service Time Windows (STW)**.

ATWs and STWs are automatically generated and stored along with the activity when the activity is created and are regenerated when time information or restrictions change.

The scheduler only considers an activity's arrival time windows and service time windows.

To visualize when an activity can be scheduled, the Schedule tab on the Crew Shift screen depicts all of the activity's time windows in a Gantt view.

NOTE: Logic to calculate an activity's arrival and service time windows resides in a "Update Time Windows" algorithm on the activity's business object. Click [here](#) to see the algorithm types available for this system event.

Arrival Time Window (ATW)

An ATW is a generated time window representing a period of time within an activity's ETW during which the activity is allowed to be scheduled, based on applicable restrictions.

The scheduler never schedules an activity earlier than the start time of an ATW.

The scheduler can however, schedule an activity with a late arrival time (outside of an ATW), though it is not likely. Unlike a too early schedule, lateness is cost driven and not a hard limit. The scheduler's cost control parameters are typically configured to discourage lateness globally.

An activity may further specify a cost of arriving late to this specific activity relative to other activities, to further discourage planned late arrival time. For example, if an activity has a relative late cost of 3 then the cost of late arrival for this particular activity is three times more than the late cost defined globally.

Service Time Window (STW)

An STW is a generated time window record representing time within an activity's **arrival time window** during which it is ideal to schedule the activity. An STW is a time overlap between an arrival time window and a preferred time window.

Appointments

An activity may represent an appointment made with a customer and therefore needs to be scheduled within the agreed-upon appointment window. Such activity is sent in from the host system with an indication stating it is an appointment along with a single [Effective Time Window\(ETW\)](#) set to the appointment window selected by the customer.

The following sections describe topics related to appointment activities.

Booking An Appointment

Before sending in an appointment activity, the host system calls the [appointment booking service](#) to request a list of available appointment windows for that activity. The list is presented within the host application and the customer chooses one of the available time slots. The host system then sends in the activity marked as an appointment along with the selected time window.

By default, time slots offered by the appointment booking service are those defined by the [default appointment booking group](#) referenced on the type of activity being requested. The host may override this and request to use another appointment booking group.

If the activity being booked is part of a [job](#), the request must include all of the activities in the job.

Conditional Booking

By the time the user makes a selection for a specific appointment window there is a chance that the selected window may no longer be available.

Therefore the host system sends in the activity with the selected time window but marks it a "conditional booking" pending the availability of the selected appointment window.

If the time window is still available, the activity is created as usual. However, if it is no longer available, then the activity is not created and the caller is notified accordingly.

Re-Booking

The system allows users to change an existing appointment in the host application. When the user requests a change to the appointment, similar to requesting an appointment for the first time, the host system works with the appointment booking service to present the user with a new set of available appointment windows to match the new booking details.

Typically the request to rebook an existing appointment is sent from the host as conditional upon the ability to accommodate the new booking instead of the original booking. As such, if the new booking is no longer available or the old booking can no longer be canceled, the rebooking attempt fails and the user is notified of the situation. Once rebooking is confirmed, the system cancels the original booking and schedules the new one instead.

If the original booking is already dispatched to a crew, the system cannot guarantee its successful recall from the crew as this requires human intervention. If the rebooking request states that it must be fully guaranteed, the rebooking attempt fails and the user is notified of the reason. If however the rebooking request can tolerate it not being guaranteed, then the system alerts a dispatcher to follow up with the crew on the cancellation of the original appointment but at the same time also schedules the new booking. If the dispatcher manages to cancel the original appointment in time, then it is canceled. If it is not canceled in time, both the original and the new booking are scheduled temporarily until the dispatcher cancels the new one.

If the activity was manually allocated to a specific shift or locked to its depot run by a dispatcher, a subsequent rebooking change frees the activity from these allowing it to be fully rescheduled.

When rebooking activities that are part of a job, the new job composition must fully replace the old job composition so that any activity that used to be in the old job and is no longer part of the new one needs to be explicitly listed as being canceled.

Reserve Shift Time For Appointments

In order to accommodate customer appointments up until the day before, crew shifts can be configured to reserve a certain percentage of their time for appointments so that other low priority work does not fill up the shifts.

While the scheduler can be configured to [displace low priority work](#) and give precedence to appointment work, last minute changes in the schedule may cause unnecessary disruption that can be prevented by setting appropriate shift reserves for appointment work.

Refer to [Reserve Capacity](#) for more information.

Displace Low Priority Work

The scheduler can be configured to ignore low priority activities currently scheduled to shifts so that new appointments can be offered to customers in their place.

Refer to [Activity Priority](#) for more information.

The scheduler may displace work that:

- Is non appointment work with a scheduling priority lower than a specific threshold. The appointment booking request as well as the conditional booking request may explicitly specify this threshold but if not specified, the threshold is defaulted from a corresponding scheduler parameter.
- Has enough remaining days to be completed before it expires. This is defined as a scheduler parameter.

Emergency Activities

Emergency activities must be scheduled and worked as soon as possible. They typically represent a hazardous situation that has to be addressed immediately. As such, an emergency activity is scheduled to an available crew at all costs and dispatched to their mobile device using an emergency queue that bypasses any other non emergency task.

The following sections describe topics related to emergency activities.

Emergency Queue

If the activity's queue is set to emergency, the activity is considered an emergency activity. This information is defaulted from the activity type but may be overridden at the activity level. This allows activities of non emergency type to be considered an emergency for any business reason, and vice versa.

The mobile device polls for messages placed in the emergency queue before any in the normal queue. All messages sent on behalf of an emergency activity are placed in the emergency queue so they would reach the device prior to any non emergency queued messages.

Emergencies Are Scheduled Differently

The scheduler handles emergency activities differently:

- They do not have a [scheduling priority](#) as they are more important than the highest priority work.
- The time window specified on the activity is irrelevant.
- The scheduler may be allowed by configuration to assign an emergency activity to a crew that is already working another task if they can address the situation faster than any other crew.

NOTE: Emergency activities are not eligible to be scheduled to capacity contractors. They may only be scheduled to internal crews.

While emergencies take precedence over non emergency work, if there is a lot of emergency work, unnecessary same day changes and disruption to crews' schedules can be avoided by reserving capacity on shifts for emergency work.

Refer to [Reserve Capacity](#) for more information.

Unacknowledged Emergency Alert

The system may be configured to raise an alert when a crew fails to acknowledge taking on an emergency activity. In these situations every minute counts. Instead of waiting for the alert to be raised by the alert type monitor background process, the system generates the alert in advance, right when the activity enters the system. However, the alert is set to only take effect some number of minutes in the future, allowing some time for the crew to acknowledge the activity (which also closes the alert). If the crew fails to acknowledge the activity before the alert's effective time, the alert appears on a dispatcher's alert queue ready to be addressed in a timely manner.

NOTE: Raising such a alert is optional. An alert is only created if an active alert type is configured and referenced as the **Unacknowledged Emergency Alert Type**[global configuration](#) parameter.

Refer to [About Alerts](#) for more information.

Activity Priority

The scheduler prioritizes non emergency activities based on their scheduling priority.

Scheduling priority is a numerical value from 0.01 to 10 that establishes the importance of the activity relative to other activities, where 10 is the highest priority. Activities with a higher priority are scheduled versus not scheduled, scheduled to an earlier shift, or scheduled earlier in the shift.

For example, if there are two connect activities, one at a hospital with a higher priority and one at an seldom used warehouse with a lower priority, the higher priority hospital activity will get scheduled today and earlier in the day, whereas the lower priority one may not get scheduled on that day at all.

The following sections describe further aspects of activity prioritization.

Scheduling Priority Over Time

An activity's scheduling priority is typically [defaulted](#) by the system when the activity is created but can be adjusted based on business rules.

The scheduler applies a [factor](#) onto the activity's scheduling priority to calculate its priority over time. It is this calculated scheduling priority over time, and not the initial value, that is used by the scheduler to prioritize work. The calculated scheduling priority is known to the scheduler only.

Default Scheduling Priority

An activity may optionally capture a business priority number representing its business value in terms of cost, other currency amount, number of customers impacted, relative customer importance, etc.

The system uses the activity's type configuration as well as that business priority number, if provided on the activity, to determine the activity's scheduling priority based on real business numbers.

Refer to [Determine Activity Priority](#) to better understand how the system sets the default scheduling priority for an activity.

Though not typical, the host system may send in the scheduling priority along with the activity in which case the system would simply use the provided value, ignoring the business priority number if also provided. Note that the scheduler only acts upon activity's scheduling priority. When an activity is created, the priority is defaulted from the activity type unless it is specified by the host. However, the priority on the activity itself is what the scheduler ultimately considers.

This also applies for appointment booking requests. If both values are included, the scheduling priority overrides the business priority number, otherwise the scheduling priority is defaulted as described.

Priority Profile

The scheduler applies a factor onto the activity's scheduling priority to calculate its priority over time. The factor is defined as a function over the activity's broad time window to make it stronger, weaker, or unchanged as needed.

For example, short-cycle, SLA-oriented work is critical at the beginning of the window and diminishes after that. Maintenance work, such as meter inspections, is low priority in the beginning of the window and becomes critical toward the end.

The activity's type references the Priority Profile that defines the function appropriate for activities of this type.

Refer to [Defining Priority Profiles](#) for more information.

Mandatory vs. Optional Work

An activity states whether or not it is mandatory to schedule it by its due date regardless of cost. If not mandatory, scheduling the activity is considered optional and is driven by cost only. As such, optional work may eventually remain not worked by its due date and expires.

Emergency activities are always considered mandatory.

Mandatory work is treated optional until the activity's end time window is within the scheduler horizon, except if the activity is a single day activity. Mandatory single day activities are never treated as optional.

This indication is defined on the [priority profile](#) referenced on the activity's type but can be overridden on the activity.

Resource Capabilities

An activity may require the mobile workers of the assigned crew to have certain work related capabilities in order to perform the work. These may be skills or activity specific requirements such as the need for security clearance, building access, drug testing, permits, language proficiency, etc.

In the same way, an activity may require the assigned crew's vehicle to have certain pieces of equipment in order to perform the work. For example, the crew may need a bucket truck to perform line pole work.

The scheduler assigns an activity to a crew that is capable of carrying it out on a specific shift.

Refer to [Crew Capabilities](#) for more information.

The following topics describe how capacities may be used to influence the activity's crew assignment.

Required Capabilities

An activity may require the crew to possess a specific capability, worker or vehicle related.

The scheduler only considers crews that possess this capability on a specific shift when looking to find a compatible crew shift for an activity. If none exists the activity remains not assigned.

If however the activity is manually allocated to a non compatible crew on a specific shift, the scheduler honors that allocation.

Preferred Capabilities

An activity may prefer the crew to possess a specific capability, worker or vehicle related.

The scheduler is encouraged by cost configuration to prefer crews that possess this capability on a specific shift when looking to find a compatible crew shift for an activity.

Being a preference only and not a requirement, the scheduler consideration is cost based. The scheduler may assign the activity to a crew shift that does not possess the preferred capability if none exist or the selected shift is all in all less costly than a shift with the preferred capability.

Not Allowed Capabilities

There are situations where an activity may require that the crew not possess a specific capability, worker or vehicle related.

NOTE: This type of requirement does not support worker skills.

The scheduler only considers crews that do not possess this capability on a specific shift when looking to find a compatible crew shift for an activity. If none exists the activity remains not assigned.

If however the activity is manually allocated to a non compatible crew on a specific shift, the scheduler honors that allocation.

Default Capabilities

Resource capability requirements are typically the same for all activities of the same type and as such are configured on the activity type. By default, an activity's resource capability requirements are those defined on its activity type. A specific activity may [override](#) the default list by either extending it or completely replacing it.

Override Capabilities

An activity states whether it uses the default resource capabilities as defined on the activity type, extends the default list, or completely overrides and replaces the list.

When extending, the resource capabilities specified on the activity are used by the system in addition to those defined on the activity type. For example, if a customer prefers a mobile worker that speaks Spanish, the activity for that customer may be sent in with an indication that resource capabilities should be extended to also include a preference for a "Speaks Spanish" skill. When scheduling the activity, the system considers all resource capability requirements specified on the activity type as well as those on the activity.

When overriding the default list altogether, the system only uses the resource capabilities specified on the activity to schedule the activity, ignoring any defined on the activity type. For example, assume a business is installing various products, yet the skills and equipment needed to install products of category A is different from those needed to install products of category B. Rather than having to set up separate installation activity types by product category, it may be efficient to simply define a generic "Install Product" activity type and specify the capabilities needed to install the specific product based on its category on the individual activity.

Full override can also be used to indicate that the specific activity requires no resource capabilities at all when the activity type specifies some.

Resource Restrictions

An activity may specify certain restrictions that further limit the resources that may be considered for performing the work.

The following describes various options affecting the resources the scheduler considers for an activity.

Allowed Crews

Typically the host system relies on the scheduler to find an appropriate crew to work an activity.

There are situations, known to the host system, where an activity may only be worked by a specific list of crews. In this case the activity is sent in with those crews specified as the allowed crews.

If there are no crews listed, all compatible crews are considered by the scheduler as usual. When specified, the scheduler narrows the list of compatible crews to those in this list. If none of the allowed crews are compatible with the activity, the activity remains not assigned.

If however the activity is manually allocated to a crew's shift where the crew is not on that list, the scheduler honors that allocation.

Excluded Mobile Worker

There may be situations, known to the host system, where one or more mobile workers are to be prevented from being assigned an activity. For example, a negative experience a customer had in the past with a specific worker.

When specified, the scheduler excludes from consideration any shift that lists the specified worker(s) as its crew members.

Excluded Shifts And Contractor Capacities

Over time, an activity may accumulate a list of crew shifts (or contractor capacities) it was once assigned to but was explicitly removed from and should not be assigned to going forward.

These are automatically added to the activity by the system when

- A crew on a specific shift declines taking on the activity for various reasons.
- A dispatcher takes action to remove an activity from its currently assigned shift (or contractor capacity).

When scheduling the activity, the scheduler excludes from consideration any shift (or contractor capacity) specified on that list.

Manually allocating the activity back to an excluded shift (or contractor capacity) automatically removes it from the exclusion list.

NOTE: Exclusions may not be sent in along with the activity from the host system.

Allowed Contractors

By default, for activities that are eligible to be worked by contractors, the scheduler may also consider compatible resources from various contractor companies, in addition to internal company crews in the scheduling process.

There are situations, known to the host system, where such activity may only be worked by a specific list of contractor companies. In this case, the activity is sent in referencing those companies in the allowed contractors list. When specified, the scheduler narrows the list of compatible crew shifts (or contractor capacities) associated with the allowed contractor companies. If no compatible resource is found, the activity remains not assigned.

If however the activity is manually allocated to a contractor crew's shift (or a contractor capacity) that is not associated with the allowed contractor companies, the scheduler honors that allocation.

Refer to [About Contractors](#) for more information.

Preferred Resource Groups

An activity that is eligible for contracting may list the preferred order for scheduling the activity between the company and the contractor companies .

Refer to [Contractor Eligibility Rules](#) for more information.

NOTE: Preference is determined and captured periodically while the activity is in the "Being Scheduled" state. Logic to determine these preferences for an activity reside in a "Contractor Eligibility" algorithm on the activity business object. Click [here](#) to see the algorithm types available for this system event. Preference may not be sent in along with the activity from the host system.

Excluded Contractors

Over time, an activity that is eligible for contracting may accumulate a list of contractor capacities it was once committed to but should not be considered again for various reasons.

A contractor company may be excluded from working an activity when

- A capacity contractor company did not approve taking on the activity in a timely manner or they have explicitly declined taking on the activity.
- A dispatcher takes actions to **exclude** an activity from its currently assigned contractor company.

When scheduling the activity, the scheduler excludes from consideration any contractor company specified on that list.

Manually allocating the activity back to a crew shift or capacity affiliated with an excluded contractor company automatically removes it from the exclusion list.

Refer to [About Contractors](#) for more information.

Scheduling An Activity

The purpose of the scheduling process is to assign an activity to a compatible crew on a specific shift (or a contracted capacity offered by contractor companies). Refer to [Scheduling Work To Capacities](#) for more information.

To plan the best work schedule and route for each shift, the scheduler matches activity details such as location, duration, the time frame available and capabilities the assigned crew may or not need to have in order to perform the work with information about the location, capabilities and availability of crews. A dispatcher may override scheduler decisions at any time.

When an activity is scheduled to a crew shift it is also assigned a work sequence representing the order in which the crew is expected to work their schedule on that shift.

Common reasons for why an activity remains unscheduled may be

- No scheduler covers the activity's location
- The activity is placed on hold or has data issues preventing the system from sending it to the scheduler
- No compatible resource is available that matches the activity's scheduling requirements
- The activity has expired and the activity type does not allow further time window extensions
- The activity is optional and may be pushed out by higher priority work

Until the activity is queued for dispatch, its schedule is constantly being optimized by the scheduler. The activity may be scheduled to a shift (or a contractor capacity) for a while, rescheduled to another, become free, rescheduled again, etc.

Refer to [Dispatching An Activity](#) for more information.

At any point in time if the scheduler cannot schedule an activity that otherwise should be scheduled, the activity is disabled from scheduling and the scheduling error is reported in the scheduler's registry record for the activity. The system provides information on the activity portal schedule tab to assist you in determining why an activity is not scheduled or disabled. Once the issue is resolved, the activity is automatically rescheduled.

For more persistent errors, the scheduler monitor batch process converts registry errors to To Do entries for a user to follow up. For dispatchers, the [To Dos In Scope](#) dashboard zone lists these entries for activities they are monitoring.

NOTE: The **Scheduler Monitor** batch process needs to run at least once a day to convert scheduling errors to To Do entries.

Expired Activities

If an activity has not been worked by its due date the scheduler stops attempting to schedule it.

Depending on configuration and business practices, activities of certain types may be configured to handle the situation differently (for example, the activity's time window can be automatically extended).

Refer to [Handling Expired Activities](#) for more information.

Activity Actions

A dispatcher or a business process may perform various actions on an activity that may impact the activity's state and schedule.

The following section describes these actions and the business flows associated with them.

Allocate

A dispatcher or a business process can allocate an activity to a specific crew's shift (or a contractor's capacity). While allocated, the scheduler honors the decision even if it violates resource compatibility rules, and keeps the activity assigned to this shift (or capacity) only. The scheduler may still optimize the order of the activity within the allocated shift.

If the activity is currently dispatched to another shift (or contractor capacity), the dispatcher is notified of the situation as it involves a recall of the activity from the currently assigned resource. Refer to [Forced Recall](#) for more information.

If the activity is [fixed](#) to a different crew than the requested shift's crew, the user is warned that the activity would no longer be fixed to that crew once allocated.

Activities that are allocated to a specific crew shift have a special visual cue when displayed in the CDI Gantt.

NOTE: Logic to handle the business rules associated with manual allocation resides in an "Allocate to Shift" algorithm on the activity's business object. Click [here](#) to see the algorithm types available for this system event.

Allocating an Activity Using "Chooser"

[Allocating](#) an activity using the "Chooser" service allows you to choose the shift (or contractor capacity) from a list recommended by the scheduler. The list is sorted by scheduling cost from lowest to highest, but you can click on any column to sort as needed.

You can relax some of the scheduling restrictions so that more shifts (or contractor capacities) are considered by the scheduler. For example, removing equipment or attribute compatibility allows the scheduler to return shifts that do not possess the worker attributes or equipment that were indicated as required to perform the activity. A maximum number of shifts (or contractor capacities) to return can be specified to prevent the scheduler from showing an unmanageable number of results.

Remove Allocation

A dispatcher or a business process can remove an existing [allocation](#) restriction from an activity making it no longer allocated to a specific shift. If the activity was [locked to a depot run](#), that restriction is also removed.

Removing an allocation restriction does not impact the current schedule of an already dispatched activity. If the activity is ever rescheduled for some reason, the scheduler would consider any compatible crew.

Fix To Crew

A dispatcher or any business process can fix the activity to a specific crew with the purpose of restricting the scheduler to assign the work to shifts of this crew only. In the same way, they may remove such restriction and indicate the activity as no longer fixed to any crew.

Locking an activity to a specific crew overrides any other restriction around which crews are [allowed](#) to perform the work, if specified. That means an activity can be fixed to a crew that is not in the allowed list.

Fixing an activity to a crew or removing such restriction does not impact the current schedule of an already dispatched activity. It only impacts the activity if it is ever rescheduled.

If the activity is [allocated](#) to a shift of a different crew, the user is warned that the activity would no longer be allocated to that shift once it is fixed to this crew.

A different flavor of this same functionality allows crews to [keep an activity](#) with them for future shifts.

Activities that are fixed to a crew have a special visual cue when displayed in the CDI Gantt.

NOTE: Logic to handle the business rules associated with fixing an activity to a crew resides in a "Fix To Crew" algorithm on the activity's business object. Click [here](#) to see the algorithm types available for this system event.

Manual Dispatch

Activities are automatically queued for dispatch based on crew shift (or contractor capacity) configuration. The queue represents a list of activities ready to be sent to the crew once they start their shift, if not already started (or communicated to the contractor company at the agreed upon notification time).

A dispatcher may manually place an additional activity in that queue that is otherwise scheduled to a crew's shift (or contractor capacity), but has not yet been placed in the queue by the scheduler. This action is also applicable to an activity that is allocated to a crew shift but the scheduler has not yet scheduled the activity to that shift.

The scheduler responds to a manual dispatch request by adjusting the shift's schedule such that the manually dispatched activity is added last to the queue of already dispatched tasks.

Refer to [queue for dispatch](#) for more information.

Reschedule

This action may be used to reschedule an activity that is currently in the [queued for dispatch](#) state. It effectively reverses a dispatch action taken on an activity, and indicates to the scheduler that the activity is no longer locked to its current location in the schedule.

If the activity is manually allocated, the scheduler cannot schedule the activity to a different shift but can still optimize its location in the schedule. If the activity is not manually allocated, the activity can now be scheduled to any crew shift (or contractor capacity).

Unassign

A dispatcher may unassign an activity from the crew shift (or contractor capacity) it is currently assigned to and send it back to be [rescheduled](#).

In order to prevent the scheduler from assigning the activity back to the same shift (or contractor capacity) from which it was taken, system adds the shift (or contractor capacity) to the activity's list of [excluded shifts and contractor capacities](#).

If the activity is currently manually allocated to that shift (or contractor capacity), the allocation is removed.

If the activity is currently dispatched, the dispatcher is notified of the situation as it involves a recall of the activity from the currently assigned resource. Refer to [Forced Recall](#) for more information.

NOTE: Logic to handle the business rules associated with unassigning an activity resides in a "Unassign Activity" algorithm on the activity's business object. Click [here](#) to see the algorithm types available for this system event.

Defer

A dispatcher may need to put off the scheduling of an activity to a later date.

If the activity is currently dispatched, the dispatcher is notified of the situation as it involves a recall of the activity from the currently assigned resource. Refer to [Forced Recall](#) for more information.

To un-defer an activity, the same action is used providing no deferment date.

NOTE: Logic to handle the business rules associated with deferring an activity resides in a "Defer Activity" algorithm on the activity's business object. Click [here](#) to see the algorithm types available for this system event.

Place On Hold

A dispatcher may need to place an activity on hold if something needs to be done before work can be started. Putting a hold on an activity removes it from consideration for scheduling until the hold is manually removed.

Refer to the activity [On Hold](#) state for more information.

Lock Activity To Depot Run

This action is only relevant to a depot related activity.

The system locks a depot related activity to its currently assigned depot task when a user or a process closes the depot task. A depot task represents the scheduled visit to the depot to pickup or drop off goods for one or more activities.

Refer to [About Depot Runs](#) for more information.

NOTE: Logic to handle the business rules associated with locking an activity to the depot run it is currently scheduled to resides in a "Lock Depot Run" algorithm on the activity's business object. Click [here](#) to see the algorithm types available for this system event.

Exclude Contractor

A dispatcher or a business process may exclude the contractor company from working an activity committed to them. When possible, the contractor company is automatically informed of the recall. Refer to [Forced Recall](#) for more information.

The excluded contractor company is added to the activity's excluded contractors list and the scheduler would no longer consider any shift (or contractor capacity) that belongs to that contractor company.

Refer to [Excluded Contractors](#) for more information.

NOTE: Logic to handle the business rules associated with excluding the contractor the activity is currently committed to resides in a "Exclude Contractor" algorithm on the activity's business object. Click [here](#) to see the algorithm types available for this system event.

Revert To Company

This action marks an activity as allowed to be performed only by company resources. It may be used when an activity is eligible for contracting but for some reason the dispatcher needs to revert it back to company only crews.

If the activity is currently dispatched, the dispatcher is notified of the situation as it involves a recall of the activity from the currently assigned resource. Refer to [Forced Recall](#) for more information.

Forced Recall

If the activity has already been dispatched and the requested change should recall it from the currently assigned crew (or contractor company), a warning message notifies the dispatcher that the activity needs to be recalled. The dispatcher may either confirm and proceed with the action or cancel the action.

If confirmed, the system completes the requested action right away. In addition, the system attempts to recall the activity from the currently assigned crew in an effort to avoid as much as possible a situation where more than one crew is sent to perform the same activity. Refer to [Concurrent Assignments](#) for more information.

In the same way, the system sends a message to the external contractor company notifying them of the recall. This is only applicable if the contractor company's system is integrated with your system.

NOTE: The system does not wait for the activity to be actually be recalled to perform the requested action. The recall request may or may not succeed with returning the activity, depending on the assigned resource's actual progress with

working the activity as well as their connectivity. It is the responsibility of the dispatcher to coordinate the recall with the currently assigned resource.

Refer to [Recall](#) for more information on how recalls are handled.

Dispatching An Activity

The following section describes key concepts related to the process of dispatching an activity.

The Dispatch Process

The process of dispatching an activity to the assigned crew (or communicating it to the contractor company) involves multiple steps.

- **Step 1 - Placed in queue.** The activity has to be placed in the queue for dispatch for the dispatch process to begin. This step is typically performed by the scheduler as it auto dispatches work, but other processes may do so as well.
- **Step 2 - Wait.** The activity remains queued until it is time to actually dispatch it.

Refer to the [Queued For Dispatch](#) for more information about the processes involved in these steps.

- **Step 3 - Dispatch.** Once all conditions are met for the activity to be dispatched, the system creates an assignment for activity and the assigned crew (or capacity contractor), and sets the activity to the "Assignment In Progress" state where it waits for the outcome of this assignment. A copy of the assignment is sent to the assigned crew (or capacity contractor).

Refer to the [Assignment In Progress](#) for more information about the processes involved in this step.

- **Step 4 - Acknowledgement.** The activity is considered dispatched if the assigned resource acknowledges receiving their copy of the activity assignment.
- If the activity is assigned to a crew, their assignment needs to be set to the "Dispatched" state. Refer to [Automatic vs. Manual Acknowledgment](#) for more information.
- If the activity is assigned to a contractor capacity, the contractor company has to approve taking on the activity by setting their assignment to the "Approved" state. Refer to [Managing Committed Work](#) for more information.

Refer to the [Assignment Workflow](#) for learn more about how the acknowledgement step.

Disabling Auto Dispatch

Certain types of activities may require an approval from a dispatcher before they can be automatically dispatched by the scheduler to a crew.

By default all activities are allowed to be automatically dispatched by the scheduler. If the activity type explicitly states that auto dispatch is not allowed, the scheduler would still lock the schedule of an activity of this type when it is time to queue the activity for dispatch, but the system would keep the activity in the being scheduled state, waiting for a dispatcher to [manually dispatch](#) it.

Once a dispatcher manually dispatches such activity, it set to queued for dispatch and the [dispatch process](#) progresses as usual.

Refer to [Controlling Auto-Dispatching](#) for more information about this process.

Allow Activities To Break Sequence Locking

When the crew's shift uses standard drip mode, they only have access to their next few tasks on the mobile device and are expected to work the tasks in their route sequence order.

By default, the scheduler locks the sequence of dispatched tasks for shifts configured this way so that the route communicated to the crew remains unchanged. The scheduler would not dispatch a task to be worked prior to another already dispatched task, nor would it switch the order between already dispatched tasks.

You can configure the shift to deviate from this default and allow the scheduler to make such changes to the dispatched portion of the route.

Refer to [Controlling Auto-Dispatching](#) for more information about the various options available on the shift to control dispatched route changes.

There are situations though where even if the shift is configured to not allow route changes, certain activities should be allowed to be dispatched and injected before already dispatched tasks. The indication to ignore sequence locking is set on the activity and is defaulted from the activity type, if not explicitly specified. A dispatcher may directly update an activity to allow this as needed.

If the activity is marked as allowed to ignore sequence locking, the scheduler may dispatch the activity to be worked before another dispatched task as needed.

Completion Information

When an activity is completed by the assigned resource a completion message is sent to the host with all completion information, including any [attachments](#) captured in the field.

NOTE: Logic to compose the completion message for an activity resides in a "Prepare Completion Data" algorithm on the activity's business object. Click [here](#) to see the algorithm types available for this system event.

Activity Cancellation By Host

The host system may send a request to cancel an activity they have previously asked to perform with the intention to prevent the work from being performed.

If the request reaches the system when the activity is already completed then there is no way to cancel the activity and the host is notified.

If the activity has not been dispatched yet, it is [canceled](#) right away.

If the activity is dispatched to a crew (or to an external contractor company) they may have already performed the work in the field and their completion update is yet to be reflected on the server. In this situation, the system cannot cancel the activity without ensuring the crew (or the external contractor company) would indeed not perform the work, for which an alert is raised for a dispatcher to follow up. The immediate response to the cancellation request to the host system indicates that the activity cannot be canceled right away but the request is being forwarded to a dispatcher.

NOTE: The system creates the alert using the active alert type referenced as the **Activity Cancellation Alert Type** [global configuration](#) parameter.

Meanwhile

- If the crew uses a mobile device, the system automatically sends a [recall](#) message to the crew's device requesting to return the assignment if still possible. Refer to [Recall](#) for more information on how this is processed.
- If and when the activity is recalled, it automatically gets canceled instead of being rescheduled. A completion message is sent to the host indicating the activity was canceled. The dispatcher alert to follow up on this cancellation request gets automatically closed as well.
- If the activity was not automatically recalled in a timely manner, the dispatcher assigned to address the alert follows up with the crew.

- If the crew (or external contractor company) did not perform the work yet, they would decline the activity. This would automatically cause the activity to be canceled, similar to being recalled.
- If for some reason it is not possible to wait for the decline message to reach the server, the dispatcher can manually force the cancelation of the activity by choosing the [Forced Cancel](#) action on the alert raised to handle this situation.
- If the crew (or external contractor company) has performed the work already, the activity cannot be canceled. The host system would receive the completion information as usual.

Depot Related Activities

When an activity involves a transfer of goods between a specified location and a depot, the activity specifies the list of depots from which goods should be delivered or to which goods need to be returned.

The scheduler plans the route such that it also includes a trip to the depot to pick up goods before a delivery activity or to return goods picked up by a collection activity.

Refer to [About Transfer of Goods](#) for more information.

Handling Keys

If a key or card is needed to gain access to a work site or facility, the crew cannot start such activity without it. The crew needs to pick up the key before traveling to the work site. These activities need to specify the key identifier as well as the key location in order to provide information about where the crew can pick up the keys to gain access to the work site.

Keys are typically located at common locations such as service centers or depots. As such, key pick up is handled using depot functionality where the scheduler creates a depot task to pick up the key as part of scheduling such activity. A distribution depot must be set up for every location from which keys would be picked up.

If the crew has multiple activities that require them to pick up keys at the same location, the scheduler creates one task to pick up all of those keys when possible. Using the mobile application, when at the depot the crew is presented with a list of the activities that need keys and the description of each key to pick up.

Refer to [About Depot Runs](#) for more information.

Once the crew has the key or card, the activity should not be assigned to another crew or resource. If a dispatcher attempts to change the assigned crew, he or she receives a warning.

Activities that require keys may be scheduled to crew shifts as well contractor capacities. However, when scheduled to shifts, they are restricted to shifts that have the same logoff location as the location where keys are picked up to allow the crew to return the keys at the end of their shift. This is assumed to occur as part of the your business practice as the system does not validate whether or not a key was actually returned by the crew.

NOTE: Activities already associated with depot to pick up or drop goods cannot use keys since an activity can only be in a single depot run. This means that there cannot be depot activities that require "goods" to be picked up from depot A and require "keys" to be picked up from depot B. Then the activity would have to be in two depot runs to pick up both goods and keys.

About Complex Activities

Activities that require multiple shifts to complete are referred to as complex activities and are typical of, but not limited to, construction work. They are also sometimes referred to as long cycle work.

Complex activities are similar to regular activities in many aspects. It is important that you familiarize yourself with how regular activities are managed in order to better understand complex activities.

Refer to [About Activities](#) for more information.

NOTE: Transfer of goods, appointment booking, conditional booking, and site delay cannot be used with complex activities. They can also not be used for emergency work and are not eligible for capacity contracting. Refer to the [Capacity Model](#) for more information about capacity based contracted work.

This section describes concepts related to managing complex activities.

A Complex Activity Is Scheduled Differently

A complex activity is too long to complete during a single crew's shift. As such, it cannot be scheduled like a regular activity. The scheduler splits the complex activity into several crew visits to the work site and progress is tracked per visit. Each visit is a new task record created by the scheduler and scheduled to a crew's shift, very much like a regular activity.

When a crew completes a visit task, they can indicate whether this visit has completed the entire work or if more work is still needed they can adjust the estimated remaining time. A dispatcher may also update the complex activity in the same way.

With each visit completed, the scheduler re-optimizes the whole complex activity. If the entire activity is not completed yet, future planned visits are adjusted based on the remaining time indicated. If more time is needed to complete the activity, the scheduler may create more visits and/or extend the last visit accordingly. If less time is needed, the scheduler may delete future visits and/or decrease the duration of the last visit as needed. Once the complex activity is complete, remaining future visits are removed from the schedule.

Complex activities are considered mandatory work.

The following sections describe topics related to scheduling a complex activity.

Refer to [Use Case Example](#) for an example of how a complex activity is scheduled.

The Scheduler Creates Visit Tasks

Visit tasks are solely created by the scheduler when scheduling a complex activity.

Crews assigned to perform this type of complex work are assumed to be busy performing that work for their entire shift. They may occasionally be asked to take on emergency or other priority tasks, but for the most part, they are asked to perform only the complex activity for the day. Therefore, the scheduler assumes one visit would be needed per shift.

The complex activity type defines the average and minimum site time per visit. The scheduler splits the long complex activity into visits of average length and a last visit in the duration of the remaining time but no less than the minimum time requirement which may be used, for example, to allow the crew some time for setup and tear down.

As an example, assume a complex activity is set up with an estimated duration of 20 hours. The activity type defines a minimum visit time as 3 hours and average visit duration as 6 hours. The scheduler splits the complex activity into 4 visits, 3 visits of 6 hours each and uses 3 hours for the last visit because the minimum visit duration must be at least 3 hours. If the complex activity duration was 23 hours then the last visit would be of 5 hours.

The number of visit tasks created is determined solely based on the complex activity's remaining duration and the average and minimal durations defined on the task type. The scheduler reassess the number of visits needed as well as adjusts the last visit duration only when the complex activity's [remaining duration](#) changes. When the remaining duration changes, the scheduler may add new visits, delete no longer needed visits, or adjust the last visit's duration.

Unlike activities that have a predefined duration, the amount of time scheduled on a shift to work a visit task may differ from the duration used to create it. That is because a visit task is designed to consume the entire time available on the shift when possible. Nonetheless, only the visit's duration at creation time is considered by the scheduler, and not the visit's scheduled duration, when it comes to creating new visits or adjusting the last visit duration.

Refer to [Visit Task Duration](#) to better understand various aspects related to the duration of a visit task.

Visit tasks are created independently from any attempt to schedule them to a shift and may remain unscheduled if there are not enough compatible shifts to work them all.

Refer to [Scheduling Visits](#) for more information on how these visits are scheduled and adjusted over time.

Remaining Duration

The remaining duration of a complex activity refers to the time left to complete the entire work. The system recalculates a complex activity's remaining duration when

- A crew completes a visit task assignment and provides the actual time worked
- A crew completes a visit task assignment and extends the time remaining to complete the work based on their assessment in the field
- The host system provides a correction to the overall duration of the complex work activity
- A dispatcher extends the overall duration of the complex work activity

The scheduler reassesses the need for new visit tasks, removes no longer needed tasks, or adjusts the last visit duration only on these events when the remaining duration of a complex activity changes.

Visit Task Duration

Unlike activities and other tasks, a visit does not have a fixed duration before it is dispatched.

A visit task is created with a target duration which may be viewed as a close estimate of how long the crew would work on the complex activity on a shift. All visits but the last one are created with a target duration set to the average duration defined on the complex activity type. The last visit is created with a target duration set to the remaining time but no less than the minimal duration also defined on the complex activity type.

Refer to [Creating Visit Tasks](#) to better understand when and how visit tasks are created.

A visit is anticipated to be the main task a crew is busy with on their shift. As such, it may consume more or less schedule time on the shift it is assigned to, but never less than the minimum duration specified on the activity type. Each visit task, except the last one, consumes the entire remaining time available to perform work on a shift. The last visit consumes at most its target duration.

This means that until a visit task is dispatched, its scheduled duration may differ from its target duration.

NOTE: The number of visits and the duration of the last visit is solely a product of target durations and never automatically adjusts to the optimized scheduled duration.

The scheduler is encouraged by cost control configuration to find a compatible shift with enough available time to accommodate the visit's target duration, when possible, but may schedule the visit to a shift with less time available. If a shift has more time available, then more time is scheduled to the visit.

As an example,

- Assume a complex activity is set up with an estimated duration of 20 hours. The activity type defines a minimum visit time as 3 hours and average visit duration as 6 hours.
- A visit was created based on the average duration with target duration of 6 hours.
- Assume it is scheduled to a shift of 7 hours planned duration and a planned break of half an hour. If travel to the activity location is an hour, the shift has 5:30 hours left to perform work. If no other tasks are scheduled to that shift, the visit is scheduled to consume 5:30 hours.
- In the same way, if that visit was scheduled to an 8 hour shift with same break setup, then 6:30 hours would be scheduled for that visit.

Target duration is a ballpark used to split the overall complex activity task to smaller visit tasks. As crews complete their visits and provide more accurate feedback as to the remaining time left to complete the work, then more or less visits are planned accordingly.

How close is the schedule duration compared to the target duration? If the average time specified on the activity type closely reflects average work time available on compatible crew shifts, the overall schedule of the complex activity may be

insignificantly under scheduled or over scheduled. As crews complete visits and reduce the remaining time still needed to be worked, the overall scheduled duration becomes more accurate.

It is important to define realistic average duration time on the activity type so that the difference between the target duration and the scheduled duration is negligible and becomes more so over time. When setting up average duration, average travel and break time need to be taken into account and subtracted from typical shift planned time.

When the available time on a shift is less than the visit's target duration, the complex activity is under scheduled. It is preferred to set the average time slightly lower than anticipated and err on having more time scheduled on a shift for a visit than risk it being under scheduled as this situation may call for user intervention to ensure the entire work is properly planned.

Refer to [Under Scheduled Alert](#) for more information about handling an under scheduled complex activity.

Scheduling Visits

Visits are scheduled sequentially to non overlapping shifts, one visit per shift, unless a user manually allocates them differently.

Visits can overlap for a certain amount of time, for example, to allow time for one crew to pass on or debrief another crew that is taking over the activity. The maximum percentage of shift time that shifts are allowed to overlap is defined on the "Maximum Visit Overlap" scheduler configuration parameter. By default the scheduler considers shifts that overlap by more than that limit as concurrent and as such may only assign a visit of the same complex activity to one of these shifts.

A Visit Task Is Not A Business Entity

A regular activity is requested by a host system and captures details about the requested work. A visit task on the other hand is created by the scheduler as a way to divide a long running complex activity into multiple visits to the activity location until entire work is complete.

As such, a visit task bears no business information of its own. All information about the work to perform is provided on the complex activity it belongs to. A visit task may be viewed as a way to track the schedule and progress of a small portion of work time out of a long running complex activity.

A Visit Task Is Worked Like A Regular Activity

From a crew's perspective, performing the work associated with a complex activity on a specific visit is no different than working on a regular activity. A visit task is scheduled and queued for dispatch and eventually dispatched to a crew in the form of an assignment, exactly like a regular activity.

Refer to [About Assignments](#) for more information.

Actual Work Duration

When completing a visit task's assignment, the crew reports back how much time they actually worked which then impacts the remaining duration of the entire complex activity. The crew is presented with a calculated work time set to the entire time they were on site and they may adjust it as needed to provide a more accurate work time report.

Complex Activity Workflow

Complex Activity states are depicted in the following diagram:

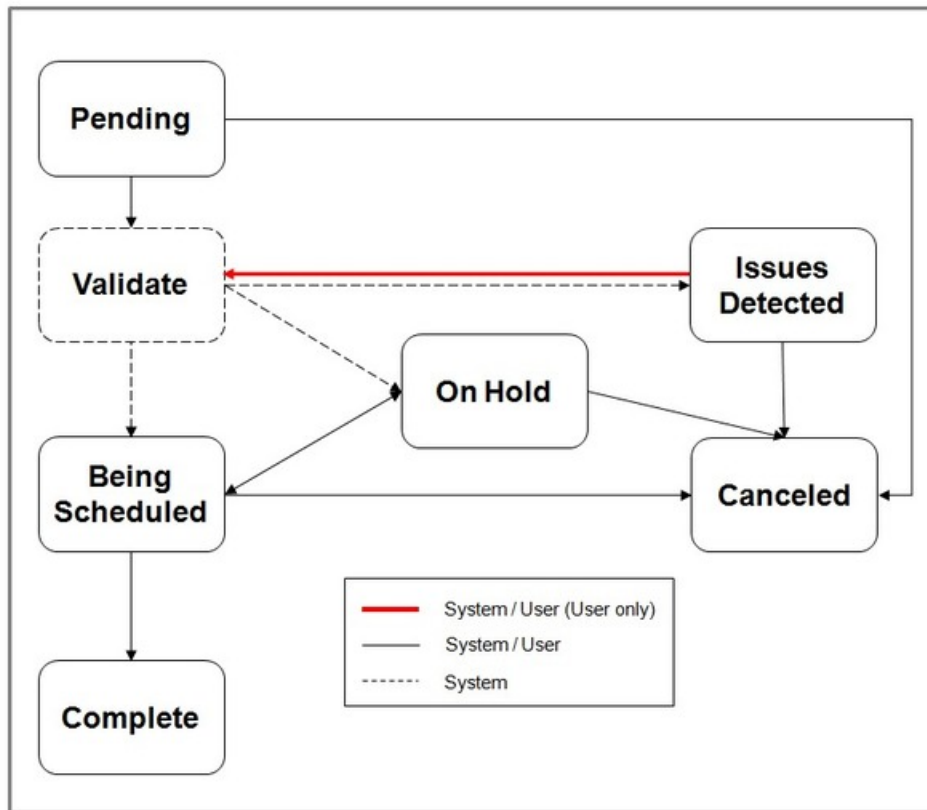


Figure 2: Complex Activity Lifecycle

The following describes key stages in the overall lifecycle of a complex activity.

Validation

A complex activity is validated very much like a regular activity.

Being Scheduled

Unlike a regular activity, a complex activity is not scheduled to a crew shift. The process of scheduling it involves splitting the activity into smaller visit tasks which are then scheduled to crew shifts.

Refer to [A Complex Activity Is Scheduled Differently](#) for more information.

While the complex activity is in the **Being Scheduled** state

- The scheduler [creates visit tasks](#) to cover the remaining duration of the complex activity as it gets updated and attempts to schedule them to crew shifts.
- The scheduler keeps improving on its scheduling decisions around the activity's visit tasks. During this continuous optimization phase, the activity's visits may frequently change their schedule until they are queued for dispatch, similar to a regular activity.
- Visit tasks are worked by crews. Refer to [Visit Task Workflow](#) for more information.
- The activity's time windows are checked and if the activity has expired, it may be automatically extended or [canceled](#) as per configuration on the activity type. Refer to [Expired Activities](#) for more information.
- The activity's time windows are recalculated to take into account new or canceled applicable override time window events, if any. Refer to [About Override Time Windows](#) for more information.
- Business rules may put the activity [on hold](#).

Completed

The activity is set to the **Completed** state when

- A crew completed a visit task assignment and indicated the entire complex activity work is complete. A completion message is sent to the host with all completion information, including any [attachments](#) captured in the field.
- The host notifies the system that the activity has been completed in the host system. This is not a typical situation but there are business practices where the crew may prefer to key in the completion information directly in the host system application. Since the host has the completion information already, there is no need to send them a completion message,

Regardless of how the complex activity is completed, any future visit tasks are deleted. If any are still dispatched to crews at that time, an alert is created to request a dispatcher to follow up with those crews similar to the process of [canceling a dispatched activity](#).

Refer to [Completion Information](#) for more information.

Canceled

The host system may send a request to cancel any further work on the complex activity. In this situation, any future visit tasks are deleted. If any are still dispatched to crews at that time, an alert is created to request a dispatcher to follow up with those crews similar to the process of [canceling a dispatched activity](#).

[Expired activities](#) may also be automatically canceled based on the activity type configuration.

When all visit tasks are finalized, a completion message is sent to the host with all completion information for work performed so far, including any [attachments](#) captured in the field.

Refer to [Completion Information](#) for more information.

On Hold

There are situations where the complex activity is sent in but it is not yet ready to be scheduled as it waits for some business event to occur or for proper timing, etc.

An existing complex activity that is already scheduled could also be placed on hold by the host system, a user, or any customer business rule.

While on hold, the complex activity is not scheduled. If it was being scheduled when placed on hold, it is removed from the scheduler and any future visit tasks are deleted. If any are still dispatched to crews at that time, an alert is created to request a dispatcher to follow up with those crews, similar to the process of [canceling a dispatched activity](#).

Visit Task Workflow

The workflow of a visit task resembles the workflow of a regular activity.

Visit states are depicted in the following diagram:

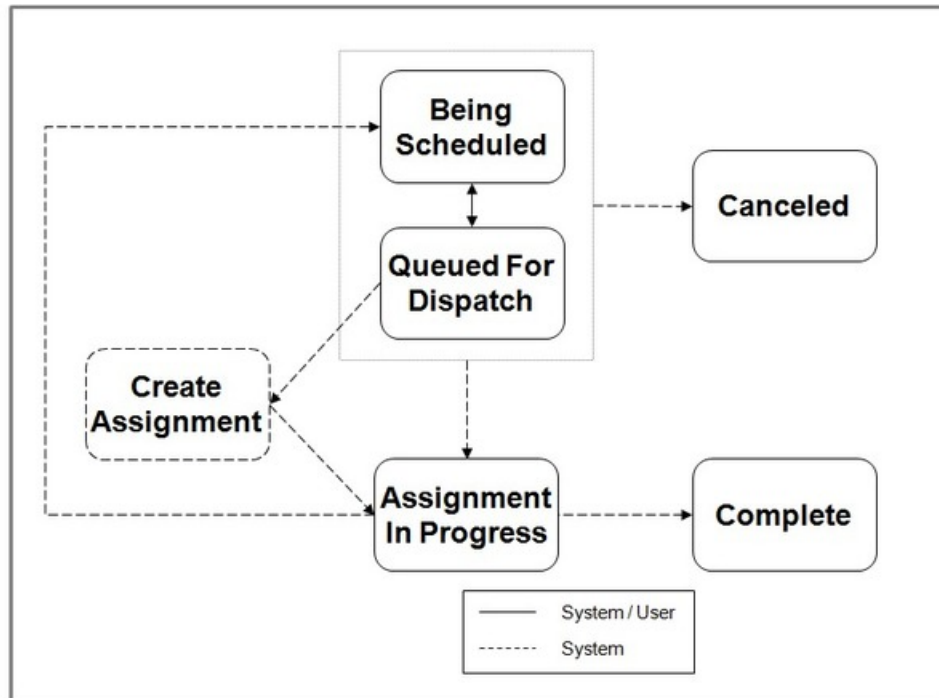


Figure 3: Visit Task Lifecycle

The following describes key stages in the overall lifecycle of a visit task.

Refer to [Activity Workflow](#) for more information.

Being Scheduled

A visit task is created by the scheduler in this state. Unlike a regular activity, a visit task does not need to go through a validation step as it contains no business information.

While the visit task is in the **Being Scheduled** state, the scheduler keeps improving on its scheduling decisions. During this continuous optimization phase, the visit task may be frequently assigned, reassigned, the order changed within its currently assigned shift's route, or become unassigned again.

Similar to the activity workflow, this goes on until it is time to queue the visit task for dispatch so it can be communicated and dispatched to the assigned crew.

Queued For Dispatch

A visit task is queued for dispatch exactly like a regular activity.

Assignment In Progress

A visit task is dispatched to a crew in the form of an assignment, exactly like a regular activity. Similar to a regular activity, the system waits for the resolution of crew assignments to assess the state of the visit task.

Completed

The visit task is set to the **Completed** state when at least one crew assignment for the visit is completed.

Unlike a regular activity, a complex activity typically requires multiple crews to complete the entire work and allows them to work concurrently if needed. As such, each crew provides their own completion information which is forwarded to the host system individually when they complete their assignment of a visit task and not when the visit task is completed. No completion is sent when the visit task itself is completed.

A final summary completion message is sent to the host when the entire complex activity is completed.

Refer to [Completion Information](#) for more information.

Canceled

The host system may send a request to cancel any further work on a complex activity or place it on hold. As part of this update to the complex activity, dispatched visit tasks may be canceled, similar to the process of [canceling a dispatched activity](#).

Use Case Example

Assume a complex activity is set up with an estimated duration of 20 hours.

Consider these parameters for this example:

- The activity type defines a minimum visit time of 3 hours and average visit time of 6 hours.
- Compatible shifts for this type of work are 8 hours shifts.
- Travel time to the activity location is 30 minutes each way, for a total 1 hour travel time.
- The crew is planned to take one 30 minute break on these shifts.

Initially the scheduler creates 4 visits: 3 visits of 6 hours average duration and the last visit of 3 hours minimal duration since the remaining duration of 2 hours is less than the minimum time required.

These visits are scheduled as follows

- Visit 1 scheduled to a shift on day 1.
- Shift available work time for the crew to perform the complex activity work during this visit is calculated as follows: 8 hours shift- 30 min break - 1 hour travel to activity location = 6:30 hours.
- Notice that even though the visit was created to consume 6 hours of work it actually consumes 6:30 hours of work because the crew is assumed to work mainly on this task for their entire shift.
- In the same way if only 5 hours were available on the shift because of other priority tasks scheduled to it then the visit would be scheduled 5 hours only. A visit would never be scheduled less than the minimum time required even if this means the crew is planned to work overtime.
- Visit 2-3 are scheduled to shifts in day 2 and 3 respectively.
- The shifts may be for same crew that started the work or another crew, depending on whether the complex activity requested that the [same crew](#) completes the entire work or not.
- Like visit 1 they are created with the target duration of 6 hours but are actually scheduled 6:30 hours.
- Visit 4, the last visit, is scheduled to a shift in day 4.
- Like visits 2 and 3 it may or may not be scheduled to the same crew.
- Unlike non last visits, the last visit cannot be scheduled more time than its target duration irrespective of the work time available on the shift. Even though 6:30 hours are available on the shift, the last visit is scheduled for 3 hours.
- Like all visits, it cannot be scheduled less than the minimal duration.

In the beginning of day 1, 3 visits are scheduled 6:30 hours each and the last visit is scheduled 3 hours which sum up to a total of 22:30 hours scheduled to work the entire complex activity.

With each visit task completion, the scheduler reassesses the remaining duration of the entire complex activity and adjusts the number of visits and/or adjusts the last visit accordingly.

- Day 1
- The crew assigned to perform visit 1 completed it and indicated they actually performed 9 hours of work as opposed to the 6:30 hours planned.

- The remaining work time for the complex activity is recalculated as 20 hours - 9 hours = 11 hours. The scheduler recalculates the number of visits needed to perform the remaining 11 hours as follows: 1 visit of 6 hours duration and a last visit of 5 hours duration.
- It now compares this with existing visits. There are 3 remaining visits out there, 2 of 6 hours assumed duration and the last visit of 3 hours assumed duration. Based on the new remaining duration there should now be only 2 visits: 1 of 6 hours and the last one of 5 hours. The scheduler therefore deletes one of the 6 hours visits and adjusts the last visit from 3 hours to 5 hours.
- When scheduled, visit 2 would be scheduled 6:30 hours and the last visit is scheduled 5 hours.
- Completion information captured by the crew on this visit is sent to the host system.
- Day 2
- The crew assigned to perform visit 2 completed it indicating they actually performed 5 hours of work as opposed to the 6:30 hours planned.
- The remaining work time is recalculated as 20 hours - 9 hours - 5 hours = 6 hours. The scheduler recalculates the number of visits needed to perform the remaining 6 hours and concludes it needs 1 last visit of 6 hours.
- It now compares this with existing visits. The last remaining visit is extended to 6 hours.
- When scheduled, the last visit would be scheduled 6 hours and not the 6:30 hours available because it is the last visit.
- Completion information captured by the crew on this visit is sent to the host system.
- Day 3
- The crew assigned to perform the last visit completed it indicating they worked 7 hours as opposed to the 6 hours planned, and that the entire work still requires 10 more hours.
- This update from the crew overrides the complex activity's remaining time to 10 hours. The scheduler recalculates the number of visits needed and creates 1 new visit of 6 hours duration and another last visit of 4 hours duration.
- The new visits are scheduled to days 4 and 5 respectively.
- Completion information captured by the crew on this visit is sent to the host system.
- Day 4
- The crew assigned to perform this visit completed it indicating they worked 9 hours as opposed to the 6 hours planned but now the entire complex work is complete.
- The scheduler deletes the unnecessary last visit.
- Completion information captured by the crew on this visit as well as a summary completion message for the entire complex activity is sent to the host system.

If the no crew indicates that the entire work is complete by the time the complex activity's time window expired, the scheduler disables the complex activity and it is no longer scheduled until the time windows are [extended](#).

Same Crew

Depending on the nature of work, either the same crew can perform the work from start to finish or multiple crews can perform the work. If same crew is to perform the work then the scheduler plans all visit tasks to be scheduled to shifts of the same crew.

This indication is defaulted from the complex activity type if not specified on the complex activity.

Under Scheduled Alert

An alert can be configured to notify a dispatcher when a complex activity is under scheduled beyond the configurable percentage threshold defined on the alert type.

A complex activity is deemed to be under scheduled if the percentage of the under scheduled amount of time as calculated by the scheduler and posted on its task schedule record relative to the remaining duration is greater than the threshold on the alert type.

Refer to [About Alerts](#) for more information.

Time Tracking

Typically the time a crew worked on site of an activity is automatically calculated from the moment they indicated they started the work until they marked it complete. This calculation may not correctly represent actual work time if the crew mistakenly started or completed the task too early or too late.

An accurate report of the actual time the crew spent working on site during each visit is key to an accurate calculation of the remaining time left to complete the entire complex activity work. For this reason, complex activities always allow the crew to review and adjust the time they spent traveling to and being on site for each visit.

Typically, crews stay on site for prolonged periods of time while they perform complex activity work. This means their breaks would most likely be taken while they are on site and therefore complex activity work allow breaks to be taken while on site.

Refer to [Crew Time](#) for more information.

Completion Information

The host system is updated upon completion of each crew's visit task assignment as well as when the entire complex activity is completed or canceled.

NOTE: Logic to compose the completion message for an activity resides in a "Prepare Completion Data" algorithm on the activity's business object. Click [here](#) to see the algorithm types available for this system event. You can turn off sending interim visit completion messages using an algorithm parameter.

About Activity Dependency

One activity may be dependent on another activity in various ways. A group of activities that have dependencies is referred to as "bound work".

This section describes the various types of bound work and the dependencies created when activities are tied to each other.

Job

A job is a group of activities that must be performed by a single crew during a single shift. For example, a job may consist of an activity to pick up goods from one location and deliver them to another location.

The following sections highlight concepts related to handling jobs.

Activities Are Grouped By A Job Identifier

A job is not a standalone entity but rather a set of activities referencing the same [job identifier](#).

Typically all activities of a job are interfaced together in a single service call but activities may be added individually as needed, as long as it references the same host job identifier. If a new activity is sent in later referencing an existing host job identifier, it automatically becomes associated with the corresponding internal job identifier.

Order

The order in which the activities in a job should be performed is established as usual by their time windows. A scheduling order number may optionally be specified to achieve a more explicit order. Notice that ordering should not contradict the time windows ordering.

NOTE: Only the host system can send in a set of activities linked as a job. The base product does not provide the means to compose a new job nor manipulate the set of activities included in an existing job within the application itself.

Loaded Goods Must Be Unloaded

The shift has to start and end with no load on board. A job does not necessarily have to involve transfer of goods. But if it does then all goods picked up by one activity in the job must be dropped off by another activity in the job.

Refer to [About Transfer of Goods](#) for more information.

All Scheduled or None

If there is any scheduling issue for one activity in a job, all of the activities are disabled from scheduling until the issue is resolved. Once the issue is resolved, all of the activities are rescheduled automatically by the scheduler.

Some Actions Affect Entire Job

Actions performed on an activity that is part of a job that cause the activity to be rescheduled from its currently assigned shift must affect the entire job so that all activities in the job are rescheduled together.

If such action is not performed on the entire job, the scheduler may disable the activities until they can be scheduled together again.

Therefore, performing any of the following actions on one activity in a job automatically performs the action to all other activities in the job.

- Allocation to a shift as well as removing such restriction
- Fix to crew as well as removing such restriction
- Unassigning an activity from its current shift
- Deferring the schedule of the activity to a later date
- Locking an activity to a depot run
- Excluding the contractor company currently assigned to perform the work, as well as removing such restriction
- Revert the schedule of the activity to company crews only

The CDI Gantt provides a visual cue to indicate that an activity is part of a job and therefore has dependencies. This helps the dispatcher to easily see how each activity is related and what would be the impact of certain actions, if taken.

Appointment Booking

Appointment booking and rebooking requests must include all activities in the job as the scheduler has to find availability for all of them on a shift.

Partial Completion

As explained earlier, all activities of a job must be scheduled to the same shift otherwise they are disabled by the scheduler.

There are times when the crew completes some activities of a job but does not manage to complete all of them for various reasons. To allow the scheduler to properly reschedule the remaining activities, the system automatically removes the completed activities from the job at end of shift. Those completed activities still reference their original job identifier for reporting purposes.

Viewing Activities In A Job

If an activity belongs to a job, the **Job Activities** zone on the activity maintenance portal lists all activities in the job in the order they should be performed, highlighting the current activity in that list.

In a partially completed job, only remaining non-worked activities remain in the job so that they can be rescheduled. The Job Activities zone displays all of the activities that were part of the original job.

Chain

A chain represents linked activities where each activity is dependent upon the start or completion of another activity, with dependency defined in terms of timing and assigned resource. The entire set of dependent activities represents a project like work request that is also sometimes referred to as long cycle work.

The following sections highlight concepts related to handling chains.

A Chain Dependency Can Span Multiple Shifts

Chains differ from jobs in that the activities are not bound to a specific shift, they are bound in time and can span multiple shifts. They are set up so that each activity depends upon the start or completion of the prior activity before it can start.

Dependency Rules

A dependent activity can start within an offset range relative to the start or finish time of the parent activity. An activity can also be restricted so that it can only be worked by a crew different than its immediate parent activity or by a crew other than any of those assigned to activities in the entire chain.

For example, if there is a project to dig a trench, lay pipes, and fill the trench, a chain dependency of tasks might be set up as follows:

- A surveyor is scheduled to lay out the plan before the start of work.
- Only after the surveyor is finished can the digging begin.
- The pipes cannot be laid out until the trench is complete.
- The trench cannot be filled until the pipe is laid.
- and so on

The following considerations apply for chains:

- An activity can only be immediately dependent on one other activity which could also be dependent on another activity and so on.
- A circular dependency is not supported, i.e. the activity cannot be dependent on itself even indirectly . (Example: A->B->C->A)
- Appointment booking, conditional booking, and chooser requests can only be used for an activity in a chain if the activity does not have any other activities dependent on it.
- The time windows for a dependent activity must be wide enough to support its timing dependency constraints, including offsets, with its parent activity. This has to take into account potential early or later start of the parent activity. For example, assume activity B is scheduled to start an hour after activity A starts. Since activity A may start earlier than planned by more than one hour, activity B should be defined with time windows wide enough to accommodate a

reasonable early start of activity A. In other words that applying an offset towards the actual start of activity A would still be within the time window for activity B.

- If the scheduler finds conflicts because these considerations have not been followed, the whole chain (non-started part of it) is disabled and not scheduled.

Assist Activities

There are times where a crew arrives on site and realizes they need assistance from another crew to help with the work of an activity. For example, a crew responds to an outage activity but upon arrival, realizes there is a tree that blocks access to the work location. They contact the dispatcher that organizes an assisting crew to help with the removal of the obstacle.

The assist activity needs to be scheduled at the same time as the original activity so that both crews are on site at the same time. In addition it must be scheduled to a crew other than the original activity's crew. The original and assist activities are an example of a small chain dependency.

The scheduler attempts to assign the assist activity to a suitable crew. However, the dispatcher can also manually allocate it to any specific crew as needed. To do that the dispatcher could take advantage of the shift chooser query option which consults the scheduler for a cost based recommended list of shifts to choose from.

Only specific activity types may be selected to create assist activities.

Refer to [Assist Activity Types](#) for more information.

Host Initiated Only

Only the host system can send in a set of activities linked as a chain. With the exception of [assist activities](#), the base product does not provide the means to compose a new chain, nor manipulate the set of activities included in an existing chain within the application itself.

Complex Activity Chains

Complex Activities can be part of a chain of activities, and can precede or follow regular or complex activities. However, any restriction dependency of a complex activity towards another activity and vice versa are not applicable because complex activities may be worked by more than one resource over time.

Viewing Activities In A Chain

The CDI provides a visual cue to indicate that an activity is part of a chain and therefore has dependencies.

If an activity is part of a chain, you can navigate from the activity portal, as well as from the CDI portal, to the [Activity Dependency](#) portal for an overview of the scheduling state of entire chain.

About Assignments

An assignment is a copy of an activity (or a complex activity's visit task) created by the system to track the assigned resource's (crew or a contractor company) progress in accomplishing the work.

Refer to [About Activities](#) and [About Complex Activities](#) for more information.

When the activity is ready to be worked by the assigned resource, the system creates their assignment and sets the activity to the **Assignment In Progress** state. The activity remains in this state until the assigned resource either completes their assignment or declines it.

Refer to the [Activity](#) and [Visit Task](#) workflows to better understand when an assignment is created and how it affects the activity when finalized.

Once the crew acknowledges their assignment, they may provide status updates informing the system of their progress. For example, when they start traveling to the location, when they have arrived there, when they start to perform the work as well as to provide all the required information when completed it.

When the activity is assigned to a [contracted capacity](#), work is outsourced to the contractor company and the system no longer has visibility into when and by whom the work is actually performed, nor does it know how many attempts the contractor company made to complete the work. Your company's expectation is to receive the completion information when the work is completed, otherwise a decline update if the work cannot be completed in a timely manner so it can be rescheduled.

This section describes concepts related to managing assignments.

Current Assignment

At any point in time an activity may only be scheduled to a single crew's shift (or [contracted capacity](#)) and if dispatched, there should only be one active assignment for it. The current assignment for an activity is the assignment created for its currently assigned resource.

If the crew does not get to complete an activity assigned to them, their assignment for that activity is [returned](#) at end of their shift and the activity may be rescheduled to the same or other crew on a later day, repeatedly until the work is complete. In a similar way, if the contractor company determines they cannot perform the work in a timely manner, they return their assignment for the activity allowing it to be rescheduled as needed.

Over time there could be multiple assignments for the same activity, each representing an attempt to work it.

Refer to [Concurrent Assignments](#) to learn about situations where multiple active assignments may exist for the same activity.

Assignment Workflow

An assignment is created when the activity lifecycle reaches a point in which work should be dispatched to the assigned resource. The lifecycle of an assignment affects the lifecycle of its activity.

The assignment workflow differs when it tracks the progress of a crew working an activity as opposed to a contractor company's agreement to perform an activity.

The following sections describe key stages in each workflow.

When Assigned To A Crew

When the crew is ready to perform work and has started their shift, the system creates an assignment for each of their assigned activities to record their progress working that activity. If the crew uses the mobile application to manage their work, a copy of their assignment is sent to their mobile device.

Assignment states are depicted in the following diagram:

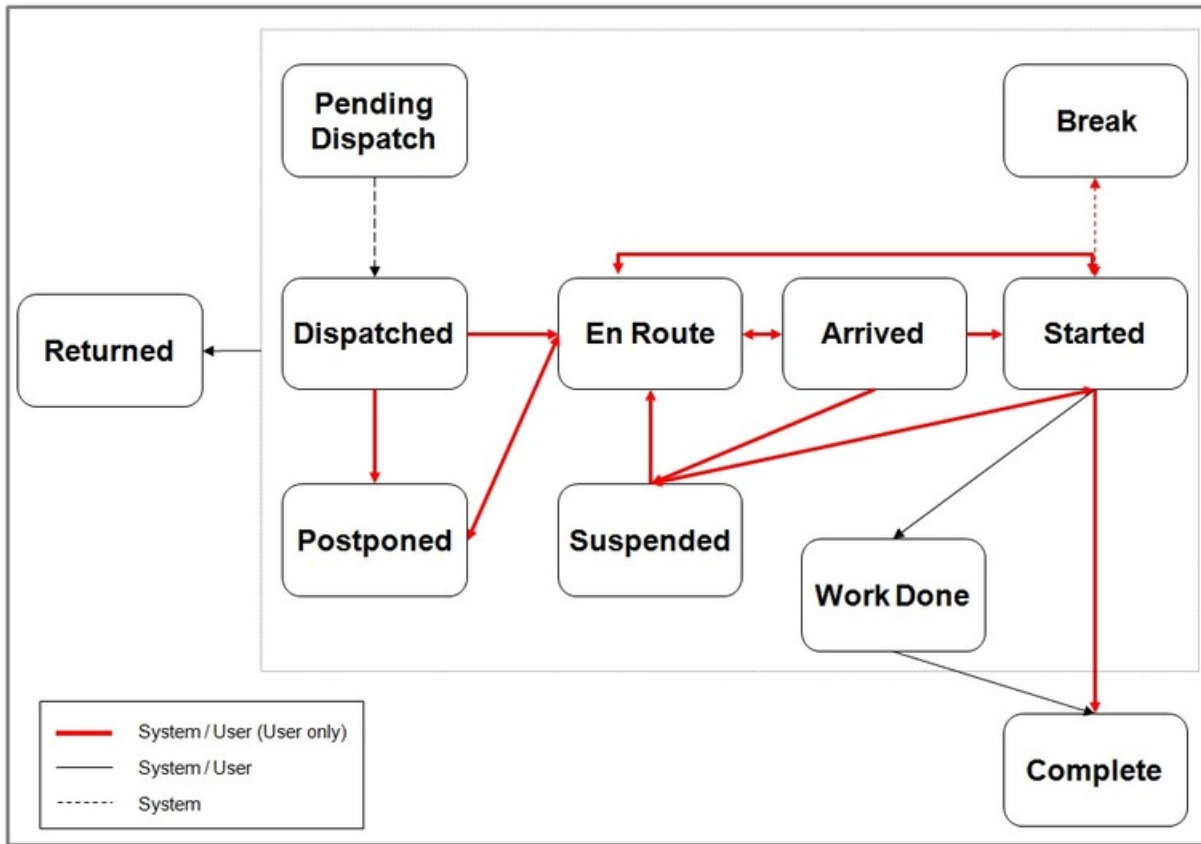


Figure 4: Crew Assignment Lifecycle

The following sections describe the crew's assignment overall lifecycle from the moment it is dispatched to the crew until the crew either completes the work or returns it back to be rescheduled.

Refer to [Crew Assignment](#) for more information.

Pending Dispatch

An assignment is initially created in this state waiting to be acknowledged as [dispatched](#).

If the crew uses a mobile device, a copy of the assignment is sent to the crew's device. If the crew uses SMS to work their tasks then a message with a brief description of the work is sent to their mobile phone.

Dispatched

An assignment is considered dispatched if the assigned crew acknowledges its receipt. This is the last step in the process of dispatching an activity. Refer to [The Dispatch Process](#) for more information

The mobile application acknowledges receipt of the assignment by sending a message to the server to update the assignment to the dispatched state. Refer to [Automatic vs. Manual Acknowledgment](#) for more information.

If the crew does not use a mobile device the assignment is automatically marked dispatched.

En route

This state indicates that the crew is currently on its way to the activity. The crew provides this [status update](#) when they are about to start traveling to the activity location. If the crew was previously out of service they are now considered back in service.

Time spent while in this state is considered travel time for [crew time](#) reporting.

Arrived

Typically the crew may start work immediately upon arrival to the activity location. This state indicates that the crew has arrived at the location but has not yet started work for some reason. For example, they may need to wait for the superintendent to provide access to the work site. The crew provides this [status update](#) when they arrive on site but cannot start the work right away.

Time spent while in this state is considered wait time for [crew time](#) reporting.

Started

This state indicates that the crew is on site and has started to perform the work requested by the activity. The crew provides this [status update](#) when they start the work.

Using the mobile application, the crew may also provide interim completion details as they work on the activity and send them right away to the server, even before fully completing the work. They may also report their estimation of how much time is still needed to complete the work when the activity takes much more or much less time than planned.

Time spent while in this state is considered work time for [crew time](#) reporting.

Break

This state indicates that the crew is taking a planned break while working on site. The system automatically sets the assignment to this state when the crew indicates they start their planned break. When the crew completes their break, the system automatically switches the assignment status back to started.

Refer to [Breaks](#) for more information.

Postponed

This state indicates the crew decided to postpone working on this activity any time prior to arriving on site, with the intention to come back to it later. Once the crew arrives on site the activity can no longer be postponed, but it can be [suspended](#) instead.

For example, the crew may have been on their way to the activity location and were instructed by a dispatcher to postpone this activity and handle an emergency situation or assist another crew nearby.

The system prompts for a reason for the postponement and the time to which the activity should be postponed, which should be within the current crew shift. A postponed activity remains dispatched to the shift but the scheduler re-optimizes its position within the route based on the estimated postponed time.

The crew provides this [status update](#) in that situation.

Suspended

This state indicates that the crew arrived on site and started work, but was unable to complete it for some reason. For example, the crew may be instructed by a dispatcher to handle an emergency situation or assist another crew nearby, etc.

The activity remains assigned to the crew, allowing them to return to it later on. At this time the system provides the crew with the option to also keep the activity with that crew for a future shift, if they are not able to complete it during the current shift.

The system prompts for a reason for the suspension, the time to which the activity should be postponed, which should be within the current crew's shift, as well as how much time is still required to complete the work once it is resumed. A suspended activity remains dispatched to the shift but the scheduler re-optimizes its position within the route based on the estimated postponed time.

The crew provide this [status update](#) in that situation.

Completed

This state indicates that the crew has completed work and all completion details have been entered.

The crew may complete their assignment but provide a reason code that marks it incomplete when actual work is impossible to perform. For example when details on the activity are invalid or there was some problem denying the crew access to the work site.

Crews that use the mobile application are required to provide the necessary completion information when they complete their assignment. If they are connected or become connected later on, the mobile application forwards the entire completion information to the server.

If the crew uses SMS to work their schedule, they cannot provide the necessary completion information using text messages and therefore cannot complete their assignment. They can only mark their assignment as [work done](#). Later on when they are back at the service center, they may enter the completion information they have recorded on paper and complete their assignment.

In the same way, if the crew uses no device (non-MDT), they typically contact a dispatcher to report their [work done](#) and complete their assignment later when they are back at the service center.

Completing the assignment causes the activity to reevaluate its state.

Refer to the [Activity](#) and [Visit Task](#) workflows to better understand how finalizing an assignment reassesses the activity state.

Work Done

An assignment in this state indicates the work is performed but the crew did not yet enter the necessary completion information. This is typically used by crews that uses SMS text messages to report status updates as well as crews with no device (non-MDT). Later on at the office, the crew enters the completion details and completes the activity.

Returned

An assignment in this state indicates that the crew is no longer responsible to perform the activity and returns it back to be rescheduled if possible.

An activity assignment may be returned in the following situations

- The crew did not manage to work the activity before the end of their shift.
- The crew explicitly declined taking on the activity for various reasons. The crew has to specify the reason for declining the activity. If the reason is due to lack of worker or vehicle capability, a To Do Entry is created for a resource manager to follow up and ensure the capability information recorded for the crew's mobile workers and vehicles is accurate.
- The system automatically recalled the activity due to a host request to cancel the activity. Refer to [Activity Cancellation By Host](#) for more information about how such request is handled
- The dispatcher performed a schedule change action that [forced a recall](#) of the activity from the crew.

Returning the assignment causes the activity to reevaluate its state.

If the shift is not completed, the shift is added to the activity's list of [excluded shifts](#) to prevent scheduling the activity back to the same shift.

Refer to the [Activity](#) and [Visit Task](#) workflows to better understand how finalizing an assignment reassesses the activity state.

When Assigned To A Contractor Company

When an activity is committed to a capacity based contractor, the contractor company receives a copy of the activity information in the form of an assignment that they use to report their progress working that activity.

The following diagram depicts the lifecycle of a capacity contractor assignment.

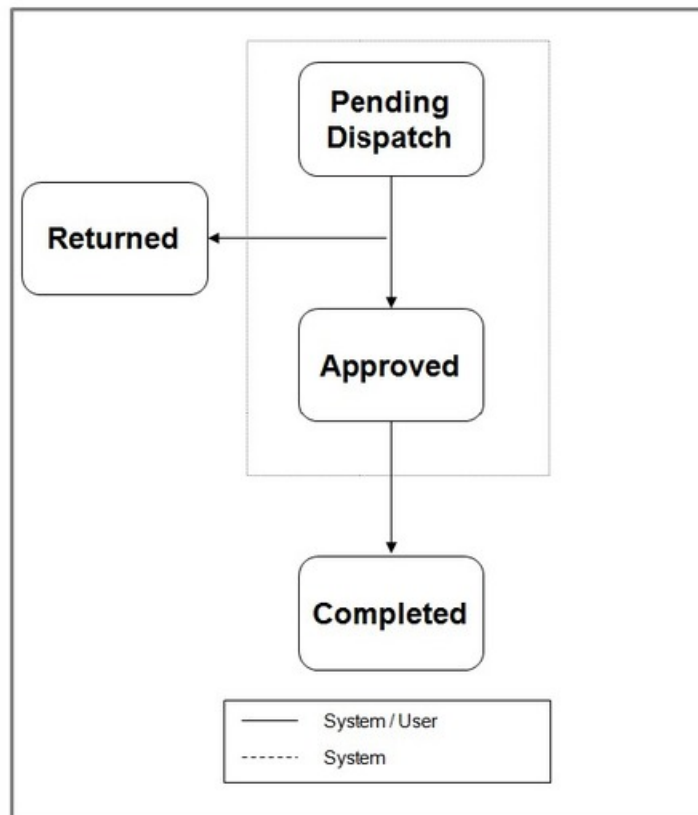


Figure 5: Contractor Assignment Lifecycle

The following sections describes the contractor company's assignment overall lifecycle from the moment it is dispatched to them until they complete the work or decline it.

Refer to [Managing Committed Work](#) for more information.

Pending Dispatch

An assignment is initially created in this state waiting to be [approved](#) or [declined](#) by the contractor company.

If the contractor company fails to approve taking on this activity within the agreed upon approval time, the system automatically declines their assignment and sets it to the Returned state.

Approved

This state indicates that the contractor company has agreed to take on this activity. From this moment on, the system has no visibility as to how the contractor company carries out the work and when.

Completed

This state indicates that the contractor has completed work and all completion details have been entered.

Completing the assignment causes the activity to reevaluate its state.

Refer to the [Activity](#) and [Visit Task](#) workflows to better understand how finalizing an assignment reassesses the activity state.

Returned

An assignment in this state indicates that the contractor company is no longer responsible for performing the activity and returns it back to be rescheduled.

An activity assignment may be returned in the following situations

- The contractor company has either declined taking on the activity committed to them or failed to approve taking the work in a timely manner. The contractor company is **excluded** from further working the activity.
- The dispatcher performed a schedule change action that **forced a recall** of the activity from the contractor company.

Returning the assignment causes the activity to reevaluate its state.

Refer to the [Activity](#) and [Visit Task](#) workflows to better understand how finalizing an assignment reassesses the activity state.

Crew Assignment

An assignment is created for each activity dispatched to a started shift. The following sections describe key aspects related to a crew's activity assignment.

Automatic vs. Manual Acknowledgment

Typically, it is enough for the crew's copy of the assignment record to reach the mobile device for the mobile application to automatically acknowledge the dispatch. The mobile application immediately sends a status update back to the server to mark the server copy of the assignment record as dispatched.

For certain activities though, typically though not limited to emergencies, mere receipt of the assignment on the device is not enough. The crew has to manually acknowledge taking on the work. If they cannot take on the work, it is important to know that as soon as possible so that the work is rescheduled quickly to another crew. Any activity, not just emergencies, may request manual acknowledgement. This may be configured on the activity type as well as requested on an individual activity.

When the activity may explicitly states that a manual acknowledgement is required, the mobile application prompt the crew to accept or decline the new assignment sent to them. If the crew accepts the new assignment the mobile application sends a message to mark the server copy of the assignment as dispatched. If the crew declines taking on the assignment the assignment on the server is declined which causes the activity to be rescheduled, excluding this crew.

If the assignment is not acknowledged in a timely manner, either automatically or manually, a synchronization alert may be raised for a dispatcher to follow up with the crew. Refer to [Synchronization Alert](#) for more information.

Synchronization Alert

When dispatching an assignment to a mobile device fails to complete in a timely manner (typically due to poor connectivity), the system creates a synchronization alert for a dispatcher to follow up with the crew.

NOTE: Initiating a synchronization alert is optional. An alert is only created if an active alert type referenced as the **Synchronization Alert Type**[global configuration](#) parameter.

Refer to [About Alerts](#) for more information.

A dispatcher assigned to handle such an alert may contact the crew and verbally communicate to them the key details of the activity. The crew can then create a local copy of the activity on their mobile device, capturing the key details provided and use it instead of the original assignment to record status updates and completion details.

This local copy of the activity has to be properly associated with the actual activity when the former is sent to the server. The activity identifier is too long to communicate by voice to the crew when they create their local copy so the system provides a shorter system generated alternate identifier to be used instead. The alternate identifier is captured on the original activity as well as provided to the crew as part of the details they capture on their local copy. The local copy of the activity is also known as a field-referenced activity (FRA) for that reason.

NOTE: Logic to handle the business rules associated with setting an alternate id for an activity resides in a "Set Alternate ID" algorithm on the activity's business object. Click [here](#) to see the algorithm types available for this system event.

Handling an assignment synchronization situation works as follows

- The dispatcher receives an alert when the synchronization of an assignment to the crew's mobile device takes too long or fails.
- The dispatcher clicks the **Voice** button on the Alert Queue. This action marks the assignment that failed to be sent as dispatched and cancels any further attempts to send it to the crew, so the assignment won't exist twice on the device once the FRA is created. Then, the system automatically navigates to the Activity portal and displays a message instructing the dispatcher to communicate the activity details to the crew. This message also includes the short alternate identifier of the activity.
- The dispatcher communicates the information to the crew (typically via phone), and the crew creates an FRA on the mobile device referencing the activity alternate identifier.
- The crew works the activity, updating its status as work progresses through completion, just like any other assignment. When the crew's connectivity is reestablished, the system automatically matches the FRA to the corresponding activity using the alternate identifier.

Keeping Mobile Copy Up To Date

If the crew is using the mobile application to manage their work they receive a copy of each of their dispatched assignments (and other tasks) sent to their mobile device from the moment they start their shift.

When key activity information changes after the assignment is dispatched, the system automatically synchronizes the mobile copy of the assignment and sends the crew a new copy of it with the latest activity information.

NOTE: Logic to initially send a copy of the assignment to the crew's mobile device as well as synchronize the mobile copy to keep it up to date when changes are made on the server resides in a "Synchronize Task" algorithm on the assignment's business object. Click [here](#) to see the algorithm types available for this system event.

Crew Status Updates

The mobile application allows a crew to advance the state of their assignment to reflect their current progress working the activity. They may do so even when they are not connected. The actual time of events is captured locally on the device and is forwarded to the server later on when they become connected again.

If the crew uses SMS to work their schedule, they send in a text message to update the state of their assignment on the server.

A crew that uses no device (non-MDT) typically calls in to report their status and a dispatcher makes the update to the assignment on their behalf.

Refer to [Working The Schedule](#) for more information.

Keep With Crew

Crews may keep certain activities with them for future shifts. Any work the crew did not manage to complete by the end of their shift may be rescheduled to a future shift for this or any other crew. A crew may decide to keep a certain activity they did not manage to complete by the end of their shift so that they can work on it the next day or during a future shift.

For example, a crew had to suspend working on an activity to respond to an emergency activity and never got the time to go back and complete the suspended activity. If the crew has keys or parts needed to perform the suspended activity, they may decide to retain the activity so that they can work on it the next day or any of their future shifts and not have it rescheduled to another crew.

The crew can make this request at the time they postpone, suspend, or decline an activity assigned to them. When the crew ends their shift, they also receive a reminder to review and mark activities they want to keep with them.

Refer to [Fix To Crew](#) for more information.

Crew Time

When the crew completes an activity, the system can be configured to present the crew with a summary of the time they spent on behalf of this activity in the following categories:

- **Travel.** This time is calculated from the moment the crew indicates they are en route to the activity location and until they indicate they are no longer traveling.
- **Wait.** This time is calculated from the moment the crew indicates they have arrived on site and until they start the work, capturing the amount of time the crew had to wait before they could start the work.
- **Work.** This time is calculated from the moment the crew start to work and until they indicate they are no longer working- either they complete the activity or suspend it for some reason.
- **Break.** This time is calculated for each break that the crew start and complete while being onsite. Refer to [Taking a Break While On Site](#) for more information.

If the crew has attempted to travel to the activity, then postponed it in order to work on another activity, and then resumed back to travel again to the previous activity, the time spent on these categories from all attempts are added up.

The crew may adjust the time information as needed to correct situations when they have provided status updates earlier or later than when they actually occurred. Both the calculated times and the adjusted times are sent back to the host system as part of the activity completion information.

NOTE: Crew time review is incorporated as part of the activity completion data entry, if enabled on the activity type, as it may be applicable to certain types of work only.

Recall

Recalling an activity refers to a system initiated request to finalize the [current assignment](#) of an activity for the currently assigned resource. When the recall is successful, the resource is no longer responsible for performing the activity.

The recall request may or may not succeed in recalling the activity. When it is successful, the assignment is set to the **Returned** state with a reason code stating it was returned due to a recall.

An assignment may be recalled when

- A dispatcher attempts to take certain actions on an activity that is already dispatched. Refer to [Forced Recall](#) for more information.
- The host system requests to cancel the activity. Refer to [Activity Cancellation By Host](#) for more information about how such request is handled.
- If allowed by shift configuration, the scheduler may make improvements to a crew's dispatched route that result in some activities being recalled and replaced with others. Refer to [Crew Undispatch and Sequence Lock](#) for more information

For crews that utilize the mobile application, the system attempts to recall the activity by sending a recall request to the crew's mobile device. When the recall message is received on the device, the mobile application determines whether the assignment is in a state that allows it to be recalled or not. If recall is possible, the assignment is automatically returned from the mobile device and removed from the crew's task list, notifying the crew of the recall. The recall is complete when the assignment status update reaches the server.

If the crew is only equipped with a SMS device, the system sends the crew an SMS message notifying them of the recall. The crew can either ignore the request if it's too late to recall or confirm the recall by sending a recall confirmation message back to the server. The recall confirmation message completes the recall by causing their assignment on the server to be returned.

The assignment remains with the assigned resource and the activity retains its schedule if the recall is not successful for any reason. If this was a forced recall, i.e. initiated by user in order to force a schedule change for a dispatched activity, then concurrent assignments may now exist for the activity.

Refer to [Concurrent Assignments](#) for more information.

For crews not using any device (non-MDT), there is no automated way of requesting and confirming a recall. The only recall allowed from these crews is that forced by a dispatcher action. In this situation the system assumes that the dispatcher has contacted the crew and confirmed the recall verbally before making the change and therefore their assignment is automatically recalled.

NOTE: The system is configured to consider all assignment statuses non recallable when the crew is already at the location.

Concurrent Assignments

The system would only schedule one crew (or contractor company) to work an activity at any point in time. As such, there should be only one active assignment for the activity, the one created for the [currently](#) assigned resource.

If that resource could not perform the work, their assignment is finalized indicating they have returned the activity to be rescheduled. When the activity is rescheduled to another resource, a new activity assignment is created for that resource which is now the currently active assignment for the activity.

However, due to a dispatcher initiated recall, it is possible for multiple active assignments for the same activity to exist concurrently. These are referred to as concurrent assignments.

Refer to [Forced Recall](#) for more information.

Typically the dispatcher coordinates the recall with the crews involved such that only one of them actually performs the work and completes their assignment where the others just return their assignment.

However, if there is some confusion in the field and more than one crew actually completed their assignment, the system cannot determine from which completed assignment to take the completion information to send back to the host. For example, a crew might arrive at the location, see that the work was already done, and then complete their assignment with no work details.

In this situation the system sets the activity to the [Collision Detected](#) state and raises a To Do Entry for a user to review the completed assignments and choose one assignment to complete the activity by clicking the "Complete Activity" action on that assignment.

About Activity Attachments

An attachment can be any file, such as a photo, a document with instructions or specifications, a spreadsheet, or any other supporting documentation. Several standard attachment file types are supported such as PDF, Word, Excel, or other document types.

Additional information about an activity, either sent from the host, added by the dispatcher, or captured by the crew, may be associated with an activity in the form of an attachment. This section describes functionality associated with activity related attachments.

Attachments Related To The Type Of Work

Attachments might be linked to specific activity types for the purpose of providing supporting information and guidelines to crews when performing this type of work. For example, an attachment may include instructions on how to perform the work or safety guidelines the crew should adhere to when performing such work.

Some attachments may be referenced on multiple activity types as applicable and are referred to as common or shared attachments.

Links to these attachments are available to the crew on their mobile device when they work activities of this type so they can download their content on demand, provided they are connected (to save bandwidth, time and storage space, etc.).

Common attachments the organization prefers to always be available on the device can be explicitly added to the mobile application deployment. This way the crew does not need to manually download them each time. Refer to [Managing Attachments](#) for more information.

These attachments may not be modified by the crew.

Attachments Related To A Specific Activity

Some attachments contain content related to a specific activity. The attachment may be provided by the host system along with the activity, captured by the crew in the field, or attached by a dispatcher to the activity.

This section describes functionality associated with activity specific attachments.

Attachments Sent In From The Host System

The host system may send supporting information in the form of attachments along with the activity.

For crews using the mobile application, the system includes links to these attachments when dispatching the activity to the crew's mobile device. The crew can download their content on demand provided they are connected (to save bandwidth, time and storage space, etc.).

These attachments may not be modified by the crew.

Capturing Content In The Field

Crews using the mobile application may capture content or attach documents related to a specific activity in the field. These attachments are later on synchronized and uploaded to the server.

NOTE: The types of attachments that can be captured depends on the mobile application and the mobile device used.

Sending Attachments To The Host System

By default, any attachments created in the field are always sent back to the host system as part of the activity completion information. Any other attachments added to the activity within the system must be marked explicitly in order to be sent to the host. Attachments sent in from the host system are never sent back.

For example, the host sends in an activity to investigate a possible electricity diversion by a customer. During the investigation, the mobile worker confirms the electricity diversion and takes a picture of the evidence. That picture is returned to the host system in the form of an attachment along with the investigation activity completion information. On a subsequent disconnect service activity resulting from the investigation, the host sends in the picture taken during the investigation to aid the mobile worker in the disconnection activity. The picture would not be returned to the host system again.

Dispatchers May Add Attachments

A dispatcher may attach files to a specific activity as needed. If the activity has already been dispatched, then for crews utilizing the mobile application the system sends the link to that attachment to their mobile device right away.

At the time of adding an attachment to an activity, the user can indicate whether it should be sent to the crew or not. For example, an attachment may not be required for work or of any interest to the crew in the field but rather added for the purpose of sending it to the host system when the activity is completed.

Maintaining Activities

This portal is used to display and maintain an activity.

Refer to [About Activities](#) for more information.

You can access the portal from the **Main Menu > Service Management > Activity**. You are brought to a query portal with options for searching for a specific activity. Once an activity has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page

- **Activity.** This zone provides information about the activity, including the various maintenance actions applicable to its current state.
- **Activity Time Windows.** This zone lists the various time windows available for the activity.
- **Activity Assignment Completion Details.** This zone displays the completion information provided by the assigned resource when they completed the work. The zone is only visible when the activity is completed.
- **Assignment History.** This zone lists all resource assignments for the activity over time.
- **Attachments.** This zone lists all attachments specific for the activity including those captured by crews in the field.
- **Job Activities.** This zone lists all activities that belong to the same job as the current activity. The zone is only visible if the activity is part of a job.
- **Activities Created By Crew.** This zone list activities created by the crew that are related to this activity.
- **Procedures.** This zone lists all procedures completed by the crew for this activity. The zone is visible only if such procedures exist for the activity.

The **Activity Schedule** zone on the portal's **Schedule** tab page displays the current scheduling information for the activity. Refer to [Scheduling An Activity](#) for more information.

Request An Assisting Crew

Use the **Assist** action to create a new activity for the assisting crew of a specific activity type. The system creates the assist activity and navigates to the Activity portal so that you can add a note to the crew if needed. You can either allow the scheduler to assign it to an available crew or you can manually allocate the activity to a specific crew. If the original activity has already started, which is typical in assist situations, the scheduler attempts to dispatch it right away. You can also manually dispatch the activity as needed.

The current activity and the assisting crew's activity are bound as a chain so that the scheduler plans their arrival accordingly.

Refer to [Assist Activities](#) for more information on activity dependency.

Maintaining Current Crew Assignment

Use the current assignment link on the activity display zone to navigate to the activity's current assignment to a resource. This may be either an assignment for a crew on a specific shift or an external contractor assignment to perform the work as part of an agreed upon work capacity.

Refer to [About Assignments](#) for more information.

Crews that do not use any mobile device (non-MDT) to perform their work may rely on a dispatcher to act on their behalf and update the state of their assignment to reflect their progress. For example, if the crew cannot perform an activity

and needs to decline it they may call in to report the situation. You can navigate to the current assignment and decline it providing the appropriate reason etc. This allows the activity to be rescheduled to another resource if possible.

However, crews using the mobile application maintain their copy of the assignment locally on the mobile device itself. When they make updates to their local copy the mobile application forwards the update the server when they become connected.

NOTE: It is important to let such crews fully control the state of their assignment and avoid manually updates to their assignment on the server. Manual updates may contradict status updates sent in from the crew and lead to unpredictable results. It is strongly recommended to confirm and coordinate such updates with the crew.

Completing Activities with Concurrent Assignments

If multiple completed assignments are received for the same activity, the system cannot automatically complete the associated activity. To resolve the situation, review each of the completed assignments for the activity and navigate to the assignment you want to use to complete this activity. Use the **Complete Activity** button to complete the activity with the completion information provided by this assignment.

Refer to [Concurrent Assignments](#) for more information.

Maintaining Complex Activities

This portal is used to display and maintain a complex activity.

Refer to [About Complex Activities](#) for more information.

You can access the portal from the **Main Menu > Service Management > Activity**. You are brought to a query portal with options for searching for a specific activity. Once an activity has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page

- **Activity.** This zone provides information about the activity, including the various maintenance actions applicable to its current state.
- **Activity Time Windows.** This zone lists the various time windows available for the activity.
- **Assignment History.** This zone lists all crew assignments for the activity from all visit tasks over time.
- **Attachments.** This zone lists all attachments specific for the activity including those captured by crews in the field from all visit tasks.
- **Activities Created By Crew.** This zone list activities created by crews at the activity location that are related to this activity.

The following zones may appear as part of the portal's **Schedule** tab page

- **Activity Schedule.** This zone displays the current scheduling information for the activity. Refer to [Scheduling An Activity](#) for more information.
- **Complex Activity Gantt.** This zone displays a graphical representation of the crews assigned to work the complex activity. The Gantt works in the same way as the dispatcher's scheduling Gantt on the CDI portal with the exception that filters and the ability to launch the geographic map are not applicable from this zone. Refer to [Using the Scheduling Gantt](#) for more information.
- **Crew Visits.** This zone lists visit tasks created for the activity.

Maintaining Assignments

This portal is used to display and maintain an activity assignment to a resource.

Refer to [About Assignments](#) for more information.

You can access an assignment from the [activity maintenance portal](#). You may navigate directly to the activity's [current assignment](#) or to any assignment listed in the "Assignments History" zone.

The **Assignment** zone on the portal's **Main** tab page provides information about the assignment, including the various maintenance actions applicable to its current state.

Activity Dependency

The portal provides a schematic and a Gantt view of the chain dependency to which a specific activity belongs to. If an activity belongs to a chain dependency you can also navigate to this portal from the Dependency section on the activity details zone of the [activity maintenance portal](#).

Refer to [Chains](#) for more information.

The following zones appear as part of the portal's **Main** tab page

- **Dependency Gantt.** This zone displays a graphical representation of the crews assigned to work activities that belong to the current activity's chain dependency. The Gantt works in the same way as the dispatcher's scheduling Gantt on the CDI portal with the exception that filters and the ability to launch the geographic map are not applicable from this zone. Refer to [Using the Scheduling Gantt](#) for more information.
- **Dependency Hierarchy.** This zone shows a schematic view of the current activity's entire chain of dependencies.

Override Time Windows

About Override Time Windows

There are times when the hours of operation for certain types of work in all or some service areas may need to deviate from normal typically due to unplanned business events. Work that is normally allowed may become not allowed and vice versa, work that is normally not allowed may become allowed during that event.

This section describes concepts related to managing activity override time windows.

Prohibit Work When Otherwise Allowed

An override time window may be used to prohibit the schedule of certain type of work for the duration of an unplanned business event. For example, prohibit disconnects or deliveries from being performed during periods of extreme weather conditions.

Allow Work When Otherwise Not Allowed

Certain types of work may be not be scheduled on holidays specified on the [work calendar](#) associated with the activity type. They may also be restricted to business hours defined on the [work profile](#) associated with the activity type.

An override time window may allow such work to be performed if needed during those holidays or outside the specified business hours. For example, allow work on a holiday to manage a backlog of certain type of work.

Overrides Regenerate Arrival Time Windows

An override time window defines the period of time, the type of work and areas impacted as well as whether work is allowed or not allowed during that time. The override impacts activities for which at least one of their effective time windows (ETW) overlaps with the duration of the event.

When an activity is added or when its time window information changes, the system automatically calculates its arrival time windows(ATW) taking into account any overrides.

NOTE: Due to the potentially high number of existing activities that may be impacted by an override, it cannot be applied to them right away. The impact of introducing a new override or canceling an already applied override onto existing activities is delayed to the next time they are monitored by the Task monitor batch process. When next processed, the activity recalculates its arrival time windows(ATW) if it detects an applicable override or an override cancellation.

Refer to [When Can The Activity Be Actually Scheduled?](#) for more information about the impact of an override on an activity's arrival time windows(ATW).

Enforcing Overrides

The scheduler may schedule an activity to be started or completed after the end of an arrival time window (ATW) if this results in an overall cost benefit. In other words, lateness is cost driven and not a hard limit.

An override event to prevent certain work from being performed has a legal binding and as such, the system has to ensure work is not scheduled during that time at all costs. While the arrival time windows (ATW) of impacted activities are regenerated to exclude override periods, the scheduler also ensures that activities are not scheduled late, outside their time windows, and into the override event period.

NOTE: The scheduler honors such override as long as the activity has not been dispatched yet.

Once dispatched, activities that overlap with such override require manual intervention. The system provides a [KPI](#) to report these violations, allowing the dispatcher to contact the crew and ensure work is recalled.

Use Case Examples

For this example assume repair work is associated with a work profile that allows work to be scheduled from 8 AM to 5 PM Monday through Friday and is not allowed to be performed on the New Year's Day holiday.

Consider the following scenario:

- A repair activity is received from the host with an effective time window that starts on Monday 7 AM until Saturday 2 PM the week of the New Year's Day holiday.
- The activity is created with the following arrival time window (ATW) that week
- Monday 8 AM - 5 PM
- Tuesday 8 AM - 5 PM
- Wednesday 8 AM - 5 PM
- Thursday 8 AM - 5 PM
- No ATW is generated for Friday as that is the New Year's Day holiday.

- No ATW is generated for Saturday since it is not a business day based on the work profile.
- Due to a snowstorm expected to hit the area on that Tuesday, the company is required to suspend repair work in the area. To achieve that the company enters an override time window that does not allow repair work on that day and lists the specific service area.
- Due to a backlog in repair work, the company needs to allow repair work to be scheduled on that Friday from 9 AM to 6 PM in all areas even though it is the New Year's Day holiday. To achieve that the company enters an override time window that allows repair work on that Friday.
- Next time the Task monitor batch process processes the activity its arrival time windows (ATW) would be regenerated as follows
- Monday 8 AM - 5 PM
- No ATW for Tuesday because of the override to prevent work that day.
- Wednesday 8 AM - 5 PM.
- Thursday 8 AM - 5 PM
- Friday 9 AM - 6 PM because of the override to allow work during that time.
- No ATW is generated for Saturday since it is not a business day based on the work profile.
- New activities sent in from the host would have their ATWs reflect the overrides right away when created.
- On Monday evening the weather forecast cancels the snowstorm alert and the company cancels the override placed on Tuesday.
- The next time the Task monitor batch process processes the activity its arrival time windows (ATW) would be regenerated to add back the ATW for Tuesday.

Maintaining Override Time Windows

This portal is used to display and maintain an override time window event.

Refer to [About Override Time Windows](#) for more information.

You can access the portal from the **Main Menu > Service Management > Override Time Window**. You are brought to a query portal with options for searching for a specific override event. Once an override event has been selected you are brought to the maintenance portal to view and maintain the selected record.

The Override Time Window zone on the portal's **Main** tab page provides information about the override time window, including the various maintenance actions applicable to its current state.

Crews

About Crews

A crew is a uniquely named group of one or more [mobile workers](#) and [vehicles](#) that perform work in [shifts](#).

The specific mobile workers and vehicles that make up a crew may be typically constant, like single person crews, or they can vary by day of week, from shift to shift. A crew might also perform different types of work and cover different service areas on different shifts.

This section describes concepts related to managing crews.

Work Is Scheduled To Crew Shifts

A crew is available to perform work during their shifts. The specific mobile worker(s) and vehicle(s) allocated to a crew may vary from one shift to another, and a crew may perform different types of work and cover different service areas on different shifts. Hence work is scheduled to crew shifts, and not to crews, as they represent the availability of that crew as well as what they should do and where they should work during that time.

[Mobile workers](#) and [vehicles](#) make up the crew's actual resources on a specific shift. A mobile worker or a vehicle may be part of only one crew shift at any given time.

Refer to [About Crew Shifts](#) for more information.

Mobile Workers

A mobile worker is an individual resource that performs work as part of a crew.

This section describes concepts related to managing mobile workers.

Worker Capabilities

An activity may require the mobile workers of the assigned crew to have certain work related [capabilities](#) in order to perform the work. These may be skills or activity specific requirements such as the need for a security clearance, building access, drug testing, permits, language proficiency, etc.

Some capabilities may have effective periods. For example, a skill might require a license that must be renewed every year. The mobile worker record lists the capabilities the individual has acquired over time.

A [mobile worker's type](#) defines worker capabilities that workers holding that job description are assumed to have. This list is used as the default capabilities list when setting up a new mobile worker, or when changing their mobile worker type, but can be further adjusted to remove or add capabilities as needed to describe the individual's actual capabilities.

When scheduling work, the system matches the worker capability requirements of each activity to the capabilities of the workers that make up the crew on a specific shift. It is important to properly describe a mobile worker's actual capabilities to avoid sending them work they are not able to or should not perform.

Base Areas

A mobile worker may be associated with one or more base service areas within which they typically work.

This information is intended to facilitate the search for suitable mobile workers to work in a certain area. It is not used to restrict the mobile worker to work solely in these areas.

Relative Efficiency

This specifies the mobile worker's efficiency relative to expected efficiency for mobile workers of this type.

The scheduler may plan for more or less time to work an activity based on the overall efficiency of the crew on that shift; the overall efficiency is the average of the relative efficiency of all mobile works allocated to that shift.

Home Address

A mobile worker's home address may be used as their logon or logoff location on their shift when the mobile worker is the only person on that crew. For this reason the home address needs to be geocoded so that the scheduler can plan the shift route to start and end from the home location.

Unless explicitly provided the system attempts to geocode the address using the [geocoding service](#) defined on the installation options expecting an exact match.

NOTE: Logic to geocode the home address resides in an algorithm associated with the mobile worker's business object. Review this algorithms and adjust its match quality parameter as needed.

Allow To Work Shift Using SMS Messaging

When a mobile worker uses SMS messaging to work their shift schedule, the system identifies the worker and the shift they work using their mobile phone number. The mobile worker has to be explicitly allowed to use their mobile phone to work their shift this way.

Refer to [Using SMS Messages](#) for more information.

My Calendar

As a best practice, you may want to set the home page for mobile workers to the [My Calendar](#) portal. This allows them to view their shift schedule and any planned leaves on a calendar view.

Vehicles

A vehicle may be needed for the crew to perform their work. The vehicle is mainly used for travel but may also carry pieces of equipment or have certain features that are required to perform certain types of work.

This section describes concepts related to managing vehicles.

Vehicle Capabilities

An activity may require the assigned crew's vehicle to have certain pieces of [equipment](#) in order to perform the work. For example, the crew may need a bucket truck to perform line pole work.

Some equipment may be available on the vehicle for limited time only. The vehicle record lists equipment associated with this vehicle over time.

A [vehicle's type](#) defines equipment assumed to be available on vehicles of that type. This list is used as the default equipment list when setting up a new vehicle but can be further adjusted to remove or add pieces of equipment to reflect those actually available on the vehicle.

When scheduling work, the system matches the equipment requirements of each activity to the equipment actually available on the crew's vehicle on a specific shift. It is important to properly describe a vehicle's actual capabilities to avoid sending the crew work they are not able to perform.

Relative Speed

This defines the vehicle's speed relative to the expected speed for vehicles of this type.

The scheduler may plan for more or less travel time based on the overall speed of the vehicles utilized on that shift; overall speed is calculated as the slowest relative speed of all vehicles in use on the shift.

Fixed Crew Resources

Based on the [crew type](#) configuration, if the resources that make a crew are typically the same on every shift, they may be specified as the default resources on the crew record.

For example, the base product supports a setup for a single person crew where the mobile worker and their vehicle are defined as part of the crew record. Your implementation may support other fixed resource configuration. These resources are then defaulted onto any new shift created for the crew, whether added online or generated by a template.

Note that this does not prevent the crew from having different resources on specific shifts as needed.

Resource Capabilities May Change

When scheduling work, the system matches the requirements of each activity to the capabilities of available crew shifts. The scheduler considers the capabilities of all mobile workers and vehicles allocated to a crew shift to determine which activities the crew can work on a particular shift.

Adding or removing worker capabilities from a mobile worker and/or pieces of equipment from a vehicle impacts all non-finalized crew shifts to which the mobile worker and the vehicle are currently allocated. The scheduler then reevaluates the shift schedules accordingly, and may add or remove activities as a result.

However, if an activity has already been dispatched or marked for dispatch, it remains so. It is the responsibility of the crew to decline an activity they cannot perform. Once the activity is declined it may be rescheduled to another qualified crew.

Crew Locations

A crew may be affiliated with specific locations for various purposes.

For example, if the crew typically starts their shift from the same service center or depot, it can be defined as their default logon location. The system then would then use this information when creating shifts for the crew and a logon location is not explicitly specified.

Planned Time Off

A mobile worker may be on leave for a period of time.

A vehicle may be scheduled for some maintenance work causing it to not be available for a period of time.

A crew may need to attend work related meetings during which they are not available to perform work.

Periods of time when a resource is not available to perform work impacts their available work capacity. This information is recorded in the system using a period of unavailability record.

Refer to [About Periods of Unavailability \(POUs\)](#) for more information.

Crew Hierarchy

Depending on your business practices, crews might be organized within a reporting hierarchy.

Using crew hierarchy can help with reporting and also allows for advanced searching and/or filtering crews in the Gantt and from related records.

Refer to [Understanding Crew Hierarchies](#) for more information.

Depot Restrictions

If your organization uses depots, a crew may be restricted to specific depots they are allowed to visit. If depot restrictions are indicated on a crew, the crew's shift schedule is restricted to the specific depots indicated. If no depots are specified, the crew can travel to any depot as part of their schedule.

If for some reason a crew on a specific shift should not be allowed to visit a depot they are normally allowed to visit, this information can be manually overridden on the specific shift.

Refer to [Depots](#) for more information.

Crew May Be Managed Externally

The product supports the following types of integration flows with a host system.

Crews Managed By Us

In this type of integration the system fully manages crews, schedules work to them, optionally communicates with the crews via a mobile application, and forwards the specific completion information they have provided for each activity back to the requesting host system.

As such, all the information needed to perform an activity as well the data needed to be collected in the field to complete it has to be available to the crew on the mobile application.

The system may be configured to send interim activity status updates to the host or the host may request that on demand.

This is the default configuration.

Crews Managed By The Host System

In this type of integration the system is used solely as a scheduling engine.

Crews are managed by the host system itself which sends in (through synchronization) enough details to generate shifts for scheduling purposes.

Crews communicate directly with the host system which then forwards activity status updates to the current system so as to provide real time feedback to the scheduling process. In the same way, activity completion information is sent directly to the host system and never passes through this system.

To indicate a crew is externally managed, reference the external system that manages it and with which the system integrates.

Refer to [Crews Managed By Host](#) for more information.

Maintaining Mobile Workers

This portal is used to display and maintain a mobile worker record.

Refer to [Mobile Workers](#) for more information about mobile workers and their relationship to crews.

You can access the portal from the **Main Menu > Resource Management > Mobile Worker**. You are brought to a query portal with options for searching for a specific mobile worker. Once a mobile worker has been selected you are brought to the maintenance portal to view and maintain the selected record.

The **Mobile Worker** zone on the portal's **Main** tab page provides information about the mobile worker.

The **Calendar** zone on the portal's **Calendar** tab page provides a calendar view of shifts and other items on the mobile worker's schedule for the current month.

Refer to the Calendar (F1-CALENDAR) zone type for more information.

Maintaining Vehicles

This portal is used to display and maintain a vehicle record.

Refer to [Vehicles](#) for more information about vehicles and their relationship to crews.

You can access the portal from the **Main Menu > Resource Management > Vehicle**. You are brought to a query portal with options for searching for a specific vehicle. Once a vehicle has been selected you are brought to the maintenance portal to view and maintain the selected record.

The **Vehicle** zone on the portal's **Main** tab page provides information about the vehicle.

The **Calendar** zone on the portal's **Calendar** tab page provides a calendar view of shifts and other items on the vehicle's schedule for the current month.

Refer to the Calendar (F1-CALENDAR) zone type for more information.

Maintaining Crews

This portal is used to display and maintain a crew record.

Refer to [About Crews](#) for more information.

You can access the portal from the **Main Menu > Resource Management > Crew**. You are brought to a query portal with options for searching for a specific crew. Once a crew has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page

- **Crew.** This zone provides information about the crew.
- **Shift Weekly Templates.** This zone lists shift weekly templates the crew is associated with over time.

The **Calendar** zone on the portal's **Calendar** tab page provides a calendar view of shifts and other items on the crew's schedule for the current month.

Refer to the Calendar (F1-CALENDAR) zone type for more information.

My Calendar

This portal provides a calendar view of your shifts and other items on your schedule for the current month. The portal is typically used by mobile workers, company or contractor users, as it provides them with an easy access their shifts and their schedules. If they do not use the mobile application to perform their work they may navigate to their daily shift from this portal in order to review their activities and enter completion information. This should also be set as the home page for these users.

You can access this portal from the **Main Menu > Resource Management > My Calendar**.

Resource Route Replay

This portal shows a replay of the actual route crews and vehicles during a specific period of time based on GPS data reported either from the crew's mobile device or directly from a vehicle's GPS system.

This functionality may be used to analyze route patterns and for auditing purposes. You can review routes for up to 12 crews and/or vehicles, with each route displayed in a different color.

You can access this portal from the **Main Menu > Resource Management > Resource Route Reply**.

The following zones may appear as part of the portal's **Main** tab page

- **Crews / Vehicles Search.** This zone is used to select up to 12 crews and/or vehicles to replay their routes. Broadcast each resource and when ready use to the Routes zone to complete the query.
- **Routes.** This zone shows the routes for the crews / vehicles selected using the search zone and for a specified period of time provided by the user.

Refer to [Shift Route Replay](#) view the route for a specific crew shift.

Crew Shifts

About Crew Shifts

A crew shift is a planned period of time in which a crew (one or more mobile workers and vehicles) is scheduled to perform work. For example, a crew shift might be defined as Crew A, consisting of Jack and his van, working on a specific date from 9 am - 5 pm.

Refer to [About Crews](#) for more information.

This section describes concepts related to managing crew shifts.

Crew Shift Type Controls Everything

Each crew shift references a [shift type](#) which defines attributes common to all crew shifts of that type, including the shift start time and duration of the shift etc.

For example, night shifts would have a different start time and duration than morning shifts, so they would have different shift types.

Creating Shifts

Typically, crew shifts are generated by templates that define the basic characteristics of the shifts. A crew's shifts may be generated by a single shift weekly template at any given point in time.

Refer to [About Shift Weekly Templates](#) for more information.

Shifts can also be manually created to address special situations on particular days.

Default Crew Shift Details

When creating a shift for a crew, either manually or via a shift template, certain details may be defaulted from the crew record. For example, a single person crew's record would state the mobile worker and potentially the vehicle that typically make up the crew each day. This information is used as to default the resource allocation for the crew when a shift is created for it online or by the shift generation process.

NOTE: Logic to default resource allocation and any other shift related information from the crew record onto a new shift upon creation resides in a "Prepare Shift Data" algorithm on the crew's business object. Click [here](#) to see the algorithm types available for this system event.

Crew Shift Workflow

The following describes key stages in the overall lifecycle of a crew shift.

Pending

A shift is initially created in this state. While in that state it is not sent to the scheduler and is considered a draft until marked as fully planned.

Planned

A shift in this state is no longer a draft and is available for scheduling as per its configuration.

If generated by a template, the shift is automatically set to the planned state. A user creating a shift online has to explicitly mark it as planned when set up is complete and the shift should be sent to the scheduler.

Started

When the crew starts the shift, their working state is either in service, i.e. performing work, or out of service, i.e. busy with non work related tasks such as taking a break, attend a meeting etc. Refer to [Planned Unavailability](#) for more information.

By default, when the crew starts their shift they are assumed to be out of service until they start an activity at which time the system sets their shift to in service automatically.

Once the shift is started, the system automatically dispatches tasks that are queued for dispatch to the crew. Refer to [Locking The Schedule](#) for more information.

Out Of Service

This state indicates that the crew is not currently working or available for work.

Each time the crew starts a non work related task, such as attending a meeting, their shift is switched back to out of service, if it is not already in this state.

A crew may also be out of service for unplanned circumstances, for example, their vehicle breaks down. The crew may use the mobile application to report they are out of service and for how long. They can also do the same if they use SMS messaging to work their shift. Crews not using any device may choose to notify the dispatcher of their state so it can be reflected in the system.

In Service

This state indicates that the crew is currently working (traveling to an activity location, working an activity, idle between activities, etc.). Each time the crew starts a work related task, such as an activity or a depot task, their shift is switched to in service, if it is not already in this state.

While in service, time between completing one task and before starting another is considered idle time.

Completed

This state indicates that the crew is done working. Activities the crew did not manage to complete are returned back to be rescheduled. Refer to [End of Shift](#) for more information on this process.

Canceled

This state indicates the shift is canceled and should no longer be used for scheduling. If any tasks are currently scheduled to the shift, they will be rescheduled.

Planned shifts that were not started before their planned end time are automatically canceled and removed from the scheduler.

A user or any process can cancel a shift that has not started yet. Once started, it can only be completed.

Controlling The Schedule

This section describes the various aspects of a shift used by the scheduler to control the crew's schedule of work on that shift.

Fixed Shifts

By default, the duration of a shift is fixed and defined as the time between its planned start and end times. The crew is assumed to start their shift at the planned start time and end it at the planned end time.

A default duration is defined on the [shift type](#), allowing different types of shifts to be of different lengths; for example, night shifts could be shorter than day shifts. When setting up a shift online, the system automatically sets the planned end time based on the planned start time entered and the default duration, but allows the user to adjust the shift to be shorter or longer as needed.

While typically shifts are shorter than a day, there is not such limitation in the system; they can be set up to any length as the business requires.

Variable Shifts

When crews work on appointments, there may be a lot of idle time between the time they start their shift and their first appointment. To help with more efficient scheduling, shifts can be set up as "variable shifts" to provide the crew with some flexibility on when they can start and end work.

A variable shift defines a broad time window within which a fixed duration shift is planned based on work load. The shift has to retain a fixed duration so the latest the shift can start is the broad time window end time minus the fixed duration. For these shifts, the scheduler continuously adjusts the estimated start work time as the start time of the first task scheduled to the shift, minus any travel time, but never later than the earliest shift start time.

The estimated start time is locked at some point prior to the broad time window start time so as to give the crew enough time to know when they should show up for work. Once locked, the shift is updated and sent to the scheduler as a [fixed shift](#) that is planned to start at the locked estimated start time and planned to end after the fixed shift duration specified. The estimated start time is rounded down based on the rounding option specified on the shift. The amount of time prior to start that the shift is locked is set by the advanced dispatch offsets shift configuration.

For example, a variable shift with a planned range of 8:00 AM to 8:00 PM and a fixed duration of 9 hours can start anytime between 8:00 AM to 11:00 AM (the start window is defined as the earliest shift start time to the latest end time, minus the duration - a three hour offset in this example). Depending on the shift start time, the shift will end 9 hours later for an end time between 5:00PM and 8:00PM. If the first activity is scheduled after the latest shift start time, the shift will start at 11:00AM to ensure that the requirement for a 9 hour duration is met.

The default duration of the broader flexible period of time is also defined on the [shift type](#) along with the default fixed shift duration. When setting up a variable shift online, the system automatically sets the broad planned end time based on the planned start time entered and the default variable duration but allows the user to adjust the shift to be shorter or longer as needed.

Refer to [Advance Dispatching](#) for more information.

NOTE: [Non productive tasks](#) cannot be created for variable shifts.

Logon / Logoff Locations

The scheduler considers the crew's location at any point in time, including where they start and end their shift, to determine the most optimal route for them.

A crew shift must either reference a service center location or indicate the [mobile worker's home](#) as their logon and logoff locations. The latter is only applicable when the shift is for a single worker.

You may specify a maximum travel time the crew is expected to spend traveling from/to the logon/logoff location on their own time. You can also reserve time at the beginning and/or end of the shift to perform review and preparation work.

Crew Allocation

A crew on a shift has to include at least one mobile worker. A vehicle is not required unless it is needed by the crew to perform their work. Multiple mobile workers and vehicles may work together as a crew on a specific shift.

A crew and any of its allocated resources can only be part of one shift at any given point in time. This means shifts cannot be planned for the same crew or mobile worker or vehicle to overlap in time.

This section describes concepts related to allocation of resources to a shift.

Crew Allocation May Change At Logon

When a crew leader starts their shift, the additional resources that make up the crew that day may differ from those originally planned. For example, if a worker reported sick and is replaced by another worker that day, the crew leader can specify the new worker instead of the sick person.

Changes to the crew allocation are communicated to the scheduler, which may reconsider the planned work for the shift and rebuild the shift schedule as needed to reflect the [crew's current capabilities](#).

Resource allocation may also [change throughout the shift](#).

A crew and any of its allocated resources can only be part of one shift at any given point in time. Whether at shift start time or later, the system always validates changes to crew allocation to ensure neither crew nor resources are already part of another started or completed shift at this time.

Crew Allocation May Change Mid Shift

There are situations where the crew's resource allocation may change mid shift.

For example, a mobile worker may leave their crew mid shift for some reason or their vehicle breaks down and is replaced. In this situation, the crew's schedule for the remaining of their shift may be adjusted to meet the crew's [recent change of capabilities](#).

The scheduler reevaluates not yet dispatched work and may add or remove activities as a result. However, if an activity has already been dispatched or marked for dispatch, it remains with the crew. It is the responsibility of the crew to decline an activity they can no longer work. Once the activity is declined, it may be rescheduled to another qualified crew.

The system keeps track of all the resources that were part of the crew throughout the shift for reporting and timekeeping purposes.

Crew Capabilities

The scheduler assigns an activity to a crew that is capable of carrying it out. The crew's overall capabilities on a specific shift are those associated with the mobile workers and the vehicles [allocated](#) to that shift.

Refer to [Worker Capabilities](#) and [Vehicle Capabilities](#) for more information.

By default a crew is considered having a certain capability on a specific shift if at least one of the resources (worker or vehicle) allocated on that shift possess it. If a capability has to be possessed by all members of the crew, i.e. all workers for a worker capability or all vehicles for equipment, then the [capability](#) itself needs to be defined as required by all. In this situation, the crew shift is considered having the capability if all relevant resources have it.

Covered Areas

A crew may work in specific service areas on a shift. The scheduler would only assign the crew work within these service areas. A shift must cover at least one service area.

A shift may be configured to prefer work in some service areas over others. The scheduler is encouraged to assign work in the preferred service areas over other areas covered by the shift.

Primary Function

Types of work may be classified by service class. One of these service classes can be designated as the primary function of the crew on a specific shift. This encourages the scheduler to prefer work of that class, if any, over other classes of work.

Not Allowed Work

A crew shift may list specific service classes of work that the crew should not be assigned on that shift, regardless of whether they are capable of performing that work or not. The scheduler would not assign work that belongs to these service classes to that shift.

Reserve Capacity

A certain percentage of shift capacity may be reserved for up to 10 different categories of work.

Common reasons to reserve capacity on a shift may be as follows:

- To offer appointments up until the day before, your organization would like to reserve a certain percentage of a shift's time for appointments so that other non-appointment work does not take precedence over appointments.
- Your organization may have a lot of emergency work during certain months of the year for which they need to reserve a certain percentage of a shift's time for emergency work so that it does not get filled up with appointments that might be missed if pushed out to accommodate emergencies.
- Some time may be reserved for non appointment low priority work to handle backlogs.

A work category may be broadly defined for emergency work, appointments or non appointment work, and may be further restricted to specific activity types by referencing a service class.

Each reserve states the minimum and optionally, a maximum percentage of shift time to reserve for that category of work. Note that the maximum percentage is not applicable to emergency work as it needs to be addressed as soon as possible without any capacity restriction.

In addition, each reserve defines whether the reserved capacity should be released some time before or after the shift is planned to start, or not released at all.

For example, you might reserve 10% of a shift's time for emergency work that would never be released and 50% for appointment work that would be released 1 hour prior to planned shift start time.

Shift Cost Profiles

A shift is associated with a shift cost profile to define how the scheduler should cost travel time, overtime and other scheduler costs for this shift relative to other shifts.

Refer to [Understanding Shift Cost Profiles](#) for more information.

Depot Restrictions

If your organization uses depots, and for some reason a crew on a specific shift should not be allowed to visit a depot they are normally allowed to visit, this information can be overridden on the specific shift.

Refer to [Depots](#) for more information.

Standby Shifts

Standby shifts are potential shifts that can be activated only when needed. They are not considered as available capacity, and as such are not scheduled until the crew actually starts the shift or the dispatcher [opens](#) it to be scheduled as usual.

For example, a crew may be planned to have a stand by shift on Sundays to handle emergencies. If the crew is indeed called on that Sunday and starts that shift, it becomes an active shift as usual and gets sent to the scheduler.

A shift is typically marked as "stand by" when created, but it can also be changed to standby while this shift is still in the pending state.

Time Zones

The time zone on a shift indicates the time zone in which the crew is planned to be working. All date and time information on the shift portal, including set up of breaks and non productive tasks for the shift, is entered and displayed in that time zone.

Crew Shift Actions

A dispatcher or a business process may perform various actions on a crew shift that may impact the shift's state and schedule.

The following section describes these actions and the business flows associated with them.

Allocate All

[Allocates](#) all of the activities currently scheduled to the shift that have not yet been dispatched.

NOTE: This action does not [close](#) the shift which means more work can be assigned to it by the scheduler.

Open

A user or any business process can open a [closed](#) or [standby](#) shift's schedule to allow the scheduler to assign more tasks to the shift, reassign work to another shift, and adjust the schedule as needed.

Close

By default and unless explicitly set up to be a [standby shift](#), the schedule on a shift is open for the scheduler to plan and optimize continuously. Closing a shift [allocates](#) all scheduled activities to it. The scheduler is prevented from removing any of the allocated activities or adding more activities to the shift.

When closing the shift you may also [manually dispatch](#) the allocated activities at the same time.

A shift can be closed in the following ways:

- A shift may be configured to automatically close (and queue for dispatch all tasks) some time prior to its planned start time. This is typically useful for crews that do not use a mobile device to work their schedule (non-MDT). Refer to [Advance Dispatching](#) for more information.
- A dispatcher can manually close a shift so as to finalize its schedule and prevent more work from being assigned to the crew. The dispatcher can re-open the shift at any time.
- If the crew goes offline, then all scheduled tasks are dispatched to the crew and the shift is closed. Refer to [Offline Mode](#) for more information.

Force Logoff

This action is only applicable to crews that utilize the mobile application to work their schedule.

When a crew uses the mobile application to start their shift, the system stamps the identifier of the device (MDT) they are using on the shift record. Message exchange between the crew and the system is assumed to be made to and from this device only.

This action forcibly logs off the association between the shift and the device currently being used to update it. This may be required if a logged on crew's mobile device is not working and they need to receive data on a new device. Once the shift is forced logged off, the crew can now access the shift from another device which then gets stamped on the shift as the current device being used by the crew.

NOTE: It is assumed that data on the old device is not recoverable. Any messages that may arrived stamped with the old device identifier would be discarded to avoid contradictory updates from multiple devices.

NOTE: This action solely disconnects a shift from a device and does not impact the status of the shift.

Planned Unavailability

There may be planned periods of time during a shift when the crew is not be available to perform work, for example, when they take their lunch break, stock their vehicle in the morning, or attend a meetings.

These non work related tasks consume time on a shift leaving less time to perform work. When planned in advance as part of the shift configuration, the system has a more accurate account of how much time remains available on a shift for work related tasks.

While performing these tasks, the crew is considered out of service and when they are completed, the crew is assumed in service again.

Refer to [Breaks](#), [Non-Productive Tasks \(NPTs\)](#) and [Meetings](#) for more information

Breaks

Crews may be entitled to take planned breaks after working some amount of time on their shift.

For example, a shift might include two 15 minute breaks, one in the morning and one in the afternoon. The morning break could be taken anytime from 1 to 2 hours after the crew has started their shift and the afternoon break could be taken anytime from 5 to 6 hours into the shift.

The following section describes key concepts related to planning and scheduling breaks.

Planned Relative To Shift Start Time

Breaks are planned to be scheduled during a time window starting at an offset relative to shift start time. Each break is a task associated with a break type. The duration of the break as well as the duration of the window during which the break can be taken is defined on the break type, whereas the earliest time it can be taken, i.e. the offset from shift start time, is specified on the break task.

For example, if the offset is 2 hours and the shift starts at 8 AM, the break cannot start before 10 AM. If the shift starts at 9 AM with the same 2 hour offset, then the break task can be scheduled as early as 11 AM.

NOTE: Break tasks are only scheduled if enough work is scheduled to warrant a break from work.

No Location

Unlike other classes of tasks that take place at a specific location, a break does not have a defined location. The scheduler may schedule it to be taken anywhere on the way to a task location. A break may also be planned to be taken on site while working on a long duration activity, if permitted by the activity type.

If your company uses depots, the break may be scheduled to be taken at the depot while the crew is there to load or unload goods.

Creating Breaks

While you can manually add a break to an existing crew shift, breaks are typically incorporated into planned shifts at shift template design time.

Refer to [Planning Breaks](#) for more information.

Break Workflow

The following describes key stages in the overall lifecycle of a break.

Pending

A break is initially created in this state waiting to be scheduled and [queued for dispatch](#).

Queued For Dispatch

Depending on shift configuration, the crew may receive their entire schedule for the day or the first few tasks dispatched to them only when they start their shift (which they can preview in advance).

Refer to [Locking The Schedule](#) for more information.

Similar to an activity, when it is time to make the task available for the crew to preview or actually dispatch it to a crew's mobile device, the system queues it for dispatch so that its schedule is locked and changes can no longer be made once communicated to the crew.

Setting the break to the **Queued For Dispatch** state is the first step in the dispatch process.

The following business events queue breaks for dispatch

- If a crew's shift is configured to support [Advance Dispatching](#), the configured number of first tasks they may be previewed are queued for dispatch at a predefined time prior to the shift's planned start time.
- When a crew starts their shift, the scheduler automatically queues for dispatch (if not already queued in advance) their first few scheduled tasks, all of them or none, as per the shift's drip mode configuration. If a drip horizon is used, each time the crew starts or completes a task the scheduler proactively queues the next one for dispatch. Refer to [Controlling Auto-Dispatching](#) for more information

NOTE: Queuing a break task for dispatch should not be confused with the actual process of dispatching the task to the crew's mobile device. These are consecutive but separate steps.

Actual dispatching can only happen once the crew has started their shift and is ready to work. Moreover, the device the crew uses to work their schedule is only known once they use it to start their shift. Once known the device identifier is stamped on the started shift and the system directs all communications to the crew to that device. If the task was placed in the queue in advance, prior to the crew starting their shift, then it can only be dispatched once the shift starts, otherwise it is dispatched right away.

NOTE: When conditions are met, the task would be automatically dispatched without relying on any background process to move it along. Even when the task is advance dispatched, it would automatically be dispatched when the shift starts.

When ready to be dispatched, if the crew uses a mobile device, a copy of the task is sent to the crew's device. If the crew uses SMS to work their tasks, a message with brief description of the task is sent to their mobile phone. Refer to [Keeping Mobile Copy Up To Date](#) for more information.

Refer to [Dispatching A Break](#) to better understand the big picture of the dispatch process.

Dispatched

The mobile application acknowledges receipt of the task by sending a message to the server to update the task to the dispatched state.

If the crew does not use a mobile device, the task is automatically marked dispatched.

Refer to [Dispatching A Break](#) for more information.

Started

This state indicates that the crew has started to take their break and are not working at the moment. While taking a break, the crew is considered out of service.

The crew provides this status update when they take this planned break. Task status updates by the crew are similar to assignment updates. Refer to the assignment [status update](#) for more information.

Time spent while in this state is considered break time for [crew time](#) reporting.

Completed

This state indicates that the crew is no longer at break and is back in service .

The crew provides this status update when they resume back to work. Task status updates by the crew are similar to assignment updates. Refer to the assignment [status update](#) for more information.

Canceled

A break in this state indicates it is no longer needed and should not be scheduled. Typically breaks are canceled before they are dispatched, but a dispatcher may decide to cancel a break at any time.

The crew may also decide to not take a planned break and cancel it. Task status updates by the crew are similar to assignment updates. Refer to the assignment [status update](#) for more information.

Once a shift is finalized all breaks that were not taken are automatically canceled.

Dispatching A Break

The process of dispatching a break to the crew involves multiple steps.

- **Step 1 - Placed in queue.** The break has to be placed in the queue for dispatch for the dispatch process to begin. This step is typically performed by the scheduler as it auto dispatches work, but other processes may do so as well.
- **Step 2 - Wait.** The break remains queued until it is actually dispatched and acknowledged (see step 4).
- **Step 3 - Dispatch.** Once all conditions are met for the break to be dispatched, the system sends a copy of the break task to the crew's mobile device, if any. Refer to [Keeping Mobile Copy Up To Date](#) for more information.

Refer to the [Queued For Dispatch](#) for more information about the processes involved in these steps.

- **Step 4 - Acknowledgement.** For crews using the mobile application the break is considered dispatched when the mobile application acknowledges receiving it. For all other crews a break is automatically acknowledged if the shift is started.

Refer to the [Dispatched](#) for more information about the processes involved in this step.

Keeping Mobile Copy Up To Date

If the crew is using the mobile application to manage their work, they receive a copy of each of their tasks sent to their mobile device from the moment they start their shift.

When key activity information changes about the task after it was dispatched, the system automatically synchronizes the mobile copy of the task and sends the crew a new copy of it with the latest information.

NOTE: Logic to initially send a copy of the task to the crew's mobile device as well as synchronize the mobile copy to keep it up to date when changes are made on the server resides in a "Synchronize Task" algorithm on the task's business object. Click [here](#) to see the algorithm types available for this system event.

Non-Productive Tasks (NPTs)

Crews may need to restock their vehicles or perform other administrative tasks at a specific location when they begin their shift or sometime after. These tasks are referred to as non productive tasks (NPT).

For example, a shift might start with a 30 minute task to replenish supplies on the truck from the service center in the morning. The task might take place anytime up to an hour at the beginning of the shift.

The following section describes key concepts related to planning and scheduling these tasks.

Planned Relative To Shift Start Time

Similar to breaks, NPTs are planned to be scheduled during a time window starting at an offset relative to shift start time. Each task is associated with a non productive task type. The duration of the task as well as the duration of the window during which it can be taken is defined on the task type, whereas the earliest time it can be taken, i.e. the offset from shift start time, is specified on the task itself.

For example, if the offset is 1 hour and the shift starts at 8 AM, the task cannot start before 9 AM. If the shift starts at 9 AM with the same 1 hour offset, then the break task can be scheduled as early as 10 AM.

Unlike breaks, NPTs are associated with a location where the task is performed. The location of an NPT may reference a common facility location or a specific address. A crew may be affiliated with a specific NPT [location](#) defined on the crew record. When creating an NPT, if its location is not explicitly specified, the system defaults it from the shift's crew record, if any is defined.

The scheduler needs the geocode of the task location in order to incorporate the task as part of the shift's route. If the task references an address and a geocode is not provided, the system attempts to geocode the address using the [geocoding service](#) defined on the installation options expecting an exact match.

NOTE: Logic to geocode the home address resides in an algorithm associated with the task's business object. Review this algorithms and adjust its match quality parameter as needed.

Creating NPTs

While you can manually add an NPT to an existing crew shift, these tasks are typically incorporated into planned shifts at shift template design time.

Refer to [Planning Non Productive Tasks](#) for more information.

NOTE: Non productive tasks cannot be created for [variable shifts](#).

NPT Workflow

The workflow of an NPT is similar to the workflow of a break, with the exception of handling the travel to the NPT location. As with other tasks associated with a location, the crew indicates they have started traveling to the NPT location and once there, they can start performing the task.

The following describes key stages in the overall lifecycle of an NPT.

Pending

An NPT is initially created in this state waiting to be scheduled and [queued for dispatch](#).

Queued For Dispatch

Setting the task to the **Queued For Dispatch** state is the first step in the dispatch process.

This is similar to the process of queuing breaks for dispatch. Refer to the break task [Queued For Dispatch](#) state for more information.

Dispatched

This is similar to the process of marking breaks as dispatched. Refer to the break task [Dispatched](#) state for more information.

En route

This state indicates that the crew is currently on its way to the task location. The crew provides this status update when they are about to start traveling to the task location. If the crew was previously in service, they are now considered out of service.

The crew provides this status update when they are about to start traveling to the location. Task status updates by the crew are similar to assignment updates. Refer to the assignment [status update](#) for more information.

Time spent while in this state is considered travel time for [crew time](#) reporting.

Started

This state indicates that the crew has started to perform the task associated with the NPT.

The crew provide this status update when they start the task. Task status updates by the crew are similar to assignment updates. Refer to the assignment [status update](#) for more information.

Time spent while in this state is considered NPT time for [crew time](#) reporting.

Completed

This is similar to the process of marking breaks as complete. Refer to the break task [Completed](#) state for more information.

Canceled

This is similar to the process of marking breaks as complete. Refer to the break task [Canceled](#) state for more information.

Meetings

Crews may need to attend work related events such as training sessions and meetings, also referred to as periods of unavailability (POUs). If the event overlaps with any of their planned shifts, the scheduler factors this event as part of the shift route.

For example, the crew can be out working activities, attend a training meeting and then return to work.

Refer to [Crew Meeting POUs](#) for more information

Locking The Schedule

Dispatching is an action that locks a task's sequence in the shift's schedule so no further optimization takes place, and marks it as ready to be sent to the crew.

This section describes shift configuration options related to dispatching.

Controlling Auto-Dispatching

Once a crew starts their shift, you can configure the scheduler to automatically dispatch to the crew all or a subset of their work without requiring dispatcher intervention.

This applies to any task, not just activities.

This is controlled by the shift's drip mode that supports the following options:

- **None.** Automatic dispatching is disabled. All scheduled tasks must be manually dispatched to the crew by the dispatcher or by another process.
- **All.** All scheduled tasks are automatically dispatched to the crew. Additional tasks scheduled throughout the day are also automatically dispatched. Activities for which automatic dispatch is **disabled** would still be marked as ready to dispatch by the scheduler, but they would rely on a dispatcher to manually dispatch them and place them in the dispatch queue.
- **Standard:** Only a fixed number of scheduled tasks are automatically dispatched to the crew at any given point of time. The number of tasks to dispatch at any point in time is referred to as the drip horizon. Activities for which automatic dispatch is **disabled** will not be assigned to shifts using this drip mode.

The intent behind the more progressive standard drip mode is to dispatch just enough to keep the crew busy and allow the not yet dispatched portion of their route to be flexible enough to be rescheduled to other crews as exceptions arise throughout the day.

Depending on scheduler configurations, non work tasks, i.e. breaks, NPT and meeting POU tasks, may be automatically dispatched when a crew starts their shift, even if they are not yet within the dispatch horizon.

Refer to [Auto-Dispatching](#) for more information.

Advance Dispatching

Regardless of the automatic dispatching setting, you can also configure the system to queue for dispatch a number of tasks in advance of the shift start. This would ensure the schedule of these tasks is locked for various business needs. For example, it allows the crew to preview the first few tasks of their schedule before starting their shift.

It is also at this time that variable shifts become fixed shifts. Refer to [Variable Shifts](#) for more information.

The amount of time prior to shift start that the shift's schedule is locked is set by the advanced dispatch offset specified on the shift. Using the advance dispatch mode, you can also indicate that at the same time the shift should be closed and all the tasks scheduled to it should be queued for dispatch.

Closing the shift and queuing for dispatch all its scheduled tasks is also referred to as the "shift cutoff".

NOTE: This functionality relies on the **Crew Shift Monitor** batch process to be scheduled in appropriate frequency. Refer to [M1-CRSHF](#) batch control for more information.

Crew Undispatch and Sequence Lock

Some business practices require the dispatched route to be completely locked, while others may allow it to change for a significantly more optimal route and respond to real-time exceptions.

The system supports the ability to configure a shift to allow or disallow route changes in the form of undispatching tasks or changing the order of the dispatched tasks.

Offline Mode

Some crews may use no mobile device or may operate in areas with poor or no connectivity coverage. These crews don't have the means to automatically receive new work nor changes to their route.

In this situation, there is no point in keeping their schedule open. These shifts needs to be closed with their entire schedule locked before the crew is out in the field with no means of connection.

Closing the shift and locking its schedule ensures the route remains unchanged. But in this situation, since the crew cannot provide status updates as to their progress, the scheduler does not have a way to validate whether or not the crew is on schedule. This in turn may result in false alerts being raised, for example, reporting that the crew is late for an appointment when they actually made it on time but could not or did not report it promptly.

To prevent scheduling discrepancies, the crew's shift has to not only be closed but also set to offline mode. In this mode, the scheduler assumes the crew is on track and does not push back the schedule. Also, base product alerts are designed to ignore shifts in offline mode.

NOTE: A shift has to be closed before it can switch to offline mode.

Shifts for crews that do not use a device are typically configured to close some time before the shift's planned start time. When the crew starts the shift, or a dispatcher does it on their behalf, the system automatically switches the shift to offline mode.

Refer to [Using No Device](#) for more information.

Crews using the mobile application may explicitly request to switch to offline mode before they travel to an area with poor connectivity for a long period of time. The request closes their shift, dispatches their entire schedule to the mobile device, and switches to offline mode. Note that closing the shift automatically allocates all scheduled activities to that shift and they remain allocated for the duration of the shift. Refer to the [Close](#) shift action for more information. The crew can request to switch back to online mode again, as needed.

Refer to [Switch To Offline Mode](#) for more information.

Working The Schedule

Crews may use mobile devices to carry out their scheduled work and provide real time updates as to their progress. Real time updates from the crew increases the accuracy of the information available to the dispatcher as well as the scheduler when making scheduling decisions and handling exceptions.

Crews may use the mobile application if installed on their device or provide simpler status updates using SMS messaging.

Crews that do not use a mobile device typically record their work on paper or other means, and enter the information via the main application at the end of their shift.

This section describes topics related to performing the work scheduled to a crew's shift.

Using The Mobile Application

The system supports dispatching work directly to mobile application installed on mobile devices including laptops, tablets and other devices.

The mobile application allows the crew to fully manage their work schedule, get more work as needed, handle exceptions as well as seamlessly report their whereabouts.

When the crew starts the shift from their mobile device key details about their shift are brought over and stored locally. If these details change later on the system automatically synchronizes the mobile copy of the shift and sends the crew a new copy of it with the latest information.

NOTE: Logic to initially send a copy of the shift to the crew's mobile device as well as synchronize the mobile copy to keep it up to date when changes are made on the server resides in a "Synchronize Shift" algorithm on the task's business object. Click [here](#) to see the algorithm types available for this system event.

Refer to [About The Crew's Application](#) for more information

Using SMS Messages

Crews may use SMS messaging to carry out their scheduled work. This method allows the crew to start their shift, get their tasks one at a time, provide status updates and complete their shift.

The following scenario illustrates how SMS messaging works

- A mobile worker sends a text message of "Logon". The system identifies the mobile worker via the [phone number](#) the text was sent from, starts their next shift, and sends details of their next task via SMS.
- As the crew works the task, they send messages to communicate their progress, such as "1" to indicate that they are en route to the task's location, and "2" to indicate arrival at the task location and starting to perform the work.

- When work is done, the crew sends a text message of "3". In response, the system sends the next task and so on. If this was an activity, this tells the system work is done but still waits for completion information to be keyed in. The crew would typically enter the completion details at the end of their shift when they return to the office. Refer to [Work Done](#) assignment state for more information.
- At the end of the shift, the crew sends a "Logoff " message.

The following are examples of the messages sent to crews when tasks are dispatched.

Activity Details MREAD at 12.00.00 for Swetha Paspuleti at 6284 dressler Road NW, North Canton

POU Details STATUS MEETING at 6284 dressler Road NW, North Canton from 14.00.00 to 14.30.00

Break Details COFFEE from 13.00.00 to 13.30.00

NOTE: Logic to process incoming SMS messages from the crew and send messages back to the crew resides in algorithms on the installation options record.

NOTE: Click [here](#) to see the SMS Receive algorithm types available for this system event.

NOTE: Click [here](#) to see the SMS Send algorithm types available for this system event.

Using No Device

Crews that do not use any device to manage their work may be provided with a printout of their scheduled route at the beginning of their shift, or they may rely on the dispatcher to communicate the details of their next task verbally as they work them one at a time.

NOTE: Functionality around handling a shift's schedule printout is not provided by the base product. If needed, this is assumed to be a customization effort.

Since they don't have a way to provide real time updates from the field, crews typically rely on communicating them directly to the dispatcher who performs these changes on their behalf. For example, the dispatcher can start their shift, update the status of their next task to state they went en route to the task location, update it to note they have arrived, as well as mark the [work done](#), etc.

Throughout the day, the crew records their completion information on paper or other means and when back at the office, they use the main application to enter the data and complete the corresponding activities.

As the crew does not have a device to automatically receive new work nor changes to their route, there is no point in keeping their schedule open. These shifts therefore are typically configured to close and lock the entire schedule some time before they are planned to start the shift.

Refer to [Advance Dispatching](#) for more information.

If the shift is closed in advance, when the crew starts the shift or the dispatcher starts it on their behalf, the shift is automatically switched to offline mode.

Refer to [Offline Mode](#) for more information.

Auto-Completion

By default, the system assumes that work scheduled to a shift that was never started was not worked and therefore needs to be rescheduled. Expired shifts are automatically canceled and all the work scheduled to them is rescheduled.

However, some customers use the product purely for scheduling purposes where activities and shifts are never updated by the crew. No completion information is recorded in the system to be sent back to the host.

To accommodate this business use case, shifts can be configured to automatically complete as per their planned schedule assuming all scheduled tasks are worked as planned. If a shift is set up to automatically complete, the system starts it at the

planned start time, progresses all its scheduled tasks as planned, and finally automatically completes it at the planned end time.

If there is an exception in working an activity in the field, it is assumed to be handled directly in the host system and resent as needed.

NOTE: The **Shift Auto Complete** [global configuration](#) parameter needs to be enabled before you can set up shifts to auto-complete.

NOTE: This functionality requires the shift record and the auto-completion sync request that manages its over time to be periodically monitored. The **Crew Shift Monitor** batch process initiates the auto-completion sync request when the shift is planned to start and **Sync Request Monitor** batch process advances the shift and its scheduled tasks as planned. It is important that these batch processes are ran in a timely manner. Refer to [MI-CRSHF](#) and [F1-SYNRQ](#) batch control for more information.

Working One Task At A Time

At any point in time during their shift, the crew is assumed to be working on a single task. Attempting to start a task when another is still in progress is not allowed. In the same way, a shift cannot be completed while a task is still in progress.

End of Shift

When the crew completes their shift the following steps are involved:

- Activity assignments the crew did not manage to complete are returned and the corresponding activities are rescheduled.
- All other types of tasks that were not completed are canceled.
- Allocated activities no longer remain allocated to the shift, allowing them to be rescheduled to other shifts.
- If timesheet functionality is enabled, the timesheet records are prepared for each period a mobile worker worked as part of the crew on the shift. Refer to [About Timesheets](#) for more information.

About Shift Weekly Templates

While crew shifts can always be manually created to address a specific situation, they are typically planned and generated ahead of time based on recurring templates. Templates provide a tool for quickly generating large numbers of shifts based on predefined parameters.

A shift weekly template defines a cyclical weekly pattern made up of weekday shift templates.

For example, you might define a weekly template in which the crew works its regular 9-5 shift Monday -Friday every week and also works an on-call shift on Saturday and Sunday every third week. Thus the shift weekly template would rotate every 3 weeks.

Resource planners can create shift weekly templates for specific crews, or they can create common templates and subscribe multiple crews to one template.

This section describes concepts related to setting up and maintaining shift weekly templates.

A Weekly Template Is Made Of Shift Templates

A shift weekly template is made of one or more shift templates that describe the recurring shift patterns. The details for each shift within a cycle are captured on a shift template record. For example, a weekly template might use a different shift template on Fridays than the one used on other days.

This section describes concepts related to setting up and maintaining shift templates.

Actual vs. Template Shifts

A shift template defines the details needed to generate actual shifts based on that template. Any detail you can define for an actual shift, you can also predefine on a shift template.

To enforce the close pairing between a shift template and the shifts it creates, both the actual and the template records are stored in the same table so they can share similar details. The same crew shift type defines both the business object used to create actual shifts of this type as well as the business object used to create shift templates of that type.

Refer to [Defining Shift Types](#) for more information.

A generated shift maintains a reference to the shift template that was used to generate it so that changes made to a template may be reapplied to those shifts.

Complete vs. Partial Shift Details

If the shift weekly template is dedicated to a [specific](#) crew, each shift template may include the complete shift information needed to generate a shift.

A shift weekly template may be defined as a common template to be [shared](#) by multiple crews. As such, details unique to each crew such as the crew and the resources allocated to it cannot be specified on the common template as they differ for each crew. The crew specific details are taken from the crew record at shift generation time.

Refer to [Shift Generation](#) for more information.

Planning Breaks

When you define shift templates, you can plan one or more break types for the shifts. When the shifts are generated, the system automatically creates breaks and incorporates them into the crew shift schedule.

Please refer to [Breaks](#) for more information.

Planning Non Productive Tasks

When you define crew shift templates, you can plan one or more non productive task types for the shifts. When the shifts are generated, the system automatically creates non productive tasks and incorporates them into the crew shift schedule.

The location of the non productive task is taken from the crew record at shift generation time.

Please refer to [Non Productive Tasks](#) for more information.

Crew Specific Weekly Template

Crew specific weekly templates are dedicated to a specific crew and cannot be shared with other crews. As such, each weekday shift template is fully defined with everything that the shift generator needs to generate a shift for this crew.

The initial date of the cyclical weekly pattern is the effective date of the template.

This type of weekly template references the crew it is designed for, as well as the resources allocated to the crew, on each shift template. This allows for more complex situations where different resources are allocated to the same crew on different days of the week.

This section describes concepts related to setting up and maintaining crew specific weekly templates.

Defining and Applying A Template

When you create a new shift weekly template, its status is **Pending**. No shifts are generated while the template is in this state allowing you some time to properly define the weekday shift templates and review them as needed.

After you have defined the shift templates, you must apply the weekly template so that actual shifts will be generated from it, based on the template's effective period. Applying the template changes its status to **Applied** and the system right

away generates future shifts starting from the template's effective date (or from now if that date is in the past) and until the template's expiration date (if any).

NOTE: Generating shifts online is limited to a preconfigured number of days in order to avoid the apply action from taking too long and time out. The parameter is defined on the shift generation algorithm.

Once applied, the shift generation background process manages the generation of shifts going forward.

Refer to [Shift Generation](#) for more information about the generation algorithm as well as the overall process.

Making Changes To An Applied Template

Extending or shortening the effective period of an applied template adjusts future shifts accordingly as soon as you make the change.

If you attempt to add, delete, or modify a shift template for an applied weekly template, the system automatically creates a draft copy of the entire weekly template it belongs to where your change is made to the draft copy, leaving the original template unchanged. This allows you to make changes in a more controlled way.

You make changes to the draft copy and preview the shifts that would be generated should you accept the draft. If you decide to accept the draft, the original template is updated to match the draft and changes are immediately applied to already generated future shifts accordingly.

Refer to [Draft and Original Templates](#) for more information.

Common Weekly Templates

You can set up common weekly templates which can be shared across multiple crews. This is helpful for the efficient generation of shifts in organizations where there are sometimes hundreds or possibly even thousands of crews that follow the same weekly shift pattern.

This section describes concepts related to setting up and maintaining common weekly templates.

Subscription

The common template defines the initial date of the cyclical weekly pattern as well as the details for each shift template. These shift templates cannot include crew specific information, such as the resources allocated to the crew on that shift etc, as they need to be shared by multiple crews.

You may subscribe multiple crews to a common template as their template for generating shifts. Shifts generated from the common template are exactly the same except for crew specific details [defaulted](#) from the crew record such as allocated resources. Any crew that subscribes to a common template must have these types of details defined on the crew record.

Subscription to a common template is effective dated so that crews can subscribe to follow a common template as of a given date and this relationship can expire. Each crew can subscribe at different times, but their shifts are generated in the same rotation timing as all other subscribed resources, regardless of when they subscribed.

You may also unsubscribe a crew from the common template at any time or change effective dates to affect the creation of future shifts for those crews.

Activation

In a typical workflow, you would create a common weekly template and then subscribe multiple crews to the template. When template and subscriptions are ready to be used to generate shifts, you activate the template. Shifts are generated only when the common template is active.

Since a common template may involve a large number of subscriptions, actual shift generation is performed the next time the shift generation batch process runs. Note that you may still test your template and process it online for a subscribed crew without waiting for the next scheduled batch run.

Refer to [Shift Generation](#) for more information.

Making Changes To An Active Template

Changes made to an already active common template affect future shifts of all subscribed crews. Since a common template may involve a large number of subscriptions, such changes are not applied right away to all these shifts but rather they take effect the next time the shift generation batch process runs. This includes extending or shortening the effective period of an active template, subscribing or unsubscribing a crew etc.

If you want to test your recent changes without waiting for the next scheduled batch run you can do so by processing them online for one of the subscribed crews.

Similar to crew specific templates, if you attempt to add, delete, or modify a shift template for an activity weekly template, the system automatically creates a draft copy of the entire weekly template, leaving the original template unchanged.

This allows you to make changes in a more controlled way. You make changes to the draft copy and preview the shifts that would be generated should you accept the draft. If you decide to accept the draft, the original template is updated to match the draft. However, unlike crew specific templates, the impact on these changes on future shifts only takes effect the next time the shift generation batch process runs.

Refer to [Draft and Original Templates](#) for more information.

One Effective Shift Weekly Template For A Crew

A crew's specific weekly templates and subscriptions to common templates are either effective for a specified number of weeks starting from their effective date, or they may be perpetual, which means they have an effective date but no end date.

The crew can switch from a specific weekly template designed just for this crew or a subscribe to a common template, and vice versa, over time, but their shifts may be generated by a single weekly template at any given point in time. In other words, a crew's weekly templates must never overlap in time.

Shift Generation

The system generates future shifts for crews based on the weekly templates with which they are associated.

Common templates do not generate shifts as they are not defined for specific crews. Only a [crew specific](#) weekly template and a crew's [subscription](#) to a common template are used to generate shifts for the crew throughout their effective period. Unlike a crew specific weekly template that is self contained, a crew's subscription to a common template sets the shift generation effective period, whereas template details as well as the initial rotation date are defined on the common template.

Shifts are generated based on the cyclical pattern of the weekly template, starting from the weekly template's effective date (or from now if in the effective date is in the past) and up to a predefined number of days known as the planning horizon, or earlier, if the template expires before then.

The planning horizon is defined as a [global configuration](#) parameter and is used by the shift generation process to control the maximum number of days of shifts to generate and have available for planning purposes, at any point in time.

If the template references a work calendar, the process skips generating shifts that even partially overlap with a public holiday.

The first few shifts may be initially generated online when you apply a crew specific template or verify the setup for a crew's subscription to a common template, but the ongoing process of generating additional shifts to meet the planning horizon is handled by the weekly template monitor process.

NOTE: The shift generation process relies on the shift weekly template being periodically monitored by the **Shift Weekly Template Monitor** batch process as often as appropriate to meet your organization's needs. Refer to [M1-SHWKT](#) batch control for more information.

The background process is designed to add new shifts as needed on a daily basis. However, if any shift template was changed, the process would automatically regenerate future shifts that were earlier generated by that template to match its latest version. By default, even shifts that were manually overridden are regenerated to match the latest template version, but you may explicitly indicate to retain these changes when making a shift template change.

Refer to [Overridden Shift Details](#) for more information.

When a shift is about to be generated but happens to overlap with another shift for the crew that was not generated by this template, the generation of the shift is skipped and a user is notified by raising a to do entry that reports the situation.

NOTE: Logic used to generate shifts from a shift template resides in a "Generate Shifts" algorithm on the shift template's business object. Click [here](#) to see the algorithm types available for this system event.

Overridden Shift Details

In a situation where crew shift details differ from their template on a particular date, generate the shift initially from the template and then you can manually adjust it to override the template settings. This includes canceling a shift that is no longer needed. These modified shifts are called overridden shifts as they no longer match the template from which they were generated.

By default, any changes made to a shift template are automatically reapplied to all future planned but not yet closed shifts that were generated from that template, and any manual overrides will be lost. If you wish to retain overridden shifts you may indicate so on the shift template being changed.

Draft and Original Templates

If you attempt to add, delete, or modify a shift template for a shift weekly template that has been applied (or activated for common templates), the system automatically creates a draft copy of the entire weekly template. This allows you to make your changes to the draft and preview the shifts that would be generated if you accepted the draft.

You can also see if any manual edits were made to previously generated shifts. By default, such overrides are lost when the shift template that generated them is changed when a draft is accepted. You may explicitly request to retained overrides when making the shift template change.

If you have reviewed this information and are sure you want to make the proposed changes, you may accept the draft. Otherwise you may delete the draft copy to discard the proposed changes.

When a draft template is accepted, the original template is updated to match the draft

- If a shift template was changed, future shifts generated by this template are regenerated to match the changed template.
- If a shift template was deleted by the draft, the corresponding original template is also deleted if not yet used to generate shifts, otherwise it is marked inactive. Future shifts generated by this template are deleted.
- New shifts are generated for any shift templates that were added by the draft.

By default, changes made to a shift template impacts all future planned but not yet closed shifts generated by it, including overridden templates.

NOTE: If the original weekly template is a crew specific template, changes are reapplied to future shifts right away when you accept the draft. If the original template is a common template, the impact is delayed to the next time the shift generation batch process runs.

Refer to [Overridden Shift Details](#) for more information.

Once a draft has been accepted, any further changes made to the original template will result in a new draft being created.

Maintaining Crew Shifts

This portal is used to display and maintain a crew's shift.

Refer to [About Crew Shifts](#) for more information.

You can access the portal from the **Main Menu > Resource Management > Crew Shift**. You are brought to a query portal with options for searching for a specific crew shift. Once a shift has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page

- **Crew Shift**. This zone provides information about the shift, including the various maintenance actions applicable to its current state.
- **Breaks / Non Productive Tasks**. This zone defines breaks and NPT tasks for the shift.
- **Timesheets**. This zone lists timesheet records for resources allocated to the shift.

The following zones may appear as part of the portal's **Schedule** tab page

- **Crew Shift Gantt**. This zone displays a graphical representation of the tasks scheduled to this shift. The Gantt works in the same way as the dispatcher's scheduling Gantt on the CDI portal with the exception that filters and the ability to launch the geographic map are not applicable from this zone. Refer to [Using the Scheduling Gantt](#) for more information.
- **Scheduled Tasks**. This zone displays the list of tasks scheduled to this shift in their work order.
- **Un-Scheduled Assignments**. This zone displays activities that were once dispatched to the shift but were since recalled.
- **Planned Procedures**. This zone lists procedure types that the crew is planned to fill out when they would work this shift. The zone is only visible when the shift has not started yet.
- **Procedures**. This zone lists procedures completed by the crew during this shift. The zone is visible when at least one procedure was completed by the crew on that shift.

The **Route** zone on the portal's **Route** tab page displays on a geographic map the planned route of the crew's schedule vs. the actual route the crew took on a specific shift.

The **Capabilities** zone on the portal's **Capabilities** tab page displays the crew's overall capabilities from all mobile workers and vehicles currently allocated to the shift.

Recommended Crew Allocation

This portal is used to assist in allocating resources when defining a shift.

Refer to [About Crew Shifts](#) for more information.

You can access the portal from the **Maintain Crew Allocation** link on the [crew shift maintenance](#) portal when the shift has not started yet.

The following zones may appear as part of the portal's **Main** tab page:

- **Crew Shift**. This zone provides information about the shift, including the various maintenance actions applicable to its current state.
- **Recommended Mobile Worker**. This zone allows you to search and allocate appropriate mobile workers to the shift.
- **Recommended Vehicle**. This zone allows you to search and allocate appropriate vehicles to the shift.

Maintaining Breaks / Non Productive Tasks

This portal is used to display and maintain a break task or a non productive task.

Refer to [Breaks](#) and [Non-Productive Tasks \(NPTs\)](#) for more information.

You can access the portal from the Breaks / Non Productive Tasks zone on the [crew shift maintenance](#) portal.

The **Break / Non Productive Task** zone on the portal's **Main** tab page provides information about the task, including the various maintenance actions applicable to its current state.

Maintaining Shift Weekly Templates

This portal is used to display and maintain a shift weekly template. You may create a common weekly template to which multiple crews may subscribe or you may create a weekly template for a specific crew.

Refer to [About Shift Weekly Templates](#) for more information.

You can access the portal from the **Main Menu > Resource Management > Shift Weekly Template**. You are brought to a query portal with options for searching for a specific shift weekly template. Once a shift weekly template has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Shift Weekly Template.** This zone provides information about the shift weekly template, including the various maintenance actions applicable to its current state.
- **Shift Templates.** This zone defines the individual shift templates that comprise the rotation cycle for this weekly template.
- **Subscriptions.** This zone is used to maintain crew subscriptions to the common weekly template. The zone is only visible if this is a common weekly template.
- **New Subscriptions.** This zone assists in searching for and subscribing crews to the common weekly template. The zone is only visible if this is a common weekly template.
- **Inactive Shift Templates.** This zone lists shift templates that are no longer active and is visible if inactive shift templates exist for the weekly template.
- **Overridden Shifts.** This zone lists shifts that were generated by this template but were since modified manually and no longer match the template from which they were generated. The zone is visible if such shifts exist for the weekly template.

The following zones may appear as part of the portal's **Review** tab page

- **Preview Calendar.** This zone allows you to review a month's worth of potential shifts that would be generated if you applied the recent changes to the template. You may broadcast a specific shift to displays its details in the crew shift preview zone.
- **Crew Shift Preview.** This zone displays a shift broadcasted from the preview calendar zone.

Refer to the Calendar (F1-CALENDAR) zone type for more information.

Comparing Draft and Original Templates

When you make changes to a template that has already been applied, the system automatically creates a new draft version of the template. It puts your changes in the draft version and leaves the original applied version unchanged until you manually apply the draft. Before you apply a draft template, you should view the differences between the draft and the original to make sure you want all changes applied.

To view differences, click the **Show Differences** button of the Shift Weekly Template portal. The system displays a pop-up window listing each template that has been changed and the type of change (Deleted, Updated, or New).

You may also want to preview the shifts that will be generated if the draft is applied using the **Review** tab.

Navigating Between Original and Draft Templates

When the shift weekly template you are working with has an original version and a pending draft, you can navigate between them as follows:

- When viewing a draft template, click the **Original Template ID** link to navigate to the original template's maintenance portal.
- When viewing an original template, click the **Pending Draft Exists** link to navigate to the draft's maintenance portal.

Maintaining Shift Templates

This portal is used to view and maintain a shift template as part of setting up a shift weekly template.

You can access the portal from the shift templates zone on the [Maintaining Shift Weekly Templates](#) portal.

Refer to [About Shift Weekly Templates](#) for more information.

The following zones may appear as part of the portal's **Main** tab page:

- **Crew Shift Template.** This zone provides information about the shift, including the various maintenance actions applicable to its current state.
- **Overridden Shifts.** This zone shows shifts that were generated by this template but their setup has since been manually changed from the setup defined by the template.

The **Capabilities** zone on the portal's **Capabilities** tab page displays the crew's overall capabilities from all mobile workers and vehicles currently allocated to the shift based on this template.

Periods of Unavailability (POUs)

About Periods of Unavailability (POUs)

Resources may have planned times during which they are not available to perform work.

For example, mobile workers or dispatchers may take planned vacations, have personal appointments, attend meetings or training sessions, etc. Company vehicles may be unavailable for maintenance reasons. Crews may need to attend meetings or participate in training sessions and hence not be available to perform work during these events.

You may use a POU record to advise the system when resources are planned to be unavailable. Knowing upfront about these periods of unavailability allows for better planning and scheduling of work during those times.

This section describes concepts related to managing POUs.

Different Types of Events

Each POU references a [POU Type](#) which defines attributes common to all POUs of that type. For example, a status meeting type of event may only be applicable to crews while a vacation type of event is only applicable to mobile workers.

Leave POUs

You may use a POU to capture an individual's leave time such as when they report their planned vacation times or a vehicle's maintenance periods.

If a leave POU has been entered for a mobile worker or a vehicle, the system also takes into account whether or not the leave period overlaps with any of their planned shifts and factors this in when determining whether the resource is available to be scheduled for a shift.

If the leave overlaps with the shift for any amount of time, the resource is considered unavailable, even if the overlap is only for a very short duration of time. In this situation, the shift is resent to the scheduler with its capabilities adjusted to not include those possessed by the resource on leave.

Note that a shift requires at least one mobile worker for it to be properly scheduled. If all mobile workers on a shift are on leave, the shift is automatically disabled by the scheduler and no work is scheduled to it as there is no one to perform the work. If there is a change in the resources allocated to the shift or if a previously planned POU is canceled or moved to a different time, the shift would be automatically scheduled as usual.

Leave time periods are displayed on the resource's calendar so that it is easy for the resource planner to determine their availability.

Crew Meeting POUs

Crews may need to attend a meeting or a training session that takes place at a particular time during their shift hours. When they do, their schedule for the day is adjusted to take into account the time needed for them to participate in this event. In the same way, the time it would take them to travel to and from the event location is incorporated into their scheduled route.

You may use a POU record to capture the details of such event as well as the list of invited crews. This section describes concepts related to crew related POU.

Service Areas

In addition to the event time, duration, location, and the list of invited crews, the event may optionally be restricted to crews working in specific service areas.

This instructs the scheduler to incorporate the POU in the invited crews' shifts only if they are planned to work in any of the specified service areas that day. This is used to discourage the incorporation of the POU event into a crew's schedule if the crew happens to work too far from the event that day.

If no service areas are specified, there is no restriction around a crew's proximity to the event that day.

Tracking Crew Attendance

While a single POU record captures the meeting event, the system has to track the attendance of each of the invited crews individually.

The system automatically creates a POU task for every invited crew and then incorporates this task into the crew's shift schedule. The system uses the POU task type defined on the POU type record to create the POU task and if not specified a default task type is used from the global master configuration.

A POU task is dispatched to a crew, just like an activity, and its status is updated and tracked by the crew in the same way.

POU Task Workflow

Similar to other tasks associated with a location, the crew indicates they have started traveling to the event location, arrived at the location, as well as when they leave (complete) the meeting.

The following describes key stages in the overall lifecycle of a POU task.

Pending

A POU task is initially created in this state waiting to be scheduled and [queued for dispatch](#).

Queued For Dispatch

Setting the task to the **Queued For Dispatch** state is the first step in the dispatch process.

This is similar to the process of queuing breaks for dispatch. Refer to the break task [Queued For Dispatch](#) state for more information.

Dispatched

This is similar to the process of marking breaks as dispatched. Refer to the break task [Dispatched](#) state for more information.

En route

This state indicates that the crew is currently on its way to the event location. The crew provides this status update when they are about to start traveling to the location. If the crew was previously in service they are now considered out of service.

The crew provides this status update when they are about to start traveling to the location. Task status updates by the crew are similar to assignment updates. Refer to the assignment [status update](#) for more information.

Time spent while in this state is considered travel time for [crew time](#) reporting.

Started

This state indicates that the crew has arrived to the event location.

The crew provides this status update when they arrive at the location. Task status updates by the crew are similar to assignment updates. Refer to the assignment [status update](#) for more information.

Time spent while in this state is considered POU time for [crew time](#) reporting.

Completed

This state indicates that the crew has left the event.

The crew provides this status update when they leave the location (and thus complete the event). Task status updates by the crew are similar to assignment updates. Refer to the assignment [status update](#) for more information.

Canceled

This state indicates the crew is not going to attend the event and the POU is removed from the crew's schedule. A POU task may be canceled when the event itself is canceled or rescheduled to another time, or a dispatcher may have canceled the participation of the crew in the event. In either situations, if the crew is using a mobile device, the cancellation is communicated to the crew.

The crew may also explicitly decline their attendance of the event by canceling the task. Task status updates by the crew are similar to assignment updates. Refer to the assignment [status update](#) for more information.

Once a shift is finalized, tasks for events the crew did not attend are automatically canceled.

Time Zone

All date and time information on the POU portal is entered and displayed in the event location's time zone.

Creating POU

Depending on the nature of the unavailability periods, they may be recurring or one time events. You may enter a POU to record a one time event or set up a POU template to handle a recurring event.

Refer to [About POU Templates](#) for more information.

About POU Templates

An unavailability event may either be recurring or a one time event. For a recurring event, you define a POU template from which the actual POU are generated. For example, if you create a template for a regular weekly staff meeting, the system will automatically generate individual POU for each meeting.

Reoccurrence may not be applicable to all types of events. For example, an individual's vacation time does not tend to repeat in a pattern.

Whether or not a POU can be recurring is configured on the [POU Type](#). If supported, then in addition to a one time POU event, you may also create a POU template to initiate a recurring event of this type.

In addition to the event details and invitees, a template includes information about the event's recurrence pattern and expiry date that the system uses to generate the recurring POU events.

This section describes concepts related to setting up and managing POU templates.

Actual vs. Template POU

A POU template defines the details needed to generate actual POU based on that template. Any detail you can define for an actual POU, you can also predefine on a POU template.

To enforce the close pairing between a POU template and the POU it creates, both the actual and the template records are stored in the same table so they can share similar details. The same POU type defines both the business object used to create actual POU of this type as well as the business object used to create POU templates of that type.

Refer to [Defining POU Types](#) for more information.

A generated POU maintains a reference to the template that generated it, so any changes made to a template are automatically applied to all future POU generated from that template. If you modify a POU template after POU have been generated from it, the system notifies you that all previously generated POU will be deleted and regenerated, and that any manual overrides will be lost.

POU Generation

POU are generated based on the recurring pattern defined on a POU template starting from the template's effective date (or from now if in the effective date is in the past) and up to a predefined number of days known as the planning horizon, or earlier if the template expires before then.

The planning horizon is defined as a [global configuration](#) parameter and is used by the POU generation process to control the maximum number of days of POU it generates and have available for planning purposes, at any point in time.

The system supports the following patterns for a recurring POU event:

- **Daily.** The POU can recur every x days.
- **Weekly.** The POU can recur on a specific day of the week every x weeks. If you want the POU to occur on Tuesday and Thursday every week, create two POU templates: one that occurs on Tuesday every week and one that occurs on Thursday every week.

- **Monthly.** The POU can recur every x months on an exact date or a relative date (a day in a specific week). For example, the POU can occur on the 21st of every month or the last Friday every three months.
- **Yearly.** The POU can recur every x years on an exact date or a relative date (a day in a specific week in a specific month). For example, the POU can occur the 4th of July every year or the first Monday of January every two years.

If the template references a work calendar, the process skips generating POUs that fall on a non-workday or a holiday as defined by the calendar.

You may specify either an explicit expiration date or request a specific number of occurrences to be generated, after which the template would automatically expire. The number of occurrences defined on a POU template only counts actual POUs that are generated. For example, a POU template is defined as the first Monday of every month for 10 occurrences starting from the 1st of January. If two of those Mondays are holidays, the POUs will be created through December, skipping the two holiday days.

When a POU template is initially saved, the system generates POUs from the template automatically. The system only generates POUs to meet the planning horizon. The Period of Unavailability Monitor is responsible for generating more POUs going forward.

NOTE: The POU generation process relies on the POU template being periodically monitored by the **Period Of Unavailability Monitor** batch process as often as appropriate to meet your organization's needs. Refer to [M1-POU](#) batch control for more information.

On a daily basis, the background process is designed to add new POUs as needed. However, if any POU template was changed the process would automatically regenerate all future POUs that were generated by that template to match its latest version.

Refer to [Overridden POU Details](#) for more information.

NOTE: Logic used to generate POUs from a POU template resides in a "Generate POUs" algorithm on the POU template's business object. Click [here](#) to see the algorithm types available for this system event.

Overridden POU Details

In a situation where POU details differ from their template on a particular date, generate the POU initially from the template and then you can manually adjust it to override the template settings. These modified POUs are called overridden POUs, as they no longer match the template from which they were generated. The Overridden POUs zone on the POU Template portal displays a list of overridden POUs linked to the current template.

A generated POU maintains a reference to the template that generated it, so any changes made to a template are automatically applied to all future POUs that were generated from that template. If you modify a POU template after POUs have been generated from it, all previously generated POUs will be regenerated, and any manual overrides will be lost.

Maintaining Periods Of Unavailability

This portal is used to display and maintain a recurring or non recurring period of unavailability (POU). A recurring POU is also referred to as a POU template.

Refer to [About Periods of Unavailability \(POUs\)](#) for more information.

You can access the portal from the **Main Menu > Resource Management > Period Of Unavailability**. You are brought to a query portal with options for searching for a specific POU. Once a POU has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page

- **Period Of Unavailability.** This zone provides information about the POU, including the various maintenance actions applicable to its current state.

- **Invited Crews.** This zone maintains the list of invited crews for a meeting POU. The zone is only visible for a crew related POU.
- **Crew Search.** This zone allows you to search and invite crews to a meeting POU. The zone is only visible for a crew related POU.
- **Overridden Periods Of Unavailability.** This zone lists POUs that were generated by this template but were since modified manually and no longer matches the template from which they were generated. The zone is visible only for a POU template for which such POUs exist.

Maintaining Period Of Unavailability Tasks

This portal is used to display and maintain a POU task. Such task exists for every crew invited to attend a crews meeting event.

Refer to [Crew Meeting POUs](#) for more information.

You can access the portal from the Invited Crews zone on the [Maintaining Periods Of Unavailability](#) portal.

The **Period Of Unavailability Task** zone on the portal's **Main** tab page provides information about the task, including the various maintenance actions applicable to its current state.

Dispatchers

About Dispatchers

A dispatcher is a person within your organization who manages day-to-day field operations. Dispatchers are responsible for monitoring crews and activities within certain dispatch areas.

This section describes concepts related to managing dispatchers.

Managing Crews And Activities

Dispatchers are responsible for all the crews and work associated within the dispatch areas they manage. A dispatch area identifies a set of service areas and service classes representing the type of work being monitored in those service areas.

Refer to [Defining Dispatch Areas](#) for more information.

The dispatcher's record defines the dispatch areas that the dispatcher is allowed to monitor. When a dispatcher starts their shift, they may decide to monitor all or some of these areas as needed and, depending on their user privileges (also defined on their dispatcher record), they may also be able to set the range of dates to monitor.

These settings determine which crews and activities they are responsible for during their shift and they may change them as needed at any time.

Refer to [Dispatcher Shifts](#) for more information.

A crew is monitored by a dispatcher when at least one of their shifts covers a dispatch area managed by the dispatcher and falls within the dispatcher's monitoring time period. To be included in the dispatch area, the shift has to cover at least one of the service areas and allowed to perform one of the service classes defined by the dispatch area. Shifts that have not started yet need to have their planned time fall within the dispatcher's monitoring time period.

Crews that have started their shift and are still estimated to work during the dispatcher's shift, as well as those crews that have completed their shift during the dispatcher's shift are also considered irrespective of their original planned times.

NOTE: It is important to properly logoff your dispatcher shift and log back in each day to properly remove yesterday's completed shifts from the list of crews you monitor.

If your organization uses contractor management functionality, dispatchers can also monitor contracted work. Similar to crew shifts, any contractor capacity is considered within the dispatcher's responsibility when its service areas and service classes are covered by at least one of the dispatch areas currently being monitored, and it overlaps with the dispatcher's monitoring period.

Refer to [About Contractors](#) for more information.

Dispatchers monitor work within their dispatcher areas that is not yet assigned to any crew as well as all work already scheduled to crews (or contractor capacities) they monitor (even if these activities are not within their dispatch areas). Activities where the earliest time they can be scheduled falls outside the dispatcher's monitoring period are excluded.

An activity is considered to be within a dispatch area if its service area and at least one of the service classes it is associated with are covered by the dispatch area.

The dispatcher's scope of crews and activities they watch determines the data displayed on their CDI portal.

Refer to [About CDI](#) for more information.

If the dispatcher is set up to [handle alerts](#), the system may direct to them alerts related to the crews and activities they monitor, representing situations that require attention or manual intervention.

Refer to [About Alerts](#) for more information.

Dispatcher Shifts

There is no need to create a dispatcher shift in advance. The system creates a new dispatcher shift automatically when an authorized dispatcher logs on to start their shift from the CDI portal.

When logging in, the dispatcher can select which dispatch areas (of those specified in the dispatcher record) they want to monitor for the shift. They can also set the date range to monitor if they are allowed to do so based on their dispatcher record. These settings may be changed as needed throughout their shift.

The list of dispatcher areas and the period of time being monitored control the [scope](#) of crews and activities the dispatcher is responsible for during their particular shift.

While they are logged on, the dispatcher may use the CDI portal to manage their crews and activities as well as receive alerts related to these crews and activities. When the dispatcher logs off, they are officially no longer responsible for any crews or activities and alerts that may have been assigned to them are removed from their queue and reassigned to other logged on dispatchers as needed.

NOTE: For a dispatcher to be considered "logged on," they must have a logged on dispatcher shift. This is not achieved by just logging in to the application, but rather explicitly using the Logon button in the CDI portal.

Refer to [Handling Alerts](#) for more information.

A dispatcher can log off and log on again at any time, and each time they log on, the system creates a new dispatcher shift record. Each new dispatcher shift is automatically defaulted with the previous shift's settings that the dispatcher may adjust as needed.

From business perspective, a dispatcher's shift may be an 8-hour period from 9:00 to 5:00, but from a system standpoint, the shift lasts from logon to logoff. It is strongly recommended that the dispatcher logs off when they are done for the day so that alerts are properly reassigned to other logged on dispatchers and thus get addressed in a timely manner.

The dispatcher type record (and optionally also the dispatcher record) may be used to define an estimated duration for a dispatcher's shift. The system informs the dispatcher at logon time of their planned logoff time which they can adjust any time throughout their shift. If the dispatcher does not properly logoff at their estimated logoff time, the system would automatically do so some time after their planned time.

NOTE: Logic to handle dispatcher logoff is handled by a monitor algorithm defined on the logged on state of the dispatcher shift business object. Refer to this algorithm for more information.

NOTE:

NOTE: Additionally, the **Dispatcher Shift Monitor** batch process needs to run frequently to properly keep the list of crews being monitored by a dispatcher up to date by taking into account new shifts that may have been added after the dispatcher has logged on to their shift. It is also responsible for handling the logoff of dispatcher shifts that were not explicitly logged off in a timely manner. Refer to [M1-DSMTR](#) batch control for more information.

Handling Alerts

When a situation related to a crew or an activity warrants a dispatcher's attention, the system creates an alert for it and automatically assigns the alert to a logged on dispatcher to address it in a timely manner. Based on alert type configuration, these alerts may be reassigned to another logged on dispatcher if not addressed in a timely manner.

A dispatcher is considered logged on and hence available to receive alerts while their [dispatcher shift](#) is still in the logged on state. When they logoff their dispatcher shift, any remaining alerts still assigned to them are removed from their queue to be reassigned as needed.

NOTE: It is important for dispatchers to properly logoff their dispatcher shift in the CDI portal in a timely manner so as not to receive new alerts while they are not actually there to handle them.

Alerts assigned to the dispatcher and awaiting their action are listed in the alert queue zone on the dashboard. A dispatcher may indicate on their dispatcher shift that they back up another dispatcher while that person is temporarily away from their desk. When they do so the alerts assigned to the dispatcher they back up appear on their backup alert queue zone.

Dispatchers that are focused on reviewing future schedules may be set up to opt out of handling alerts on their dispatcher record. This allows them to concentrate on reviewing the next few days' schedule as opposed to handling today's exceptions.

Refer to [About Alerts](#) for more information.

KPIs

Key performance indicators (KPIs) are quantifiable measurements related to crews and activities dispatchers monitor during their dispatcher shift. These measurements are presented on the CDI portal to support dispatcher decision making and in-day exception handling.

KPIs are associated with either activities or crew shifts. For example, an activity-related KPI might indicate the number of appointments in jeopardy of being missed. A shift-related KPI might indicate the number of idle crews.

Refer to [Understanding Key Performance Indicators \(KPIs\)](#) for more information.

The dispatcher record lists the key performance indicators (KPIs) that the dispatcher should monitor and only those may appear in the CDI portal's "KPI Summary" zone.

Refer to [Monitoring KPIs](#) for more information.

To Dos In Scope

Unlike an alert that represents a situation that requires a dispatcher's immediate attention, a To Do Entry may be raised to notify a user, not necessarily a dispatcher, of a situation that typically does not call for an immediate attention. To Do Entries associated with crews and activities that the dispatcher is currently monitoring are summarized in the "To Do In Scope" dashboard zone.

About CDI

The Common Dispatching Interface (CDI) portal helps dispatchers effectively manage day-to-day operations and facilitates communication with crews in the field. The CDI supports context-based decision making for handling exceptions that arise during the day. Although the system automatically schedules and dispatches tasks to available and qualified crews, a dispatcher can override virtually any decision made by the system to handle special situations.

Most dispatcher functions are performed from the CDI portal, which provides a single point of interaction for monitoring and managing tasks, crew shifts, and KPIs. You can also view and manage alerts and To Do entries from dashboard zones without ever leaving the portal.

Refer to [About Dispatchers](#) for more information.

This section describes key aspects related to functionality supported by the portal.

CDI is for Dispatchers Only

You have to be defined in the system as a dispatcher to take advantage of the functionality supported by the CDI portal. The crews and activities you are responsible for as a dispatcher and that are managed by the portal are determined by settings provided by you when you log on to start your [dispatcher shift](#). Until you log on to your shift, no information is displayed.

Refer to [Dispatcher Shifts](#) for more information.

CDI Data

The dispatcher's scope of crews and activities they watch during their shift determines the data displayed across all zones of the CDI portal and the CDI geographic map.

Only crew shifts, contractor capacities, and activities managed by the dispatcher during their current shift may be accessed and displayed by the CDI portal. Use corresponding general search portals, outside of the CDI portal, to access entities not currently managed by the dispatcher.

In the same way, if the dispatcher handles alerts, alerts are assigned to them based on the crews and activities they currently watch.

Refer to [Managing Crews And Activities](#) for more information.

NOTE: The **Dispatcher Shift Monitor** batch process needs to run frequently to properly keep the list of crews being monitored by a dispatcher up to date by taking into account new shifts that may have been added after the dispatcher has logged on to their shift. Refer to [M1-DSMTR](#) batch control for more information.

The scheduling Gantt zone supports the ability to apply filters that narrow down the list of crews displayed in the Gantt. It is important to note that filtering is a Gantt presentation feature only and as such does not impact the list of crews in the dispatcher's scope.

NOTE: While the Gantt list of crews is filtered, all other zones, CDI map, and alert functionality still access and rely on the dispatcher's entire scope of crews and activities.

CDI Map

The CDI map is launched from the toolbar on the Scheduling Gantt zone on either the main CDI or the KPI Specific KPI portals. The map's scope of data is the same as the scope of data of the CDI portal.

NOTE: Filtering the list of crews in the Gantt does not also filter the crews visible on the map.

The map supports the display of following geographic information:

- The location of any type of task, including activities, meeting places, etc
- The current location of a crew. The location of a crew is determined by its last known or estimated location. The estimated location is used when the last actual location is either not known or considered too old as per the **GPS Threshold Minutes** [global configuration](#) parameter.
- The location of a vehicle is determined based on its last known GPS location, if any.
- The locations where crews logon and logoff their shifts.
- The location of depots.
- The boundaries of service areas, if the information is available. Refer to [Understanding Service Areas](#) for more information.
- The boundaries of active alert areas. Refer to [About Alert Areas](#) for more information.
- The boundaries of custom geographic areas. Refer to [Defining Geographic Areas](#) for more information.
- Any information provided by custom MapViewer themes, for example, a network map.

Information is presented in layers that you can turn on and off as needed. Some of these layers may be configured to be turned on and displayed when the map initially opens.

NOTE: All CDI map layers are defined using the **CDI Map Layer (M1-CDIMapLayer)** extendable lookup which also allows you to customize which layer is to be displayed by default.

In addition to layered information, the map also highlights the following on demand from another portal:

- A specific task's location of interest
- A specific crew on the map
- One or more shift routes, each in a separate color. The route(s) presented by the map is the same route used by the scheduler to create the schedule. This allows the dispatcher to see an accurate depiction of the crew's actual work route.

CDI Is Customizable

Many aspects of the CDI portal are configurable and customizable. For example, you can configure custom task color rules, custom filters, control functionality used by the Gantt, add custom CDI map layers and more.

Refer to [Customizing The CDI Portal](#) for more information.

Maintaining Dispatchers

This portal is used to display and maintain a dispatcher record.

Refer to [About Dispatchers](#) for more information.

You can access the portal from the **Main Menu > Resource Management > Dispatcher**. You are brought to a query portal with options for searching for a specific dispatcher. Once a dispatcher has been selected you are brought to the maintenance portal to view and maintain the selected record.

The **Dispatcher** zone on the portal's **Main** tab page provides information about the dispatcher.

The **Calendar** zone on the portal's **Calendar** tab page provides a calendar view of items on the dispatcher's calendar for the current month.

Refer to the Calendar (F1-CALENDAR) zone type for more information.

NOTE: Typically the CDI portal should be defined as the Home page for dispatchers.

Maintaining Dispatcher Shifts

This portal is used to display and maintain a dispatcher shift record. Typically dispatchers create and maintain their dispatcher shift records directly on the [CDI](#) portal.

Refer to [Dispatcher Shifts](#) for more information.

You can also access the portal from the **Main Menu > Resource Management > Dispatcher Shift**. You are brought to a query portal with options for searching for a specific dispatcher shift. Once a dispatcher shift has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Dispatcher Shift** zone on the portal's **Main** tab page provides information about the dispatcher, including the various maintenance actions applicable to its current state.

CDI

The CDI portal helps dispatchers effectively manage day-to-day operations and facilitates communication with crews in the field.

As a dispatcher, this portal should typically be set up as your home page. You can also access the CDI portal from the **Main Menu > Service Management > CDI**.

Refer to [About CDI](#) for more information.

The following zones may appear as part of the portal's **Main** tab page:

- **Logon.** Use this zone to logon and start your dispatcher shift. The zone is not visible once you have logged on to your shift. Refer to [Managing your Dispatcher Shift](#) for more information.
- **KPI Summary.** This zone provides a current summary of the key performance indicators (KPIs) related to crews and activities you monitor. Refer to [Monitoring KPIs](#) for more information.
- **Scheduling Gantt.** This zone displays a graphical representation of the crews you are monitoring and their scheduled tasks in a time line, allowing you to take action on those tasks. Refer to [Using the Scheduling Gantt](#) for more information.
- **Activity Search.** Use this zone to search, sort, and filter as well as perform standard dispatching functions on one or more selected activities you are responsible for during your shift. Refer to [Using the Activity List](#) for more information.

The following zones may appear as part of the **Contractor Capacity** portal tab page:

NOTE: This tab only appears if your company works with external contractor companies and you manage contracted capacities as part of your dispatcher shift responsibilities.

- **Capacity Search.** Use this zone to search, sort, and filter the contractor capacities you are responsible for during your shift.
- **Assigned Work.** This zone displays the activities assigned to a specific contractor capacity you have selected in the Capacity Search zone.

Refer to [Monitoring Contracted Work](#) for more information.

The following zones may appear as part of the **Shift** portal tab page

- **Logon.** The same zone as on the main tab is provided here for easier access.

- **Dispatcher Shift.** This zone displays your current dispatcher shift settings. Refer to [Managing your Dispatcher Shift](#) for more information.

Managing your Dispatcher Shift

The crews and activities you are responsible for as a dispatcher and that are managed by the portal are determined by settings provided by you when you log on to start your dispatcher shift, as well as the time you have started your shift.

Until you log on to your shift, no information is displayed.

To log on and start your dispatcher shift, click the **Logon** button either in the Main tab or the Shift tab. When you do so the system prompts you for your dispatcher shift settings, automatically defaulting them from your previous shift, if any.

Once logged on you can now begin monitoring your crews and work in your areas.

Use the Shift tab to view and change the settings on your dispatcher shift as well as to logoff the shift at the end of your day.

Occasionally as part of your shift, you may be responsible for backing up another dispatcher by monitoring their alerts when they step out for a while. To do so, add their user IDs to the list of users you back up on your dispatcher shift. Alerts assigned to these dispatchers would now appear on your backup alert queue.

NOTE: When done for the day, it is very important that you properly logoff your dispatcher shift so that you are no longer considered logged on and available to handle alerts.

Refer to [Dispatcher Shifts](#) for more information.

Using the Scheduling Gantt

The Scheduling Gantt zone displays a graphical representation of the crews you are monitoring and their scheduled tasks in a time line, allowing you to take action on those tasks.

The list of crews appears on the left pane and their schedules appear on the right pane. Each row in the schedule pane displays one or more shifts along with their scheduled tasks for the corresponding crew in the left pane. The schedule appears across a time line where a vertical line indicates the current date/time for reference.

This section of the user guide describes the various views you may use and actions you may take using this zone.

Gantt Views

The timeline view is the main Gantt view presented when you initially navigate to the CDI portal. The Gantt supports additional views, each providing a different perspective on the schedule.

The left and right panes, as well as the [toolbar](#), may show different content and options that are relevant to the type of view you are currently using.

The same information displayed using the main timeline view may be presented using an alternate "Plan View" where tasks in the schedule area are displayed in the order they are planned to be worked within the shift, irrespective of time. In this view the left pane lists individual shifts (as opposed to crews) and the schedule area only displays tasks, not shifts.

Refer to [Plan View](#) for more information.

You can also switch the Gantt view to display a single shift's list of scheduled tasks along with each task's available time windows in a time line. In this view, the left pane lists tasks and the schedule area portrays their schedule overlaying their time window information.

Refer to [Tasks View](#) for more information.

Crews Pane

When using the main timeline view, the left pane shows a list of crews you monitor. Shift related columns in that pane pertain to the crew's currently active or earliest planned shift in the scope of data.

NOTE: Your organization may have configured the zone to show individual crew shifts on the left pane as opposed to crews. If the left pane shows individual shifts, shift related columns pertain to the specific shift. Refer to [Configuring The CDI Portal](#) for more information.

The **Crew** column on the left pane displays the crew name as well as their current working state, i.e. whether they are out of range, en route to an activity, working on site, out of service, etc.). The columns may also include the following icons:

- If the crew is using a mobile device to work their current shift, an icon appears to indicate whether they are connected or not.
- If your company works with contractor crews, a symbol next to the crew name indicates that this is a contractor crew.

You may include additional crew related columns that you prefer to work with and organize them as you like. Refer to [Customize Your Gantt Display](#) for more information.

You can temporarily further restrict the list of crews you see by using [filters](#).

Right-click anywhere on a row in this pane to choose from the various [actions](#) you can take on the corresponding shift.

The Schedule Area Pane

When using the main timeline view, each row in the schedule pane shows one or more shifts for the corresponding crew in the left pane along with their scheduled tasks.

When multiple shifts are displayed, large gaps in time in between shifts, may be collapsed by default to provide a more concise view of the schedule. You may turn this feature on or off to your liking. Refer to [Customize Your Gantt Display](#) for more information.

Icons, colors, symbols, and other graphical elements provide visual cues to information available regarding the crew's shifts and tasks.

Refer to [Visual Cues](#) for more information on how they may be customized.

This section describes the various visual cues used in this pane.

The Shift Bar

A shift is portrayed as a larger bar on top of which smaller bars are displayed representing tasks scheduled to the shift or locked to the shift but not yet scheduled to it.

The shift bar provides multiple layers of information. Using different **colors**, it shows the shift's originally planned start and end times, its broad variable scheduling period when applicable, any scheduled overtime, as well as the actual times the crew worked that shift.

- A shift's original planned time is displayed in the "planned time" color for the width of the shift's planned start to end times.
- For a shift defined with a variable scheduling period (and has not yet become fixed in duration), the planned time is displayed in the "planned time" color whereas the variable broader period is displayed in the "variable time" color as an underlay bar to the planned time bar. As the shift's start time approaches and it becomes fixed in duration, the variable underlay bar is no longer displayed.
- Once started, the actual shift time bar is displayed on top of the originally planned time bar for reference. The actual time bar starts from the actual time the crew started their shift and ends at their planned end time. The color of the actual time

bar reflects the shift state. While still in progress, if the shift enters overtime, the remainder of the shift is drawn using the "overtime" color. When the crew completes their shift, the actual duration bar ends at the completion time.

A thick black **frame** around the shift time bar represents a closed shift.

Right-click on the shift bar area to choose from the various [actions](#) you can take on that shift.

NOTE: Since the shift bar area is overlaid with task bar images, you are likely to right click on a task bar area. In this situation the task's list of options is presented from which you should choose the "Shift Actions" option to access the shift related actions.

The Task Bar

Each task bar uses visual cues to convey information about the task.

A task that is scheduled as part of a shift's route uses a preceding [travel time line](#) representing travel time for the crew to get to the task's location from the previous task's location (or from where they start their shift).

The length of the task bar represents the task's scheduled duration, from start to end, as well as any additional time spent on site. If the crew is planned to arrive or has arrived to the location before they can start the work, that waiting time is also included in the total length of the task bar.

The **frame** of the task bar distinguishes between emergency activities, appointments, and other tasks as well as the scheduling state of the task.

- Emergencies and appointments use a thicker frame with designated colors. Emergencies also have their frame blink to attract the dispatcher's attention to the situation.
- When a task is locked to the schedule but has not yet been incorporated into the shift's route, its frame uses a dotted line as opposed to the solid line used for scheduled tasks. Such a task may be a break planned for the shift but not yet used by the scheduler, or an activity that has just been manually allocated to the shift but the scheduler has not yet processed the request.
- While waiting to be scheduled, the task is stacked at the end of the new crew's schedule, with no travel to it (since it is not yet sequenced in the route and the previous task is not yet known).
- When an activity has just been manually deferred to a future time and the scheduler has not yet processed the deferral, a dashed lined frame is used. The activity remains with this visual cue until the scheduler provides a new schedule for it.

By default the **color** of the task bar indicates its status. Refer to [Task Color Rules](#) for more information on how to add custom color rules.

Icons are used to convey the following information about the task:

- This is a break task
- This is a depot task
- These are administrative tasks such as meetings and Non Productive Tasks (NPT). Some types of meetings may be configured with a unique icon designed to make it easier for you to distinguish them on the Gantt at a glance.
- Some activity types may be configured with a unique icon designed to make it easier for you to distinguish that work on the Gantt at a glance. If the activity type is configured with such an icon then it is displayed as part of the task bar for all activities of this type.

NOTE: Showing an icon on many of the tasks on the Gantt may be overwhelming for some and useful to others. Showing icons for activities is therefore optional. You may turn this feature on or off to your preference. Refer to [Customize Your Gantt Display](#) for more information.

- A dot at the top left corner of the task bar indicates that the activity is manually allocated to this shift.
- A dot at the bottom right corner of the task bar indicates that the activity is fixed to this crew.

- This activity is part of a dependency with other activities. This means the task is either part of a job or a chain, or is a complex activity visit task.

In addition, the task type code may also be included as part of the task bar. You may turn this feature on or off to your preference. Refer to [Customize Your Gantt Display](#) for more information.

Right-click on the task bar area (or the travel line if present) to choose from the various [actions](#) you can take on that task.

Travel Time Line

Until the crew notifies the system of the exact time they went en route to their next task, the system assumes the crew starts traveling right after completing their previous task (or starting their shift) without delay. Any idle time that may be incurred because the crew is planned to arrive (or arrived) at the next location before they can actually start their next task is included as part for the next task's time on site. This idle time is also referred to as schedule idle time.

Once the system knows the exact time the crew has started their travel to the next task, the travel line is adjusted to start at that actual time. The gap between the previous task (or the beginning of shift) to the start of the travel line is considered as crew idle time.

A travel line is also present from the last task in the shift route to the crew's logoff location.

The Toolbar

Use the toolbar to view and change the information that you see in the Gantt as described in this section.

Zoom In / Zoom Out. Adjusts the task schedule area to show a narrower/broader time frame.

Previous / Next. Scrolls the Gantt view to the previous and next day within your monitoring period.

Go to Now. Scrolls the Gantt view to the current date and time.

Show Map. Launches the [CDI Map](#) (or switches focus to it if already open).

Customize Display. Use this tool to change the look and organization of your Gantt view and save your preferences for future use. Refer to [Customize Your Gantt Display](#) for more information.

Filter. Use this tool to temporarily refine the list of crew shifts that are displayed in the Gantt. Refer to [Filter The List of Crews](#) for more information.

Plan View: Toggles the Gantt view between the default timeline view and [Plan View](#), and back.

Color Rule: By default task colors are determined based on their class and status. If your organization has defined other rules to set a task's color, you may choose such rule from the list.

Time Zone: By default the Gantt is displayed in your time zone as specified on your user record. If your company operates in multiple time zones, you may change the display to use another time zone from the dropdown list.

Customize Your Gantt Display

You can add additional columns to the left pane, change some look and organization aspects of the right pane, and save your preferences for future use.

To set up a new display preference:

- Click on the **Customize Display** option in the [toolbar](#) and select the **Change Display** to specify which columns you want to see on the crews pane as well as set options to control various aspects of the task schedule pane display.
- Back on the Gantt, rearrange the columns in the left pane as per your liking by dragging a column and dropping it before or after another column.
- Click on a column header to sort the list by that column. Hold down the **Ctrl** key before you click on the next column header to further sort by additional columns.

- Resize the left pane to best fit the columns selected.
- Set the Color Rule to be different than the default set for this display preference.
- After the display is changed, you can use the Customize Display tool again to save the display for easy access later as one of your display preferences. Select **Save Display** and provide a name for your current display.

Access your listing of saved displays by selecting the **My Displays** option. You may select any display from this list to make it your current display at any time.

You may use the **Set as Default** option to mark the current display as the default view that opens whenever you open the portal.

Filter The List of Crews

Use this tool to further refine the list of crew shifts that are displayed in the Gantt according to various criteria such as by activity information, by capability, by coverage, by resource, by emergencies, or by custom filters that you save.

To set up a new filter:

- Click on the **Filters** option in the [toolbar](#) and select the type of filter you want to use.
- Provide corresponding criteria to filter by and test it before applying it into your Gantt.

Once applied, the list of crews on the left pane includes only those that have a shift you monitor that matches the filter criteria. Only one filter may be applied at any point in time and its description is displayed on the top bar along with any criteria that you may have specified.

It is important to note that filtering is a Gantt presentation feature only and as such, does not impact the list of crews in the dispatcher's scope.

NOTE: All other zones, including the CDI map, do not reflect the filtered list. In the same way, if you are set up to handle alerts, you are assigned alerts as usual and not based on the filtered list.

Click the close icon to remove filtering and resume the full view.

You can save filters that you frequently use to your list of preference filters which you can refer back to later:

- Click on the Filters option again
- Select **Save Current** to store your currently applied filter, along with the criteria you have specified, to your preference list and provide a name for the filter.

These filters can be saved to a list that you can easily select to toggle the display based on different filters and preferences that you have defined.

Access your list of saved filters by selecting the **My Filters** option from the Filters icon on the toolbar. You may apply any filter from this list at any time.

NOTE: Filters cannot be saved as your default display since it is a best practice to review all the information in your scope of crews and activities before applying filters.

Taking Actions

The Gantt allows you to perform actions on a shift as well as one or more tasks.

Shift Actions

Right-click on a row in the crews list pane or on a shift bar in the scheduler area pane to choose from the various actions you can perform on this shift. The available options may vary depending on the selected shift and your system's configuration.

The following highlights some of the more common actions you may perform for a shift:

- Actions that impact the schedule of the shift. Refer to [Crew Shift Actions](#) for more information.
- **Allocate.** This action works in conjunction with the [Activity Search](#) zone. Click the checkbox for one or more activities in the search zone, then use this action to allocate the selected activities to this shift.
- **Show Crew On Map.** Highlights the location of this crew in the open [map](#) window. This action is only applicable for crews that have started their shift and are still working. Before they start their shift and once they complete the shift, the crew's location is not known to the system.
- **Show / Hide Route On Map.** These actions draw the route for this shift on the open [map](#) window and remove it when no longer needed.
- **Tasks View.** Toggles the Gantt view to display the tasks scheduled to this shift only. Refer to [Tasks View](#) for more information.
- **View Details.** Allows you to navigate to the shift's maintenance portal to view more information about it or take further actions.
- **Send Mail.** Sends a message to the selected crew using the internal messaging system. If the crew uses a mobile device, the message is directed to their device, otherwise they may view the message when they are in the office.

Task Actions

Right-click on a task bar (as well as its travel line, if any) in the scheduler area pane to choose from the various actions you can perform on this task. The same menu is also available when you use the [Tasks View](#) and right-click on a task row in the left pane. The available options may vary depending on the type of task, its state as well as your system's configuration.

The following highlights some of the more common actions you may perform on a task:

- If this is an activity, you can perform actions that impact its current schedule. Refer to [Activity Actions](#) for more information.
- **Show On Map.** This action highlights the location of the task on the open [map](#) window.
- **View Details.** Allows you to navigate to the task's maintenance portal to view more information about it or take further actions.

Some of the actions may be taken on multiple tasks at once. To select multiple tasks, hold down the shift key and select the tasks. Right-click to choose from the applicable options.

You may also use **drag-and-drop** to unassign one or more activities from their currently assigned shift and allocate them to another shift.

Plan View

In addition to the time line view, the zone also supports a plan view that shows scheduled tasks in the order they are planned to be worked within the shift, irrespective of time.

When using this view:

- The left pane lists individual crew shifts (as opposed to crews), but other than that it works exactly like the [Crews](#) pane of the timeline view.
- The right pane shows the tasks scheduled to each shift, in their work sequence order. Each task is portrayed as a [Task Bar](#), exactly like the schedule area pane of the timeline view. There is no timeline, no shift bars, and no travel time information.

This view allows you to assess the overall schedule at a glance. As such, it may be even more useful when used in conjunction with [Task Color Rules](#) to highlight task related scheduling conditions in color.

Use the **Plan View** option on the [toolbar](#) to switch the timeline view to Plan View. Click on the icon again to toggle back to timeline view.

Tasks View

You can switch the Gantt view to display a single shift's list of scheduled tasks, along with their time window information, in a time line.

When using this view:

- The left pane lists individual tasks scheduled to the shift, but other than that, it works exactly like the [Crews](#) pane of the timeline view.
- The right pane displays time windows for each task on a time line, each type of time window in a different color. For reference, the task's schedule on the shift route is displayed as an overlay over the time windows. Each task is portrayed as a [Task Bar](#), along with a preceding [Travel Time Line](#), if any, exactly like the schedule area pane of the timeline view.

Time window information includes the activity's available time windows, as well as any override time windows that may have impacted them.

Refer to [When Can The Activity Be Actually Scheduled?](#) for more information.

This type of view allows you to easily visualize where each task is scheduled relative to its time windows, and assist you when you make scheduling decisions, monitor crew progress, and identify tasks that are in jeopardy.

To switch the Gantt to this view for a specific shift, right-click on a shift (left or right pane as applicable) and choose the **Tasks View** option. The view changes accordingly and the description of the selected shift appears in the Gantt bar to indicate you are in this view. Click the close icon to revert back to previous view.

Using the Activity Search

The Activity Search zone allows you to search for activities and take action on them. Only activities you currently manage in your shift may be accessed using this zone. Use the general activity search portal to access activities you do not currently manage.

Use the action buttons to perform standard dispatching functions on one or more selected activities.

Refer to [Activity Actions](#) for more information.

You may also allocate selected activities to a specific shift on the Gantt:

- Select the activities to allocate as usual, but do not use the Allocate action button on this zone
- Instead, locate the shift you want to allocate the activities to on the Gantt and using the right-click shift actions menu, choose the Allocate option to complete the allocation update.

Click on the **Show On Map** icon to highlight the location of the activity on the open [map](#) window.

You can also locate a specific activity on the Gantt, if presented there, using the **Show On Gantt** icon.

Using the CDI Map

The CDI Map allows you to view on a geographic map the location of the activities you manage on your shift, your crews' current location, their shift routes, nearby company vehicles, and more.

Refer to [CDI Map](#) for more information.

The content of the map and the layers available correlates to the CDI portal you currently work with, the main CDI, or the KPI specific CDI. For example, the activities and crews layers show the entire scope of activities and crews you manage on your shift, if you are currently in the main CDI portal, or those associated with the selected KPI if you currently in the KPI Specific CDI.

NOTE: Gantt filtering does not impact the scope of data presented on the map.

To **launch** the map, click on the **Show Map** icon on the Scheduling Gantt's toolbar.

The map opens in a separate window, allowing you to use the CDI portal and the CDI map side by side.

In addition to toggling on and off the various layers of information on the map itself, you can also interact with the map from the CDI portal, requesting to:

- Highlight a specific task's location on the map. Only one task may be highlighted at any time.
- Highlight a specific crew's current location on the map. Only one crew may be highlighted at any time.
- Show or hide a crew's route on the map. Each new route to display is drawn in a different color.

NOTE: You must launch the map before you can display specific crews, activities, or routes on the map. After launching the map, leave the map window open and return to the **CDI** portal to use these options.

The **Clear All** button clears all route lines, if any are displayed, and also clears the latest marker of a highlighted crew or task shown on map.

The **Refresh** button updates the map with the latest available data.

The map interface works similar to other common mapping programs. You can pan and zoom using the icons in the upper left-hand corner of the map window. You can also click and drag any point on the map to move its position within the map window. The legend icon in the lower right-hand corner of the map window can be expanded to show the icons used to represent different objects on the map.

Monitoring KPIs

Key performance indicators (KPIs) are quantifiable measurements related to crews and activities you currently monitor. They highlight certain situations or provide additional information that may assist with your decision making and exception handling.

The KPI Summary zone uses colors and icons to easily identify KPIs based on their importance. Typically, green indicates normal KPI values; red is used for values that fall outside a pre-defined range and require immediate attention; yellow is used for warnings and to indicate a potential problem.

Some of the KPIs provide information only while others may require an action to resolve them. If action is required to resolve a KPI, click on that KPI to navigate to the KPI Specific CDI portal, which provides a narrower CDI view with information necessary to resolve the issue.

Refer to [KPI Specific CDI](#) for more information.

Monitoring Contracted Work

Contractor work capacities represent an external contracting company's agreement to take on a certain amount of work of a certain type in specific service areas for a period of time. The tab is only accessible if your company uses contractor functionality and the dispatcher's data scope includes contractor capacities.

The zones in this tab allow you to search for contractor capacities you monitor in your dispatcher shift and review their state and progress. You can also display the activities assigned to a specific capacity and take actions on these activities as needed.

Refer to [About Contractors](#) for more information.

My Alert Queue

The "Alert Queue" appears on the dashboard portal when you have alerts assigned to you.

Similarly the "Backup Alert Queue" zone appears for as long as you are backing up other users and they have alerts assigned to them.

These zones are there for you irrespective of whether you are currently working with the CDI portal or not.

Refer to [Handling Alerts](#) for more information.

Managing To Do Entries

The "To Do In Scope" zone appears on the dashboard portal when any of the crews or activities you currently monitor have To Do Entries associated with them.

The zone summarizes open To Do entries sorted by their type. Click on a specific To Do Type to navigate to the [Dispatcher To Do List](#) portal to review and address issues of the selected type.

KPI Specific CDI

You navigate to this portal when you click on an actionable KPI in the general [CDI](#) "KPI Summary" zone.

The "KPI Specific CDI" portal is similar in appearance and functionality to the general CDI portal, but only displays crews and activities needed to address a particular situation represented by the selected KPI. This is useful for troubleshooting the underlying conditions that contribute to the KPI, for example, when reviewing shifts going into overtime.

In the same way, the content of the [CDI map](#) and the layers available are adjusted to match the crews and activities related to the selected KPI.

NOTE: Using this portal does not narrow down your overall scope of crews and activities you are monitoring during your shift.

The following zones may appear as part of the portal's **Main** tab page:

- **Selected KPI.** This zone provides the current measurement for selected KPI. .
- **KPI Scheduling Gantt.** This zone displays a graphical representation of the crews associated with the selected KPI and their scheduled tasks in a time line, allowing you to take action on those tasks. Tasks associated with the selected KPI are distinguished from the rest of the tasks. The zone works similarly to the scheduling Gantt zone in the general CDI portal, with the exception of filtering. The latter is not supported by this zone as the list of crews is already filtered by the selected KPI.
- Refer to [Using the Scheduling Gantt](#) for more information.
- **KPI Activity Search.** Use this zone to search, sort, and filter, as well as perform standard dispatching functions on one or more selected activities associated with the selected KPI. Refer to [Using the Activity List](#) for more information.

Dispatcher To Do List

You navigate to this portal when you click on a specific To Do Type listed in the "To Dos In Scope" dashboard zone. The portal allows you to manage To Do entries related to activities or crews you are currently monitoring on your dispatcher shift.

Refer to [To Dos In Scope](#) for more information.

The **Dispatcher To Do List** zone on the portal's **Main** tab page lists open To Do entries for you to review and address as needed.

NOTE: You have to be defined in the system as a dispatcher to take advantage of this functionality.

Alerts

About Alerts

An alert represents a situation in which a predefined condition has been met that requires immediate user intervention. The alert may be related to a specific activity, a crew on a specific shift, or a company vehicle that caused the condition. For example, your system might generate an alert if an emergency activity is not acknowledged by a crew within 5 minutes.

NOTE: Alerts are not to be confused with To Do Entries. Alerts represent an issue that requires a dispatcher's immediate attention and as such, should be kept to exceptions of that nature.

The system automatically assigns an alert to a single logged on dispatcher, based on configurable business rules, and preferably to a dispatcher that currently monitors the entity in question.

Once assigned, the alert appears on the user's [Alert Queue](#) dashboard zone, waiting for the user to acknowledge they are aware of the situation. Depending on the configuration of each type of alert, acknowledgement may be all that is needed to close the alert and if not, the alert would reappear in the queue after some period of time until the issue is resolved or it is no longer relevant.

For alerts to be addressed in a timely manner it is important that dispatchers properly logoff their dispatcher shift at the end of their work day. This ensures alerts still assigned to them are properly recycled to other dispatchers that are still working, as well as prevent the system from assigning them alerts while off duty.

Refer to [Dispatcher Shifts](#) for more information.

This section describes concepts related to managing alerts.

Alert Type Controls Everything

Each alert references an alert type that represents a unique business condition to be checked and for which an alert should be raised when the condition is met. The alert type defines the attributes and controls the behavior associated with all alerts of that type, including:

- how the alert is raised
- how important the alert is relative to other types of alerts
- how the system determines if the underlying issue has been resolved and the alert can be closed
- how long before the alert reappears in the queue once acknowledged
- how long before the alert expires or recycles, i.e. directed to another dispatcher if not acknowledged after a period of time
- and more.

Refer to [Understanding Alert Types](#) for more information.

Alert Queue

The "Alert Queue" lists open alerts currently assigned to you. The zone appears on the dashboard portal when you have at least one alert assigned to you. As a dashboard zone it is guaranteed to always be present and available to you regardless of where you navigate to in the system.

The queue is sorted by priority order as defined on the alert type, and then by time so that more important types of alerts show before others, irrespective of time. Critical alerts may be configured to make a sound to ensure they are addressed as soon as possible.

Based on alert type configuration, alerts may support specific actions you may take to manage them.

Refer to [Alert Actions](#) for more information.

You should at least acknowledge viewing the alert, even if you don't have the time to address it right away. When you acknowledge an alert, you let the system know you are aware of the problem.

For some types of alerts, acknowledgement is all that is required from you and the alert automatically closes. For example, an alert that notifies you that an appointment has been missed.

When more intervention is needed, acknowledgement removes the alert from your queue for a predefined number of minutes, providing you some time to address the underlying situation, after which the alert shows up again in your queue, waiting for you to acknowledge it again. The alert may be snoozed in this manner until the underlying problem is resolved or the alert is recycled to another user, if not acknowledged in a timely manner.

From the alert queue, you can also navigate to the portal of the related subject of the alert, for example a crew's shift or activity, to better understand the problem and take actions accordingly.

Alerts are designed to automatically close when the underlying problem that caused them can be programmatically evaluated to no longer exists or is no longer relevant. This means that for most alerts you don't need to take any specific action to close them. For those where you need to address the underlying issue, addressing it causes the alert to close.

Refer to [Alert Evaluation](#) for more information.

Backup Alert Queue

The "Backup Alert Queue" lists alerts currently assigned to another user you are backing up. The zone appears on the dashboard portal when at least one alert is assigned to any of the users you back up.

As a dashboard zone it is guaranteed to always be present and available to you regardless of where you navigate to in the system.

Refer to [Handling Alerts](#) for more information.

Alert Actions

The available actions for an alert vary depending on the alert type and its current state. Common alert actions include the following:

Acknowledge

Indicates that you have viewed the alert and are aware of the problem. The system removes the alert from the alert queue for a specified number of minutes defined on the alert type and then reissues it. If an alert is not acknowledged within the time specified in the alert type, the system recycles it to another dispatcher.

Close

Closes the alert and removes it from your queue. This action is only available if the alert type supports manual closure.

Voice

For synchronization alerts only, this indicates that you want to voice-dispatch the activity.

Refer to [Synchronization Alert](#) for more information..

Force Cancel

You may respond to an activity cancellation alert assigned to you and use this action to recall the activity from the crew shift (or contractor capacity) it is currently dispatched to, and cancel it. This action also removes the alert from your queue. This is only supported for cancel activity alerts.

NOTE: It is important that you confirm the cancellation with the assigned crew (or contractor company) to ensure the work is not going to be performed. Failing to do so may result in a discrepancy of information provided to the host vs. the actual situation in the field.

Refer to [Activity Cancellation By Host](#) for more information..

NOTE: Logic to handle the business rules associated with forcing the cancellation of an activity resides in a "Cancel Activity" algorithm on the activity's business object. Click [here](#) to see the algorithm types available for this system event.

About Alert Areas

Alert areas are geographic areas that can be displayed on a map and that trigger an alert when resources enter or leave this area. Such alerts are sent to the dispatcher monitoring the resource or the closest service area to the alert location.

This section describes concepts related to managing alert areas.

Hazard Areas

Hazard areas establish a boundary or zone that, if entered, triggers an alert. If the most recent actual location of a crew or vehicle shows that they have entered the area, the system generates an alert. The hazard area supports an optional list of crew primary functions allowing you to limit the alert to crews performing certain types of work, as necessary.

An alert is generated when crews currently working these listed functions, if any, or any vehicles that are not part of any started shift enter the area. For example, supervisors or other company workers might drive around, but are not identified in the system associated to a shift. These would trigger an alert if they enter a hazard area. If no restrictions are indicated, any crew that enters the area will cause the system to generate an alert.

Vehicle Leave Areas

Vehicle leave areas establish a boundary or zone where vehicles are prohibited from leaving. This type of alert area supports an optional list of vehicle types, or vehicle primary functions, allowing you to limit the alert to vehicles of a certain type, as necessary.

An alert is generated when a vehicle of the listed types, if any, leave this area. This is typically used to track stolen vehicles or to make sure that crews are not using vehicles outside of designated work areas. If no specific vehicle types are defined, the alert applies for vehicles of any type.

Excluded Resources

Once presented with an alert for a resource, entering or exiting an alert area, a dispatcher may decide that the resource is allowed to be where it is and hence manually close that alert. Manually closing the alert excludes the resource from getting additional alerts for that area. However, if for some reason alerts should be created again for that resource, the dispatcher can manually take that resource off of the exclusion list.

Maintaining Alerts

This portal is used to view and maintain an alert. Dispatchers manage alerts assigned to them directly from the "Alert Queue" dashboard zone.

Refer to [About Alerts](#) for more information.

You can also access this portal from the **Main Menu > Service Management > Alert**. You are brought to a query portal with options for searching for a specific alert. Once an alert has been selected you are brought to the maintenance portal to view and maintain the selected record.

The **Alert** zone on the portal's **Main** tab page provides information about the alert, including the various maintenance actions applicable to its current state.

Maintaining Alert Areas

This portal is used to view and maintain an alert area.

Refer to [About Alert Areas](#) for more information.

You can access this portal from the **Main Menu > Service Management > Alert Area**. You are brought to a query portal with options for searching for a specific alert area. Once an alert area has been selected you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page

- **Alert Area.** This zone provides information about the alert area, including the various maintenance actions applicable to its current state.
- **Excluded Resources.** This zone lists all resources that have been manually excluded from alert processing for the specific alert area. Closing an alert for a resource automatically excludes the resource from getting additional alerts for that area. However, if for some reason you decide that alerts should be raised again for that resource, click the **Remove** button to take that resource off of the exclusion list.

Use the **Geographic Area Map** zone on the portal's **Map** tab page to define the map boundaries of the alert area. When ready activate the alert area.

Procedures

About Procedures

The system can be configured to include procedural instructions or questionnaires to be completed by mobile workers prior to taking on a specific primary function when starting a shift or changing their function mid shift, using a specific vehicle, or starting work on an activity. If the procedure fails, the corresponding business action cannot commence.

Usually procedures relate to safety precautions or regulatory requirements. For example, the mobile worker may need to complete a vehicle inspection at the beginning of a shift, or review safety instructions for a hazardous task.

Procedure enforcement is applicable to the mobile application only and is embedded as part of the following process flows:

- Request to start a shift
- Request to change a started shift's primary function
- Request to change a started shift's vehicle(s)
- Every time the crew starts an activity of a specific type. If they suspended and start it later, again they would be requested to complete the procedures.

Refer to [Enforcing Procedures](#) for more information.

Procedures cannot be manually created but rather initiated as part of the supported business events. The system tracks each procedure the crew has completed for audit purposes.

All procedures completed during a crew's shift are listed on the crew shift portal. Those related to a specific activity are also listed as part of the activity portal. You can also search for any procedure using the general procedure search portal.

This section describes concepts related to using procedures.

Procedure Type Controls Everything

Each procedure references a procedure type that controls the form presented to the crew as well as the business rules involved in completing it. The base product supports a configurable questionnaire type of procedure, but you can implement more sophisticated custom procedure forms to meet your business needs.

Refer to [Understanding Procedure Types](#) for more information on how to configure the system to enforce your procedures as part of the supported business events.

Shift Related Procedures

Certain procedures must be completed before the crew can start their shift as well as change their primary function or any of their vehicles mid shift. When this occurs, the crew is notified on their mobile device that they are required to complete one or more of these procedures.

Procedure types specified on a [Service Class](#) must be completed before the crew can start a shift referencing this service class as their primary function as well as each time they request to change their started shift to assume a new function.

Procedure types specified on a [Vehicle Type](#) must be completed before the crew is allowed to use a vehicle of this type. A separate procedure is required for each vehicle of the specific type, each time such vehicle is used.

Before the shift starts, you can view on the crew shift portal all the procedures the crew would be required to complete if they start the shift with its currently planned configuration. This includes any procedure types that are defined for the crew's primary function or related to any of the vehicles allocated to them on that shift.

At shift start time, the system determines again the list of procedures the crew is required to perform based on the actual primary function they chose and the vehicles they are about to use. Based on this list, the system creates corresponding procedure records for the crew to fill out and sends them to the crew's mobile device. The crew has to complete them and they have to all pass before the crew can start their shift.

In the same way, if a crew changes their primary function or vehicle allocation mid shift, corresponding procedures are yet again sent from the server and have to be completed before the changes take effect.

It is important to note that the list of procedures is determined and created on the server as part of processing the crew's synchronous request to start or update their shift. Procedures are sent back to the device as part of the response to this request.

NOTE: Your implementation may have different rules to determine which procedures are applicable for shift events. Logic to determine shift based procedures resides in a "Determine Shift Procedures" algorithm on the shift's business object. Click [here](#) to see the algorithm types available for this system event. Determining and creating procedures has to occur on the server so as to allow these algorithms to consult other entities and apply custom rules.

On the device, procedures are presented to the crew which then completes then one after the other. Each procedure is presented and validated on the mobile device.

A failed procedure may or may not be allowed to be manually overridden as passed based on your procedure type configuration. Once all procedures are passed or overridden, the system proceeds to perform the requested change.

Activity Related Procedures

Certain procedures must be completed on site before the crew starts to perform any activity work.

These procedure types are defined on the activity type. The system creates corresponding procedure records and sends them to the crew's mobile device when the activity is dispatched.

This is also true for an activity created in the field. When the activity is sent to the server it goes through the same process to handle procedures. This could be problematic in situations where the crew is currently working offline. To handle this, the activity type can be set to explicitly state how important it is for these procedures to be enforced when there is no connection coverage. If the crew decides to work the new activity and the activity type requires safety procedure clearance, the crew is required to be connected to receive the procedures and fill them prior to starting the work. Otherwise the crew may start the activity without completing the procedures in this situation.

NOTE: Your implementation may have different rules to determine which procedures are needed before the crew can start working on an activity. Logic to determine activity based procedures resides in the "Determine Task Procedures" algorithm on the activity's business object. Click [here](#) to see the algorithm types available for this system event. Determining and creating procedures has to occur on the server so as to allow these algorithms to consult other entities and apply custom rules.

On the device, when the crew attempts to start an activity that requires some procedures to be completed, the list of procedures is presented to the crew which then completes then one after the other. Each procedure is presented and validated on the mobile device.

A failed procedure may or may not be allowed to be manually overridden as passed based on your procedure type configuration. Once all procedures are passed or overridden the system proceeds to mark the activity started.

Maintaining Procedures

This portal is used to view a completed procedure filled out by the crew using the mobile application. As procedures are typically viewed in a context of a specific shift, you most likely navigate to this portal from the "Procedures" zone on the crew shift portal.

You can also access this portal from the **Main Menu > Resource Management > Procedure**. You are brought to a query portal with options for searching for a specific procedure. Once a procedure has been selected, you are brought to the maintenance portal to view and maintain the selected record.

Refer to [About Procedures](#) for more information.

The **Procedure** zone on the portal's **Main** tab page provides information about the procedure.

Timesheets

About Timesheets

A timesheet record summarizes the time a mobile worker spent while working a shift on performing work, traveling, attending meetings, and other broad categories of time usage.

At shift completion time, the system automatically creates a timesheet record for each period of time a mobile worker was part of the crew on that shift, allowing them to review, accept, and sometimes change their reported time information.

If the crew uses the mobile application, timesheet records for each mobile worker are prepared on the server and sent to the mobile device for review and approval. Crews that do not use the mobile application to work their shift may review and approve their timesheet records in the office when they complete their shift.

NOTE: Timesheet functionality is optional. [Global configuration](#) parameters must be specified to enable it.

This section describes concepts related to managing timesheets.

Timesheet Period

A crew on a specific shift may involve multiple mobile resources that may join and leave at different times. For example, one of the workers may leave the shift in mid day while another may join the shift later. In the same way, if the crew's vehicle breaks down in the middle of their shift they may switch to another vehicle. The system keeps track exactly when mobile workers join and leave a crew's shift, as well as tracks when vehicles are utilized by the crew on the shift. These period of times are recorded in the shift's crew allocation history information.

A timesheet record summarizes time usage for an individual mobile worker for each period of time they were part of a shift. Most of this would mean one timesheet record per mobile worker per shift but the system is designed to handle changes in resources throughout the shift.

NOTE: While base functionality mainly focuses on preparing timesheet records for mobile workers, the application can be extended to also manage vehicle utilization timesheet records if needed.

Time Usage Summary

Every minute during the timesheet period of time is summarized into a corresponding time utilization category, a "bucket", such as travel, work, etc., providing a breakdown of total time spent for each bucket.

For example, every time the crew indicated they went en route to a certain task location and until they indicated to are no longer traveling, that duration of time is added to the "travel" time bucket. In the same way, time they spent working on an activity on site is added to the "work" time bucket.

Time spent traveling to and working various types of tasks can be configured to be grouped into specific custom time "buckets" as part of each task type definition. In order to interface with your timekeeping system, these buckets should correspond to how the company wants to track time. For example, you may want to group time for all meter work in one bucket (both travel and work time), collections work in another bucket, and repair / construction work in a third bucket. Alternatively, you may want to group all travel into one travel bucket and all work in one or more work buckets.

NOTE: Time "bucket" categories are defined in **Timesheet Time Type (M1-TimeSheetTimeType)** extendable lookup along with a description of how they are used.

Generating Timesheets

When this functionality is enabled, the system initiates a process to generate timesheet records for the entire crew at the end of their shift.

Information about how time was spent during an entire shift is gathered from all the state transition log records recorded for the shift and those recorded for all the tasks the crew spent time on during the shift. When the crew uses a mobile device, actual event times are needed from logs captured on the device and sent to the server throughout the day until the shift is completed.

Timesheets can only be prepared once all this information is available on the server. Typically all information is available very close to ending the shift, but there could be exceptions that could cause some delays.

The process to generate timesheets for the entire crew works as follows:

- Wait for all shift and task information to be available on the server
- Once available, create a timesheet record for each mobile worker's period of time they worked as part of the crew on the shift. At this point, timesheets are ready for review.
- If the crew uses a mobile device and has indicated at end of shift that they are interested in reviewing their timesheets on the device, timesheets are sent to the crew's mobile device for review.

NOTE: Logic to request timesheets for the entire crew resides in an algorithm when the shift is completed. The request itself is handled by the "Crew Timesheet Process" business object. The calculation of an individual mobile worker's timesheet record is managed by the "Resource Timesheet" business object.

Timesheet types define attributes common to timesheet records of the same type. For example, you would need a timesheet type for managing the request to process timesheets for the crew and another timesheet type for the individual mobile worker timesheet records.

Refer to [Understanding Timesheet Types](#) for more information.

Review and Corrections

A timesheet record is initially created in the pending state, waiting for the mobile worker to mark it as reviewed.

Calculated time frames are strictly based on recorded system events. If for some reason the system times do not accurately reflect actual events, then depending on your business practices and settings, mobile workers may or may not be allowed to adjust their own calculated times as part of the review process.

NOTE: Use the [Timesheet Correction global configuration](#) option to allow or not allow mobile workers to correct the system calculated times.

Corrections to a system generated timesheet record are limited to adjusting time "buckets" and not to correcting the timesheet period itself.

If the mobile worker does not agree with their timesheet report or cannot correct the data, they can choose not to mark it reviewed and escalate the issue to their supervisor. The mobile worker or supervisor would have to make the correction in the office after returning from the field. If the problem is with the overall timesheet period, a new timesheet record would need to be created to reflect the correct period of time.

For example, if a mobile worker joined the shift at noon but the crew leader recorded the crew allocation change at 1 PM, then the system generated timesheet would show that the worker joined at 1 PM. Even if the mobile worker is allowed to adjust their time "buckets", they are not allowed to correct the overall time of their timesheet.

Crews using a mobile device are asked as part of the shift completion process whether they would like to review their timesheets on their device or later in the office.

Refer to [Reviewing Timesheets](#) for more information on how timesheets are handled by the mobile application.

Crews not using a mobile device can complete timesheets "in office" as part of their general shift completion workflow using the timesheet zone on the shift portal. The mobile worker may click on the **Process Crew Timesheets Now** button to generate the timesheet record for each individual resource that worked this shift or wait for the next timesheet batch process to do so.

Maintaining Timesheets

This portal is used to view and maintain a timesheet record. Timesheets are typically viewed in the context of a specific shift so you most likely navigate to this portal from the "Timesheets" zone on the crew shift portal.

Refer to [About Timesheets](#) for more information.

You can also access this portal from the **Main Menu > Resource Management > Timesheet**. You are brought to a query portal with options for searching for a specific timesheet. Once a timesheet has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Timesheet** zone on the portal's **Main** tab page provides information about the timesheet, including the various maintenance actions applicable to its current state.

Mail

About Mail

The main purpose of the mail management system is to allow dispatchers to communicate with crews without leaving the application, as well as allow crews to respond to and communicate with dispatchers from their mobile device.

Messaging is context driven, providing dispatchers control over communication with crews based on their shift, availability, and whether or not they are logged on. They can choose to send messages as normal delivery, as emergency mail, or with other special handling such as setting the message to require manual acknowledgement or setting the message to expire after a certain amount of time without being acknowledged.

Any user can send mail to any other user or to a predefined mailing list. If the user is a mobile worker, the system knows to direct the mail message to their mobile device if they use the mobile application to perform their work.

Additional context driven functionality supports dispatchers and crews. Dispatchers can send a mail message to all of crews or only those they monitor. In the same way, crews can send mail to all logged on dispatchers or only those actively monitoring them. A crew can also send mail to a specific user, typically their supervisor.

Users have the option of selecting from a mail template to send preset mail content for specific situations, or they can create unique messages from scratch.

You can manage your mail using the [Mail Summary](#) dashboard zone.

This section describes concepts related to managing mail messages.

Emergency Messages

An emergency message is sent to a crew's mobile device in higher priority than regular messages. On the device, the message pops up over the current screen to grab the crew's attention. Messages can also be tracked to show whether or not they have been acknowledged.

Mail Summary

You receive notifications of new messages sent to you via the Mail Summary zone on dashboard.

From that zone you can also compose a new message as well as navigate to the My Mail portal to manage all of the mail messages.

Maintaining Mail

This portal is used to view and maintain one of your mail messages.

Refer to [About Mail](#) for more information.

You can access this portal by navigating to "My Mail" portal using the **Mail** link on the "Mail Summary" dashboard zone. You are brought to a query portal with options for searching for a specific mail message. Once a mail message has been selected, you are brought to the maintenance portal to view and maintain the selected record.

You can also access the "My Mail" portal from the **Main Menu > My Mail**.

The My Mail portal provides access to view mail you have received or sent. You can create new mail, reply, acknowledge mail you received, and archive your mail using this portal.

The **Mail** zone on the portal's **Main** tab page provides information about the mail message, including the various maintenance actions applicable to its current state.

In addition to the standard add menu option, you can start a new message in one of several ways:

- Dispatchers can right-click on a shift to send mail to the entire crew from the CDI portal.
- You can click the **New** button on the "Mail Summary" dashboard zone

Transfer of Goods

About Transfer of Goods

An activity may request to deliver goods to a certain location, for example to a customer. The reverse is also supported, a request to pick up goods from a certain location. Transporting goods from one location to another requires special scheduling and handling considerations around questions like:

- What are the dimensions of the delivered items? Large or heavy items may only fit in certain vehicles. Similarly, the system needs to make sure there is room available in the vehicle, considering other goods already on board the scheduled crew's vehicle.
- What can and cannot be shipped along with these goods? For example, items made of combustible materials should not be loaded together with matches.
- Where should the crew pick up the goods for a delivery activity? Where to unload goods for a pickup activity? If goods are delivered from or collected to a depot, what are the business hours of the depot?
- and more.

This section describes concepts related to managing activities that involve transfer of goods.

The Goods

Activities that handle goods specify information related to them such as the list of items being delivered, [load capacities](#) (weight, volume, size, etc) and restrictions relative to other goods on board. For example, an activity would indicate if hazardous load combinations should be avoided.

When applicable, the activity may indicate the following restrictions around transfers and goods on board:

- A list of other types of goods that **must not be** on board with this activity's goods.
- A list of other types of goods that **preferably should not be** on board with this activity's goods.
- No other goods must be on board, only this activity's goods.
- If a transfer is made using a pair of activities in a job, the activities may indicate that no other activities may be scheduled in the route between the pickup and the delivery, i.e. make the transfer continuous without any other stops in between.
- Set a maximum transport time until goods reach their destination. For example, this would be important for items that require refrigeration or that are otherwise perishable or volatile.

If the [transfer](#) is from or to a depot, the activity also specifies:

- At least one valid depot where goods might be distributed from (for deliveries) or collected to after pickup
- An estimated time for how long it would take for the crew to load or unload the goods at the depot.

The Transfer

A delivery activity specifies the location to where goods should be delivered, whereas a pickup activity specified the location from where goods are picked up.

The system supports the following transfer options:

- Use a pair of activities to define a transfer, one activity to pick up the goods from one place and another to deliver the goods to another place. These activities would have to be scheduled to the same crew shift in a specific order, pickup then delivery. Refer to [Job](#) for more information.
- If goods are to be loaded from a depot, the activity must specify one or more depots to choose from.
- If goods are to be collected to a depot, the activity must specify one or more depots to choose from.

As a rule, a crew's vehicle starts empty and should be empty again at the end of their shift. Goods must be delivered during the same transfer effort that picked them up. For example, assume a job with one activity to pick up an item from location A and another activity to deliver that item to location B. The total goods being transferred must be 0 at the end of the job, 1 picked up - 1 delivered = 0 on board at the end of that job.

Refer to [Depots](#) for more information about depots.

Load Capacity

Specifying the overall capacity of the load on the activity is only needed if the [maximum capacity](#) on the vehicles used to transfer the goods or [depot capacity](#) limits are to be considered.

Load capacity is not limited to any specific dimensions. The system supports up to 5 cumulative and 5 non cumulative types of capacities of your choice and applicable to your business.

Cumulative refers to dimensions like weight, volume, etc that can be added up for entire load to check against the vehicle's maximum capacity limit of this type if any. For example, if the maximum weight allowed on the vehicle is 10 tons, all of the activities' goods cannot weigh more than 10 tons.

Non cumulative may be for example length, width, etc. and for these, the maximum value of any of the goods is compared against the vehicle's maximum capacity limit, if any. If the vehicle's container size is 20 meters in length, than each of the activity's goods must be 20 meters or less in length.

NOTE: The types of load capacity limits to consider such as weight, volume, size, etc. must be configured before vehicle limits functionality can be used. Refer to [Defining Load Capacities](#) for more information.

Vehicle Maximum Capacity

For some types of goods being transported it is important to know how much of a load the transport vehicle can handle such as maximum weight, maximum volume, maximum size or any other [load capacities](#) as needed.

Maximum capacity of vehicles is defined on the vehicle type, not the crew shift. Any shift that references the vehicle type is bound by those capacity restrictions. When multiple vehicles are used on specific shift, the maximum capacity is set to the total of all vehicles' cumulative capacities, and the lowest limit for the non cumulative capacities.

However, this can be overridden on a specific shift to handle exceptions. For example, if a dispatcher is aware that a truck has low tire pressure, he might override the maximum capacity to indicate that the truck cannot carry such a heavy load.

Depots

If your business handles delivery and pickup activities where goods are loaded or unloaded at particular centers, your organization can set up depots to capture the information related to these centers, such as their physical address, the days and times the depot is open, and more.

This section describes concepts related to managing depots.

Location

A depot record defines the physical address of the center by referencing a common [location](#). The scheduler uses this information to plan a trip to the depot's location to load or unload goods as needed when scheduling delivery and pickup activities respectively.

Depots may serve distribution and collection operations. A depot where crews pick up goods for delivery is referred to as a distribution depot whereas a depot where collected goods are dropped off is called a collection depot.

NOTE: If an actual depot participates in both collection and distribution operations then two depots need to be set up for the same location.

Depot Time Windows

Access time windows define the business hours when the depot is open. Each access time window also has an associated window cost and late cost. The window cost indicates the cost of using the time window, whereas the late cost indicates the cost of being late and arriving at the depot outside of the allotted time frame.

Service time windows define the preferred business hours within the normal operational hours. Lower cost is associated with scheduling work during preferred service times.

A work calendar may be specified to define public holidays when the depot is closed.

The scheduler uses this information to calculate the actual time windows during which depot tasks to visit the specific depot should be scheduled.

Multiple depots can share this information by referring to a [depot profile](#).

The system supports exceptions to time windows when the depot is open or closed.

Refer to [Depot Time Window Overrides](#) for more information.

Depot Capacity

Maximum capacities may also be specified for an access time window to indicate total capacity that can be handled during that time window at the depot, for example maximum weight, maximum volume, or any other type or capacity as configured by your implementation.

The scheduler uses maximum capacity information to ensure that the total goods associated with all depot tasks scheduled to a depot time window does not exceed this capacity limit.

NOTE: The types of maximum capacity limits to consider such as weight, volume, size, etc. must be configured before limits can be specified for a depot time window. Refer to [Defining Types Of Load Capacity](#) for more information.

Depot Site Delay

There may be a fixed delay time that is not related to the actual load or unloading of the goods when visiting a specific depot. For example, the amount of time needed to access the depot center on each visit, regardless of specific goods being transferred. The scheduler uses this information when planning the duration of the crew's visit to the depot.

Multiple depots can share this information by referring to a [depot profile](#).

Depot Task

When scheduling a delivery activity, the scheduler plans a task for the crew to visit one of the depots mentioned on the activity and pick up the goods prior to the delivery activity's scheduled time.

In the same way, when scheduling a pick up activity to collect goods from some location the scheduler also plans a task for the crew to drop off the collected goods at one of the depots specified on the activity.

Tasks to visit depots for loading and unloading purposes are generated by the scheduler and incorporated into the crew's shift route, like any other task. As with any task they perform, crews can provide status updates related to a depot task, allowing the system to track the crew's travel, arrival, and completion of the depot visit.

The scheduler creates a depot task using the task type associated with the depot record, its [depot profile](#) if any, and otherwise if not specified, it is defaulted from the [global configuration](#) record.

For example, you might have a task type to pick up keys, one to pick up hazardous goods, and one to drop off goods. Using a designated task type allows the description of the depot task everywhere in the system it is displayed to more clearly convey the purpose of the visit to the depot.

The complete transfer operation that includes the depot related activity (delivery or pickup) and its corresponding depot task (to load or unload the goods) is referred to as a depot run.

Refer to [About Depot Runs](#) for more information about depot tasks and the entire transfer run.

Depot Profile

A depot may refer to a depot profile for shared settings of certain details.

Depot profiles can be set up to establish common settings for depot task type, site delay, access time windows, and service time windows across multiple depots. All other details are defined on each individual depot.

If a depot refers to a profile, the settings established by the profile cannot be overwritten. However, anything that is changed on the depot profile affects all depots that references the profile.

Depot Time Window Overrides

An override time window is used to redefine a depot's access or service time window as originally defined on the depot. For example, if the dispatcher knows that the depot will be open later than usual on a particular day, he can override the default time window and adjust for the later time. The override will apply to a specific day and time for the depot rather than the generalized weekly schedule defined by the default time windows.

An override can also be used to completely remove a specific time window or to adjust the maximum capacities defined on the time window, such as the maximum size or weight of distribution that the depot can handle.

The host system may send in a depot time window override. Refer to [Depot Management](#) for more information.

Crews Can Be Restricted To Specific Depots

Typically in depot based operations, crews logon and logoff from their base depots. Crews can be configured so that they are restricted to their base depots only. Restricting crews to specific depots may be necessary, for example, if the crew's vehicle is too large to visit other depots.

The scheduler is limited to schedule depot related work to shifts of crew that are allowed to visit these depots. If a crew is not limited to specific depots, it is assumed that they are allowed to visit any depot. A dispatcher can override the list of depots a crew is restricted to on a specific shift to handle exceptions that day.

Depot Breaks

Certain break types can be configured to be taken at a depot. If the shift is not scheduled for depot work, the break is scheduled as a standard break.

About Depot Runs

Depot related activities involve either picking goods from a distribution depot and delivering them to a customer location, or picking up goods from a customer location and bringing them to a collection depot.

[Depot tasks](#) represent a scheduled visit to a depot, with a task duration for loading or off-loading the goods. The scheduler automatically creates these depot tasks to handle activities as needed.

Depot runs represent the entire trip or travel to visit a depot, including the depot task and one or more depot activities. A single run always involves one depot and as such, never mixes distribution and collection operations.

Refer to [About Transfer of Goods](#) for more information.

This section describes concepts related to managing depot runs.

Depot Task Duration

If the crew is scheduled to pickup or drop off goods for multiple activities, they truly only visit the depot once. In this situation, the scheduler consolidates the individual depot tasks into a single depot task larger in duration to better represent the single depot visit. Only the scheduler is responsible for creating, deleting, and consolidating depot tasks as it optimizes the scheduler.

The duration of the depot task reflects the site delay for visiting the specific depot once, as well as the total estimated time for loading or unloading goods as specified on each of the activities in the depot run. For example, if there is a site delay of 10 minutes to arrive at the depot, pass through security, and get to a depot bay, and the depot run has 4 delivery activities that each specifies 6 minutes for loading, the estimated duration of the depot task is 34 minutes.

Depot Task Workflow

As with other tasks associated with a location, the crew indicates they have started traveling to the depot location and once there, they can start performing the depot task.

The following describes key stages in the overall lifecycle of a depot task.

Pending

A depot task is initially created in this state waiting to be scheduled and [queued for dispatch](#).

Queued For Dispatch

Setting the task to the **Queued For Dispatch** state is the first step in the dispatch process.

This is similar to the process of queuing breaks for dispatch. Refer to the break task [Queued For Dispatch](#) state for more information.

Dispatched

This is similar to the process of marking breaks as dispatched. Refer to the break task [Dispatched](#) state for more information.

En route

The crew provides this status update when they are about to start traveling to the location. Task status updates by the crew are similar to assignment updates. Refer to the assignment [status update](#) for more information. If the crew was previously out of service, they are now considered in service.

Time spent while in this state is considered travel time for [crew time](#) reporting.

Started

The crew provide this status update when start the loading or unloading work at the depot. Task status updates by the crew are similar to assignment updates. Refer to the assignment [status update](#) for more information.

Time spent while in this state is considered depot time for [crew time](#) reporting.

Completed

This state indicates that the crew has completed the work and left the depot. Task status updates by the crew are similar to assignment updates. Refer to the assignment [status update](#) for more information.

Canceled

A depot task in this state indicates it is no longer needed and should not be scheduled.

Once a shift is finalized, depot tasks that were not started are automatically canceled.

Depot Task Actions

A dispatcher or a business process may perform various actions on a depot task that may impact the task's state and schedule.

The following section describes these actions and the business flows associated with them.

Close

Closing a depot run prevents the scheduler from making further changes to the set of activities in the run.

A depot task can be closed in the following ways:

- When it is advance dispatched. Refer to [Advance Dispatching](#) for more information.
- A dispatcher can manually close the depot task to finalize its schedule and prevent more work from being assigned to the run. The dispatcher can re-open the depot task at any time.
- As part of a depot cutoff request. Refer to [About Depot Cutoff](#) for more information.

Open

Opening a depot run allows the scheduler to further optimize the run and make changes to the set of activities scheduled to it.

About Depot Cutoff

A depot cutoff event closes all distribution depot runs that are planned to depart from the depot before a specified cutoff time, and instructs the scheduler to build new runs for the depot to depart after that cutoff time. The depot cutoff automatically locks all such runs from the distribution depot.

As an example, think of depot cutoff functionality applied for grocery delivery. All deliveries for a given day must be on the truck and ready to go by 10 AM to make the "early run". To allow enough time for the picking process to prepare the trucks, all orders included in the 10 AM run should be processed no later than 5 PM on the previous day. Any orders processed after 5 PM would be pushed to later runs.

To achieve that, a depot cutoff for tomorrow at 10 AM would be processed at 5 PM today. When processed, the depot cutoff closes all depot runs planned to departure before 10 AM so that the scheduler does not add or remove activities to and from each run. At this point, new activities can only be scheduled to depot runs planned to depart after 10 AM.

Once depot runs are closed for a specific cutoff time, the host system is provided with the scheduling manifest that includes the list of these runs and the activities scheduled to each in order to start with the picking process and preparation of the trucks.

NOTE: Cutoffs only apply for distribution operations hence only to distribution depots.

This section describes concepts related to depot cutoff functionality.

Processing A Cutoff Event

A depot cutoff event specifies a cutoff time as well as a trigger time, indicating the time at which the cutoff event should be processed. If no trigger time is provided, the cutoff is processed right away.

When a cutoff is processed, all depot tasks scheduled to depart on or before the cutoff time are automatically closed thus locking their schedule. The latest cutoff time for a depot is posted on the depot record notifying the scheduler that work can only be scheduled to later depot runs.

Only one cutoff event may be processed at any point in time for a depot and each cutoff time has to be later than a previously processed cutoff.

Regardless of how the cutoff event was [initiated](#), when the cutoff event processing is complete the corresponding [scheduling manifest](#) for all the depot run it has closed may be automatically sent to the host system based on depot configuration.

NOTE: Processing of depot cutoff events in a timely manner relies on the *Depot Time Window Monitor* batch process to run frequently. Refer to [MI-DPTTW](#) batch control for more information.

Initiating Cutoff Events

Depending on the integration with the host system, cutoff events for a depot can be configured to be generated in advance from within the application or requested on demand from the host system.

Generation of cutoff events is based on a weekly schedule defined on the depot or the depot profile it uses. The schedule sets predefined cutoff times per weekday, as well as a trigger offset prior to each cutoff time to state when the event should

be processed. In addition, the schedule defines how many days in advance to generate cutoff events. A cutoff event record is generated for each cutoff time in the schedule for the cutoff horizon specified.

NOTE: Logic to generate depot cutoff events relies on the depot being periodically monitored by the **Depot Monitor** batch process. Refer to [MI-DEPOT](#) batch control for more information.

Cutoff events may also be initiated by the host system on demand using the web service. Refer to [Depot Management](#) for more information.

A user may also manually initiate a cutoff event as needed by using the **Cutoff** action on the depot portal.

Scheduling Manifest

A scheduling manifest for a depot cutoff includes information about all shifts that are associated with depot runs closed by the depot cutoff request, as well as all the activities scheduled to each depot run.

Depot configuration can indicate whether or not a scheduling manifest should be automatically sent to the host system when the processing of a cutoff event is complete.

Depending on the integration with the host system, the host system may instead request for the manifest on demand. This is typically done when the host also initiates the cutoff event, polls to check its state periodically, and when complete, requests for its corresponding manifest.

Close Entire Shift

With depot cutoff events, earlier depot runs are closed while later runs remain open. This type of operation can thus accommodate same day changes as depot runs are still open later in the day.

If your business operations needs to close and lock the schedule of all runs for a crew's shift some time prior to the shift's planned start time, you don't need to use depot cutoff events at all. Instead you should leverage the advance dispatching functionality and set up your shifts to use the "close and dispatch all" advance dispatch mode. When using this mode, the system locks down the entire schedule for the shift, including all depot runs scheduled to it, at the specified time prior to shift start.

Refer to [Advance Dispatching](#) for more information.

Maintaining Depot Tasks

This portal is used to view and maintain a depot task.

You can access a depot task from the "Scheduled Tasks" zone on the crew shift portal or a Gantt zone.

Refer to [About Depot Runs](#) for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Depot Task.** This zone provides information about the depot task, including the various maintenance actions applicable to its current state.
- **Activities In Depot Run.** This zone list all the activities scheduled as part of the depot run.
- **Assignment List.** This zone lists load tracking assignments associated with activities scheduled to the depot run.

Maintaining Depot Time Windows

This portal is used to view and maintain depot time window overrides.

You can add a new override or access an existing override depot time window from the "Override Time Windows" zone on the Depot portal.

Refer to [Depots](#) for more information.

The **Depot Time Window** zone on the portal's **Main** tab page provides information about the time window, including the various maintenance actions applicable to its current state.

Maintaining Depot Cutoffs

This portal is used to view and maintain a depot cutoff request.

You can add a new cutoff request or access an existing request from the "Cutoffs" zone on the Depot portal.

Refer to [About Depot Cutoff](#) for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Cutoff.** This zone provides information about the depot cutoff request, including the various maintenance actions applicable to its current state.
- **Closed Depot Runs.** This zone lists the depot tasks closed by the cutoff event.
- **Activities Locked To Depot Run.** This zone lists delivery activities locked to the depot tasks due to the processing of the cutoff event.
- **Remaining Depot Tasks To Cutoff.** This zone lists depot tasks that still remain to be processed by this cutoff event.

Contractor Management

About Contractors

Contractor management functionality supports having outside contractors or a supplemental workforce completing work for your organization. If your organization uses contractors to work activities, you might use contractor management functionality to assign, schedule and complete this work.

Two models for working with contractors are supported: a shift based model and a capacity based model. You might use both models since each contractor company is set up and managed on an individual basis.

This section describes concepts related to contractor management functionality in general and those specific to each model.

Shift Based Model

In this model the contractor company allocates mobile resources to work for your company. Their mobile resources, i.e. mobile workers, vehicles, crews and their shifts, are set up, configured, and scheduled as if they are internal employees. Similarly, contractor crews work their scheduled shifts just like company crews; no additional management nor approval of work is needed.

Unlike internal company resources, contractor resources may only work as part of their own company's crews or your company's crews, but never as part of another contracting company's crew. In addition, a contractor mobile worker is restricted to access designated parts of the application and view only work committed to their contracting company.

The contractor company is defined as a [resource group](#) and all of their mobile workers, vehicles and crews must reference the company's resource group as part of their resource definition record.

NOTE: Aside from mobile workers, all other user roles such as dispatchers, resource planners, etc. are assumed to be company employees. Refer to [Contractor Users](#) for more information.

Capacity Model

Instead of providing resources to work as part of your mobile workforce, in this model contractor companies agree to provide a certain capacity of work to your company. These companies are then assigned work by your company based on such agreed upon capacity.

Contracted capacities are expressed in terms of number of hours or activities of a certain type, in certain areas, for a period of time. For example, a contractor company may offer to perform 1000 activities of a certain type, in one or more areas, on a specific day or week, or any period of time.

The contractor companies receive work from your company and they are responsible for managing these activities, scheduling them to their crews, and forwarding back to your company the corresponding completion information when work is complete.

The contractor company is defined as a [resource group](#) in your system.

NOTE: While the capacity contractor company is defined within the system, their individual resources are not. Their crews are fully managed and scheduled outside of your system.

This section describes concepts related to capacity based contracting model.

Scheduling Work To Capacities

Your company establishes capacity records representing work agreements with external contractor companies. Each capacity record specifies a period of time during which the contractor company agrees to take on work of a certain type in specific areas.

Refer to [About Capacities](#) for more information.

The scheduler considers capacities as well as internal crew shifts when scheduling a particular activity based on the whether or not the activity is allowed to be worked by a contractor and other preference options. During this phase, the schedule is constantly optimized and work is moved around between crew shifts and capacities, between your company and contractor companies.

It is important to understand the difference between scheduling work to a shifts vs. capacities. When the scheduler assigns work to a specific crew on a specific shift, it has all the information needed to choose an optimal assignment in terms of the capabilities the crew possess on that shift and it can incorporate travel time from the crew's current location to the various locations on their route. On the other hand, capacity defines a "bucket" of work the contractor company may receive from your company which may be completed by an unknown number of crews anytime over the agreed upon period. As such, scheduling to a capacity cannot account for travel time nor can it consider crew capabilities.

Refer to [Activity Contracting Restrictions](#) and [Contractor Eligibility Rules](#) for more information.

Some time prior to each capacity's start time, at the notification time defined for the contractor company, the system closes the capacity and dispatches ("locks") all the activities booked to it so the scheduler does not make any further changes to their schedule. Dispatching ("locking") the activities to the capacity commits them to the contractor company. Once committed, the activity is communicated to that company.

Activities may also be manually committed to the contractor company earlier than the agreed upon notification time by allocating or dispatching the activity to a capacity for that company.

Refer to [Advance Notification](#) for more information.

Advance Notification

Contractor companies may need to be notified in advance of work your company commits to them so they can plan accordingly. Each contractor company may specify the amount of lead time prior to the beginning of the contractor's capacity period that the contractor must be notified of activities committed to them for the specific capacity. This period of time is called the notification horizon.

At the beginning of the notification horizon relative to the capacity's start time, the system [closes](#) and dispatches ("locks") the capacity's schedule; no additional activities can be added to the capacity nor removed from it.

For example, if the capacity period begins on Thursday at 8:00 AM, and the notification horizon for this contractor is set to 72 hours, the capacity is closed on Monday at 8:00 AM and the contractor is notified of all committed work booked to the capacity.

Notification horizon is specified on the contractor's record but may be overridden for an individual capacity record as needed.

NOTE: This functionality relies on the shift monitor batch process to be run at an appropriate frequency.

Refer to [Outsourcing An Activity To The Contractor](#) for more information about communicating an activity to the contractor company.

Outsourcing An Activity To The Contractor

Activities committed to a capacity based contractor company need to be communicated to that company for approval. Refer to [Committing An Activity To A Contractor Company](#) to better understand the business meaning of this commitment.

The contractor company has to approve taking on each activity committed to them or decline the work, within a specified approval amount of time also specified on the contractor's record.

If the contractor has not responded in a timely manner, the activity is considered declined by the contractor. Once declined, the activity is no longer committed to the contractor company and they are excluded from performing the specific activity.

If the contractor company has approved taking on the activity, they are responsible for performing the work. When work is complete, the contractor company must provide the necessary completion information needed to complete the activity in the system. If for some reason the contractor company cannot perform the work they agreed to take on previously, they may decline it.

Refer to [Managing Committed Work](#) to learn more about the options available for a contractor company to manage the work committed to them by your company.

Activities can also be reverted manually as necessary by a dispatcher. The company can always pull back activities and allocate them elsewhere (with the risk that the contractor has already done the work).

The system creates an assignment copy of the activity when it commits it to an external contractor company to track their approval and progress working that activity. The assignment is established at commitment time and is finalized when the activity is no longer committed to that the contractor company. This is similar to the notion of a crew's assignment to track the crew's progress working an activity.

Refer to [Contractor Assignment](#) for more information.

Managing Committed Work

Interacting with a contractor company with respect to an activity committed to them involves these steps:

- The system sends the activity information to the contractor company for approval. The system may resend the activity information to the contractor company again if key information has changed.
- The contractor company approves or declines taking on the activity.
- The contractor company sends the necessary completion information when they complete the activity.

Large contractor companies may use their own system to manage their work. The product supports a standard web service based integration between their system and yours. When integrated, all communications with the contractor company is handled by message exchanges between the systems.

If the capacity contractor company's system is integrated with your company's system, their external system definition has to be referenced on their resource group record.

Refer to [Contractor Management Integration](#) for more information.

Smaller contracting companies that do not use a system of their own may manage their work from within your company's application as follows:

- Their managers may access the [Contractor Work Management](#) portal to approve, reject, and fully manage work your company has committed to their company. When using this portal, a contractor can only see their work; they cannot see other contractor's information nor your company's information.
- Their workers may use the [contractor's mobile application](#) to perform work assigned to them by their managers.

Refer to [User Access](#) for more information about allowing contractor users to access your application.

User Access

Any contractor manager who needs to access the Contractor Work Management portal must be set up as a user within the system. Refer to [Contractor Users](#) for more information.

If contractor workers utilize the contractor mobile application to perform work given to them by their manager, they must access the application using a valid user ID. Your company may set up each of these workers with their own user ID or allow all workers from the same contractor company to share a user ID.

To assist contractors managers in distributing work amongst their workers, the Contractor Work Management portal allows them to tag work to each of their workers. A worker's tag may be their own user ID or some free form text unique identifier they may assign to them. If your company has set up each contractor worker as a user, then their tag is their unique user ID. Otherwise, if your company has chosen to provide a shared user ID to all workers from this contractor company, the tag must be a free text identifier set by the contractor manager. Free text identifiers must be unique across your company. As such, it is a good practice to have your contractor companies prefix their identifiers with the company name. For example, ABC company's tags might be "ABC John", "ABC Sally", and "ABC Paul" as opposed to just "John", "Sally", and "Paul".

NOTE: The shared user ID option is not applicable when using the contractor mobile application in "connected MCP" mode.

The option chosen by your company with respect to individual or shared use of user IDs is specified as the tag type on the contractor company's resource group record.

NOTE: Aside from contractor managers and workers, all other user roles such as dispatchers, resource planners, etc are assumed to be company employees.

Activity Contracting Restrictions

Not all work may be contracted. Each activity indicates whether it should be worked by a company crew only, any internal crew (company or shift based contractor), or any resource. If not specified, the activity is not eligible for contracting. This is defaulted from the activity type, but can be overwritten on the activity itself.

NOTE: Emergency activities are never eligible for capacity contractors.

The host may request that an activity can be assigned to specific contractors (either capacity or shift-based) by supplying a list of [Allowed Contractors](#) on the activity.

A dispatcher may intervene and set activities that are otherwise eligible for contracting as not eligible for contracting using the [Revert To Company](#) action. They can also [exclude the contractor](#) currently assigned to work an activity from performing it.

Contractor Eligibility Rules

Contractor eligibility rules define scheduling prioritization between your company and contractor companies. These rules determine how your organization wants to schedule work for a particular service area and service class.

Defining contractor eligibility rules is optional. If not specified, the scheduler automatically gives preference to internal crews, company and shift based contractors, over external capacity based contractors.

Eligibility rules specify an overall preference between company and shift-based contractors vs. external capacity contractors. The eligibility rule can also provide a ranked list of resource groups, company and contractors, to be assigned work in the specified order of preference. If multiple groups are assigned the same rank, they are equally preferred. If a resource group is not listed, it is assumed to be least preferred.

The ranked preference list is stamped onto activities that are eligible to be worked by contractors when the activity is created or rescheduled, and forwarded to the scheduler for consideration. If the system detects a change to the eligibility rules while the activity is still being scheduled, it re-stamps the preferences to reflect the change.

Eligibility preferences are used by the scheduler when scheduling the activity. Scheduling cost is associated with not honoring this preference option. For example, if your preference is capacity over internal resources, then it would be less expensive to use a capacity over an internal resource.

Activities can further limit which contractors are eligible to work the activity by specifying one or more [Allowed Contractors](#) on the activity. The scheduler may only assign the activity to a company in that list.

Committing An Activity To A Contractor Company

Regardless of the contracting model used, committing an activity to a contractor company means it becomes their responsibility. Once committed, contractor users, irrespective of their contracting model, can view and act upon activities committed to their company and their company only.

NOTE: Until an activity is committed to a contractor company, only internal company users may access it!

[Allocating](#) or [dispatching](#) an activity to a contractor crew's shift or contractor company's capacity commits it to that company. Commitment stamps the activity with the resource group identifier of the contractor company and is key to ensuring contractors access their committed work only.

An activity remains committed to a shift based contractor company as long as it remains allocated or scheduled to one or the contractor company's crew shifts. If a contractor crew could not perform an activity and had to return it, or if the activity was recalled from them for any reason, the activity is no longer committed to that contractor company and can be rescheduled to any resource.

An activity remains committed to a capacity based contractor company until they decline it or a dispatcher recalls the activity from them for any reason. If the contractor company declines the activity, they are also explicitly excluded from being scheduled that activity again.

Refer to [Managing Committed Work](#) to learn more about the options available for a capacity based contractor company to manage the work committed to them by your company.

Resource Groups

Resource groups define information about the contractor companies doing work for your organization. This information includes name, contact information, and the base service area where they work.

Your organization must also be defined as a resource group of class "company" if you need to establish scheduling preference ranks between contractor companies and your own company using [Contractor Eligibility Rules](#).

The contracting model as well as whether this is your own company is defined on the [resource group type](#).

Regardless of the contracting model used, any individual who needs to access the system as a contractor needs have a user record set up to reference the contractor company they work. Once users are set up as contractors you may use this portal to associate them with the company they work with.

Refer to [Configuring Contractor Management](#) for more information.

CDI and Contractors

Contractor crews are highlighted on the Scheduling Gantt, and the activity search supports filtering by contracting related criteria.

The Contractor Capacity tab allows a dispatcher to manage contractor capacities they monitor.

About Capacities

A capacity defines a period of time during which a [contractor company](#) has agreed to perform a certain amount of work of a certain type, in certain areas, assigned to them by your company.

For example, your company may have the following work capacity agreements with contractor companies:

- Allow contractor company A to perform up to 500 hours worth of inspection work in service areas "Center" and "North" for a specific period of time.
- Allow contractor company B to perform up to 1000 activities of a specific type in service areas "Center" and "South" for a specific period of time.

Refer to [Capacity Model](#) to better understand how work capacities are used.

This section describes concepts related to managing contractor work capacities.

Capacity Type Controls Everything

Each capacity record references a [capacity type](#) which defines attributes common to all capacities of that type, for example whether the amount of work is expressed as a number of activities or number of hours.

Creating Capacities

Capacities are assumed to be set up by your company based on agreements made with the contractor company. Similar to the notion of crew shifts, capacity records may be generated based on weekly templates or created manually as needed to reflect agreements between your company and the contractor company. Capacities for a specific contractor company may be generated by a single weekly template at any given point in time.

Refer to [About Capacity Weekly Templates](#) for more information.

Capacity Workflow

The following describes key stages in the overall lifecycle of a capacity.

Pending

A capacity is initially created in this state. While in that state it is not sent to the scheduler and is considered a draft until marked as fully planned.

Planned

A capacity in this state is no longer a draft and is available for scheduling as per its configuration.

If generated by a template, the capacity is automatically set to the planned state. A user creating a capacity online has to explicitly mark it as planned when the set up is complete and the capacity should be sent to the scheduler.

While in this state, the scheduler may assign work to the capacity and change its decisions as needed to optimize and improve on the overall schedule. When it is time to [notify](#) the contractor company of the final list of activities committed to them, the system closes the capacity.

Refer to [Closing The Capacity](#) for more information.

Expired

When the end of the capacity's period is reached, it becomes expired and reaches a final state. This has no impact on the work scheduled to the capacity nor the responsibility of the contractor to that work.

Canceled

Until it expires, a capacity can be canceled and removed from the scheduler. A user or any process can cancel a capacity as long as no work is committed to it. Once started, it can only be completed.

Activities currently scheduled to the canceled capacity will be rescheduled.

Controlling The Schedule

This section describes the various aspects of a capacity that are used by the scheduler to control the activities assigned to that capacity.

Capacity Period

A capacity is offered for a specific period of time. Only activities that can be performed anytime during that time frame may be scheduled to the capacity. Refer to [activity time windows](#) for more information.

The time zone on a capacity indicates the time zone in which the period of time is specified.

Covered Areas

Contractor companies may be associated with a list of service areas where they are eligible to work. Capacities defined for that company may only cover these areas.

Similar to a crew's shift, a capacity must cover at least one service area and the scheduler would only assign work to capacity within the areas it covers.

Covered Work

A capacity defines the type of work it covers by either specifying an activity type or a service class, which is a group of activity types. If the amount of work is expressed as a number of activities, you may only specify an activity type.

The scheduler would only assign work that belongs to these activity types up to the number of activities or hours specified by that capacity.

Depot Restrictions

If depot related activities are covered by the capacity, you may limit the work to specific depots only. If no depots are specified, work from any depot may be considered.

NOTE: Assigning depot work to a contractor capacity assumes the contractor manages the travel to and from the depot to handle the goods as part of their scheduling process. In other words, the system does not schedule depot runs to for the contractor, nor leverages any transfer of goods functionality at all.

Capacity Actions

A dispatcher or a business process may perform various actions on a contractor's capacity that may impact the capacity's state and schedule.

The following section describes these actions and the business flows associated with them.

Close

Closes the capacity and **commits** all scheduled activities to the contractor company. The scheduler is prevented from removing any of the allocated activities or adding more activities to the capacity.

Capacities can be closed in the following ways:

- When it is time to **notify** the contractor company of the work committed to them, the system closes the capacity and commits all the work in to the contractor company.
- A dispatcher can close a capacity to finalize its schedule earlier than planned.

Open

Reopens a closed capacity's schedule to allow the scheduler to assign more activities to it or remove some activities from it, as needed. A capacity may only be reopened before the notification window is reached for the contractor company.

Commit All

This action **commits** to the contractor all of the activities scheduled to the capacity that have not yet been committed to them. This action does not close the capacity.

About Capacity Weekly Templates

While capacities can always be manually created to address a specific situation, they are typically planned and generated ahead of time based on recurring templates. Templates provide a tool for quickly generating large numbers of capacities based on pre-defined parameters.

A weekly template defines a cyclical weekly pattern made up of weekday capacity templates for a specific contractor company during a specific period of time.

For example, you might define a weekly template where the contractor company agrees to take on a specific type of work in a specific area on Mondays, Tuesdays and Wednesdays from 9 AM to 5 PM in one week, and cover another type of work in another area on Thursdays and Fridays from 9 AM to 5 PM the following week. Thus, the weekly template would rotate every 2 weeks.

Each contracting company may be associated with multiple effective dated capacity weekly templates representing their agreements with your company over time, but only one template is effective at any point in time.

This section describes concepts related to setting up and maintaining capacity weekly templates.

A Weekly Template Is Made Of Capacity Templates

A capacity weekly template is made of one or more capacity templates that describe the recurring capacity patterns. The details for each capacity within a cycle are captured on a capacity template record. For example, a weekly template might use a different capacity template on Fridays than the one used on other days.

Capacity weekly templates are dedicated to a contractor company. As such, each weekday capacity template is fully defined with everything that the capacity generator needs to generate a capacity for this contractor company.

The initial date of the cyclical weekly pattern is the effective date of the template.

Actual vs. Template Capacities

A capacity template defines the details needed to generate actual capacity records based on that template. Any detail you can define for an actual capacity you can also predefine on a capacity template.

To enforce the close pairing between a capacity template and the capacities it creates, both the actual and the template records are stored in the same table so they can share similar details. The same capacity type defines both the business object used to create actual capacities of this type, as well as the business object used to create capacity templates of that type.

Refer to [Defining Capacity Types](#) for more information.

A generated capacity maintains a reference to the capacity template that was used to generate it so that changes made to a template may be reapplied to those capacities.

Refer to [Capacity Generation](#) for more information.

Defining and Applying A Template

When you create a new capacity weekly template, its status is **Pending**. No capacities are generated while the template is in this state allowing you some time to properly define the weekday capacity templates and review them as needed.

After you have defined your capacity templates, you must apply the weekly template so that actual capacities will be generated from it, based on the template's effective period. Applying the template changes its status to **Applied** and the system right away generates future capacities starting from the template's effective date (or from now if that date is in the past) and no later than the template's expiration date (if any).

NOTE: Generating capacities online is limited to a preconfigured number of days in order to avoid the apply action from taking too long and time out. The parameter is defined on the capacity generation algorithm.

Once applied, the capacity generation background process manages the generation of capacities going forward.

Refer to [Capacity Generation](#) for more information about the generation algorithm as well as the overall process.

Making Changes To An Applied Template

Extending or shortening the effective period of an applied template adjusts future capacities accordingly right away when you make the change.

If you attempt to add, delete, or modify a capacity template for an applied weekly template, the system automatically creates a draft copy of the entire weekly template it belongs to; your changes are made to this draft copy, leaving the original template unchanged. This allows you to make changes in a more controlled way.

You make changes to the draft copy and preview the capacities that would be generated should you accept the draft. If you decide to accept the draft, the original template is updated to match the draft and changes are immediately applied to already generated future capacities accordingly.

Refer to [Draft and Original Templates](#) for more information.

One Effective Weekly Template For A Contractor

A contractor company's capacity weekly templates are either effective for a specified number of weeks starting from their effective date, or they may be perpetual, which means they have an effective date but no end date.

The contractor company can have been associated with multiple weekly templates over time but their capacities may be generated by a single weekly template at any given point in time. In other words, a contractor company's weekly templates must never overlap in time.

Capacity Generation

The system generates future capacities for contractor companies based on the weekly templates your company has set up to reflect agreements made with them. Capacities are generated based on the cyclical pattern of the weekly template starting from the weekly template's effective date (or from now if in the effective date is in the past) and up to a predefined number of days known as the planning horizon, or earlier, if the template expires before then.

The planning horizon is defined as a [global configuration](#) parameter and is used by the capacity generation process to control the maximum number of days worth of capacities it generates and has available for planning purposes, at any point in time.

If the template references a work calendar, the process skips generating capacities that even partially overlap with a public holiday.

The first few capacities may be initially generated online when you apply a weekly template, but the ongoing process of generating additional capacities to meet the planning horizon is handled by the weekly template monitor batch process.

NOTE: The capacity generation process relies on the capacity weekly template being periodically monitored by the **Shift Weekly Template Monitor** batch process as often as appropriate to meet your organization's needs. Refer to [M1-SHWKT](#) batch control for more information.

On a daily basis, the background process is designed to add new capacities as needed. However, if any capacities template was changed the process would automatically regenerate future capacities that were generated by that template to match its latest version. By default, even capacities that were manually overridden are regenerated to match the latest template version, but you may explicitly indicate to retain these changes when making a capacity template change.

Refer to [Overridden Capacity Details](#) for more information.

Unlike crew shifts that represent working hours for individual resources, capacities are allowed to overlap in time because they represent working hours for multiple contractor resources.

NOTE: Logic used to generate capacities from a capacity template resides in a "Generate Shifts" algorithm on the capacity template's business object. Click [here](#) to see the algorithm types available for this system event.

Overridden Capacity Details

In a situation where capacity details should differ from their template on a particular date, generate the capacity initially from the template and then you can manually adjust it to override the template settings. This includes canceling a capacity that is no longer needed. These modified capacities are called overridden capacities, as they no longer match the template from which they were generated.

By default, any changes made to a capacity template are automatically reapplied to all future planned, but not yet closed, capacities that were generated from that template, and any manual overrides will be lost. If you wish to retain overridden capacities, you may indicate so on the capacity template being changed.

Draft and Original Templates

If you attempt to add, delete, or modify a capacity template for a capacity weekly template that has been applied, the system automatically creates a draft copy of the entire weekly template it belongs to. This allows you to make your changes to the draft and preview the capacities that would be generated if you accepted the draft.

You can also see if any manual edits were made to previously generated capacities. By default, such overrides are lost when the capacity template that generated them is changed when a draft is accepted. You may explicitly request to retained overrides when making the capacity template change.

If you have reviewed this information and are sure you want to make the proposed changes, you may accept the draft. Otherwise you may delete the draft copy to discard the proposed changes.

When a draft template is accepted, the original template is updated to match the draft:

- If a capacity template was changed, future capacities generated by this template are regenerated to match the changed template.
- If a capacity template was deleted by the draft, the corresponding original template is also deleted if not yet used to generate capacities; otherwise it is marked as inactive. Future capacities generated by this older template are deleted.
- New capacities are generated for any capacity templates that were added by the draft.

By default, changes made to a capacity template impacts all future planned, but not yet closed, capacities generated by it, including overridden templates.

NOTE: Changes are reapplied to future capacities right away when you accept the draft.

Refer to [Overridden Capacity Details](#) for more information.

Once a draft has been accepted, any further changes made to the original template will result in a new draft being created.

Maintaining Resource Groups

This portal defines a contractor or your company resource group.

Refer to [Resource Groups](#) for more information.

You can access the portal from the **Main Menu > Resource Management > Resource Group**

The following zones may appear as part of the portal's **Main** tab page

- **Resource Group.** This zone displays information about the resource group.
- **Capacity Weekly Templates.** This zone lists capacity weekly templates for the contractor. It is only visible if the resource group record is for a capacity based contractor.
- **Users.** This zone lists all contractor users that work for this contractor. It is only visible if the resource group record is for a contractor.
- **Unassigned Contractor Users.** This zone lists all contractor users that are not yet associated with any contractor company. You may associate selected users with the selected contractor company. It is only visible if the resource group record is for a contractor.

The **Calendar** zone on the portal's **Calendar** tab page provides a calendar view of the contractor's work capacities for the current month. It is only visible if the resource group record is for a capacity based contractor.

Refer to the Calendar (F1-CALENDAR) zone type for more information.

Maintaining Capacities

This portal is used to display and maintain a contractor work capacity.

Refer to [About Capacities](#) for more information.

You can access the portal from the **Main Menu > Resource Management > Capacity**. You are brought to a query portal with options for searching for a specific capacity. Once a capacity has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Capacity**. This zone provides information about the capacity, including the various maintenance actions applicable to its current state.
- **Assigned Work**. This zone lists activities assigned to the capacity.

Maintaining Capacity Weekly Templates

This portal is used to display and maintain a capacity weekly template for a contractor company.

Refer to [About Capacity Weekly Templates](#) for more information.

You can access the portal from the **Main Menu > Resource Management > Capacity Weekly Template**. You are brought to a query portal with options for searching for a specific capacity weekly template. Once a capacity weekly template has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The following zones may appear as part of the portal's **Main** tab page:

- **Capacity Weekly Template**. This zone provides information about the capacity weekly template, including the various maintenance actions applicable to its current state.
- **Capacity Templates**. This zone defines the individual shift templates that comprise the rotation cycle for this weekly template.
- **Inactive Capacity Templates**. This zone lists capacity templates that are no longer active and is visible if such capacity templates exist for the weekly template.
- **Overridden Capacities**. This zone lists capacities that were generated by this template but were since modified manually and no longer matches the template from which they were generated. The zone is visible if such capacities exist for the weekly template.

The following zones may appear as part of the portal's **Review** tab page:

- **Preview Calendar**. This zone allows you to review a month's worth of potential capacities that would be generated if you applied the recent changes to the template. You may broadcast a specific capacity to displays its details in the capacity preview zone.
- **Capacity Preview**. This zone displays a capacity broadcasted from the preview calendar zone.

Refer to the Calendar (F1-CALENDAR) zone type for more information.

Maintaining Capacity Templates

This portal is used to view and maintain a capacity template as part of setting up a capacity weekly template.

Refer to [About Capacity Weekly Templates](#) for more information.

You can access the portal from the capacity templates zone on the [Maintaining Capacity Weekly Templates](#) portal.

The following zones may appear as part of the portal's **Main** tab page:

- **Capacity Template.** This zone provides information about the capacity, including the various maintenance actions applicable to its current state.
- **Overridden Capacities.** This zone shows capacities that were generated by this template but their setup has been manually changed from the setup defined by the template.

Maintaining Contractor Eligibility

This portal is used to view and maintain scheduling prioritization between your company and contractors.

You can also access this portal from the **Main Menu > Resource Management > Contractor Eligibility**.

Refer to [Contractor Eligibility Rules](#) for more information.

The **Contractor Eligibility** zone on the portal's **Main** tab page provides information about the contractor eligibility record.

Contractor Work Management

This portal helps contractor users to manage work committed to capacities offered by their company.

As a contractor manager, this portal should typically be set up as your home page, but you can also access the portal from the **Main Menu > Service Management > Contractor Work Management**.

NOTE: If you are a company user, you need to navigate to this portal for a specific contractor company in context. Navigate to this portal from the specific contractor's resource group record using the context menu associated with the contractor identifier in the record actions section.

The following zones appear as part of the portal's **Main** tab page:

- **Capacity Search.** This zone allows you to search for capacities set up for the contractor company.
- **Committed Work.** As a contractor manager, use this zone to manage activities assigned to the selected capacity and committed to your company. Refer to [Managing Your Work](#) for more information.

Managing Your Work

NOTE: Committed work must be reviewed and either approved or declined in a timely manner. Failing to do so in a timely manner would result in the system automatically removing that work from your company and handling it as if it has been declined.

To assist you with managing your company's approved work, you may use the **Maintain Workers** link on the main zone header to set up unique tag names for your workers. You only need to do this once or adjust as needed. Once set up, use these tags to assign approved activities to specific workers. This step is necessary if your company utilizes the contractor mobile application to complete work you have approved.

If you are not using the contractor mobile application to complete the work, you may navigate to each activity to complete your company's assignment providing the necessary completion information.

Mobile Application

About The Crew's Application

Crews may use the mobile application to perform their work. It allows them to fully manage their work schedule, get more work as needed, handle exceptions, as well as seamlessly report their whereabouts.

This section provides a high level overview of the mobile application as well as describes functionality supported by it that you may want to take advantage of.

NOTE: This is not a user guide to the mobile application itself.

Overview

Before a crew starts their shift, they may preview their first few tasks for the day on their mobile device. The maximum number of tasks to preview is defined on their shift. Previewed tasks may be viewed, but no modifications are allowed. Refer to [Advance Dispatching](#) for more information.

When the crew starts their shift, one or more of their scheduled tasks are dispatched to their device as per their shift configuration and appear on the mobile application's task list. Refer to [Auto-Dispatching](#) for more information.

Once tasks have been dispatched, the crew can work disconnected if wireless communication is not available. As they work each task, they enter status updates and completion details on the mobile device. The mobile application queues up these changes and once the device is connected, sends them to the server to update the corresponding master copy. These status updates are reflected on the dispatcher's Gantt, as well as used by the scheduler to adjust the crew's remaining schedule accordingly.

New tasks and updates to existing tasks may be received throughout their shift.

When the crew ends their shift, work assignments they did not manage to complete are returned to be rescheduled. Refer to [The End Of Shift Process](#) for more information.

The Mobile Device Is Known At Start Time

The device the crew uses to work their schedule is only known once they use it to start their shift.

Once known, the device identifier is stamped on the started shift and the system directs all communications for the crew to that device. If for some reason the crew has to switch to another device, the new device is stamped on the shift and the system would discard any messages that may still arrive from the crew's previous device.

Refer to [About Remote Messages](#) for more information about the message exchange mechanism.

The system provides security measures to ensure that only one device at a time can have the shift data. Only the device stamped on the shift may have access to the shift data. The system allows any member of the crew to work on the shift as long as it this is made from this device.

If a mobile device breaks or is lost during a shift, the dispatcher can forcibly disconnect the device from the shift to allow the crew to work the shift using another device.

Refer to [Switching Devices Mid Shift](#) for more information.

Real-time Updates

The main benefit of using the mobile application is that it provides real time visibility to the crew's progress in working their schedule. Each and every significant state change is reported back to the server and gets reflected in the dispatcher's CDI portal, as well as forwarded to the scheduler for consideration.

The dispatcher and the scheduler may react to exceptions better and more efficiently when they are aware of them as soon as possible. For example, while on site working on an activity, the crew may adjust the remaining time they think is needed to complete the work if the actual work is taking significantly more or less time than planned. This information allows the scheduler or the dispatcher to adjust the crew's remaining schedule accordingly.

Sending Current Location

The mobile application may be configured to periodically obtain Global Positioning System (GPS) data from the mobile device and forward this information to the server. This information is used to identify the most recent location of the crew utilizing this device.

Knowing the precise whereabouts of crews has the following main benefits:

- Provides dispatchers with a better picture on the CDI map of where the crews are relative to the crew's route, activities, other crews, and more.
- Allows the scheduler to more accurately adjust the crew's remaining schedule based on where the crew is as opposed to where they should have been. The scheduler estimates the crew's current location based on its last known location. The scheduler's estimation is more accurate when GPS reports are available.
- If the crew enters hazardous areas, their current location is used to raise alerts to dispatchers
- In case of an inquiry, allows the dispatcher to view the actual route a crew took on a specific shift compared with their planned route
- And much more.

Sending of GPS data from the mobile device can be enabled or disabled at the shift level.

Geographic Map

If the device is configured to support mapping, a geographic map icon appears on the task list user interface as well as on task related maps. When clicked, the icon displays a geographic map showing the crew's location as well as a route to their next tasks.

You can configure custom theme layers to be displayed on the maps. For example, you might configure layers to show the location of company assets. Different types of crews can be associated with different applicable layers.

NOTE: Logic to determine the custom themes to include on the mobile geographic map resides in a "May Layers" algorithm on the crew shift's business object. Click [here](#) to see the algorithm types available for this system event.

Enforce Procedures

Crews performing a certain primary function may be required to complete a set of procedures before they are allowed to start their shift, or when they change their primary function mid shift. These procedures may be defined on the service class used for primary function purposes.

Crews utilizing certain types of vehicles may be required to complete a set of procedures before they are allowed to use the vehicle. These procedures may be defined on such vehicle types.

Crews performing certain types of work may be required to complete a set of procedures on site before they are allowed to start the work. These procedures may be defined on such activity types.

Procedure enforcement is embedded as part of the following process flows:

- Request to start a shift
- Request to change a started shift's primary function

- Request to change a started shift's vehicles
- Every time the crew starts an activity of a specific type. If they suspended and start it later again, they would be requested to complete the procedures again.

A failed procedure may or may not be allowed to be manually overridden as passed based on your procedure type configuration. Once all procedures are passed or overridden the system proceeds to perform the requested change.

When applicable a procedure form may require the crew to take attendance to record who has participated in completing the procedure.

All procedures completed by the crew are sent to the server for audit purposes.

Refer to [About Procedures](#) for more information.

Handling Attachments

In the field, mobile users can upload and download attachments as needed to support their work.

Assignments with attachments, either common or specific to this activity, show a paperclip icon next to their entry in the mobile device **Task List** to indicate that an attachment is present. The crew can download their content on demand. All attachments can be downloaded on all devices; however, if the file type is not supported on the device, it cannot be opened or viewed.

The crew may capture pictures in the field while working an activity. Each capture is uploaded to the server and linked to the activity as an attachment. This attachment is also listed on the device as part of the assignment attachments list.

Refer to [About Activity Attachments](#) for more information.

Creating Activities in the Field

Depending on settings on the activity type, crews have the ability to create new activities in the field which may or may not be related to any existing activity on their schedule. This is useful when the crew is out in the field and come across an issue. Refer to [Related Activities](#) for more information.

The crew can choose to work the activity or send it in to the server to have it assigned to any crew. They may also indicate if the work requires an appointment to be coordinated with the customer. Refer to [Appointments](#) for more information.

Switch To Offline Mode

Crews that operate in areas with poor or no connectivity coverage need to work in offline mode. In this mode, their entire work schedule for the day is dispatched to them at shift start time as they won't be able to get additional work later. In addition, their shift schedule is closed so that no further work is assigned to them.

The crew requests to switch to offline mode and waits for all their work to be dispatched to them. The system notifies the crew when the process is complete so they can start traveling. The crew can also request to be online again, as needed.

Refer to [Offline Mode](#) for more information.

Request For Help

Crews may use the mobile application to request for immediate help.

They may click on the "Panic" icon to request immediate assistance for any reason. This action raises an alert for a dispatcher to follow up with the crew and send help as needed. These alerts are immediately assigned to an available dispatcher and should be given attention right away. If the mobile device has the capability, the alert is sent with the GPS location of the crew at the time that the panic request was issued to better locate the crew on the map.

NOTE: This feature is optional. It is only available on the mobile application if an active "Panic" alert type is referenced as the **Panic Alert Type**[global configuration](#) parameter.

The crew may also set a timer for a specified amount of time such that, if they don't manage to manually turn off the timer before it expires, a "Timed Event" alert is raised for a dispatcher to follow up with the crew and send help as needed. This is used when the crew feels they may be in a dangerous or hazardous situation and would like higher visibility from the dispatcher. The alert only becomes in effect and shows up on a dispatcher's alert queue once expired. If the crew manages to close the event before it has expired the alert is closed and does not appear on any dispatcher's alert queue.

NOTE: This feature is optional. It is only available on the mobile application if an active "Timed Event" alert type is referenced as the **Timed Event Alert Type**[global configuration](#) parameter.

Sending / Receiving Mail Messages

A crew can received mail messages on their device as well as reply and compose new messages using the mobile application.

The End Of Shift Process

When the crew completes their shift the following steps are involved:

- Activity assignments the crew did not manage to complete are returned to be rescheduled. Other types of tasks that were not completed are canceled. A remote message is sent for each such task with all the information captured on the device for the task for audit and reporting purposes.
- A remote message to complete the shift on the server is sent.
- If timesheet functionality is enabled, a message to prepare timesheets for the mobile workers is sent to the server. The crew may request to review their timesheet on the device or later at the office. Refer to [Reviewing Timesheets](#) for more information.

Reviewing Timesheets

If your organization uses timesheets functionality, timesheet records are created automatically at end of shift for each period of time the resource was part of the shift.

As part of the shift completion process on the mobile device, the crew is asked whether they would like to review their timesheet records on the device.

If they chose to do so, the application waits for the timesheet records to be prepared on the server and when ready, presents them to the crew for review. Whether the crew is allowed to adjust the reported time depends on your business practices and settings.

The crew can choose at any time to not review their timesheet records on the device and wait to do that later when they are back at the office.

Refer to [About Timesheets](#) for more information.

Switching Devices Mid Shift

If the crew needs to switch mobile devices for any reason, they need to contact the dispatcher to disconnect their old device from their shift.

Once the shift is forced logoff from the old device by the dispatcher, the crew can then logon from another device and request shift details from the server. All the tasks that were previously dispatched to the previous device are dispatched to the new device. The status of the tasks reflects the latest information sent to the server before the device was switched. Data on the previous device that did not reach the server is not recoverable.

Refer to [Force Logoff](#) for more information.

About The Contractor's Application

Work assigned to a contractor's work capacity may be managed by their workers using the contractor mobile application. They use the mobile application to request a list of tasks to work on and enter the completion information needed to complete the work.

This section provides a high level overview of the contractor mobile application as well as describes functionality supported by it that you may want to take advantage of.

NOTE: This is not a user guide to the mobile application itself.

Overview

A manager from the contractor company uses the [Contractor Work Management](#) portal to review, approve, and reject activities your company has committed to their company. They then can optionally manually tag each activity they have approved to be worked by one of their workers.

Refer to [Managing Committed Work](#) for more information.

Note that these workers are not defined in your system as resources and as such the system does not schedule nor push work to them. Instead they are assigned a unique tag by their manager and use the mobile application to pull the list of activities their manager has tagged to them. These activities are pulled to the device for them to be worked on.

The worker reviews the list of activities and work each activity in no particular order.

Once tasks are on the device, the worker can work disconnected if wireless communication is not available. As they work each task, they enter the necessary completion details on the mobile device. The mobile application queues up these messages and when the device is connected, sends them to the server to update the corresponding master copy.

At the end of their day, the worker uses a menu option to clear up the data on the device.

No Dispatch Process

There is no shift to start nor end, as well as no dispatch process. The worker needs to periodically refresh their task list to reflect changes made to it by their manager.

Completion Only Updates

The system does not track the contractor individual workers nor their attempts to perform an activity. As such, interim status updates are not applicable. The only update the worker may provide with respect to an activity is completing it.

If the worker needs to decline an activity, they need to contact their manager to remove the activity from their list and then refresh their list on the device.

About Remote Messages

Remote messages are used to send data to and from the crew in the field.

For crew to server messages, remote message processing ensures against data loss by saving or 'persisting' all data sent from the device so nothing is lost when the application closes or another user logs on to a new shift. This is important because devices may be shared by multiple mobile workers -- one crew may send data and then leave for the day, and another crew may use the same device on a subsequent shift.

Messages from the crew are received continually throughout the day. If message processing fails at any time, a To Do entry is created for a user to review and resolve the problem. You can view the content of any message, try to reprocess the message as is, or edit the message content and then reprocess.

Some errors may arise due to a concurrent attempt to update the same entity by another process and may resolve on their own when the system tries to process the message again by the remote message monitor batch process.

For server to crew messages, remote message processing allows for managing messages that have been queued for a long time or have been processed with error by the crew. Callback and Error scripts may exist for these kinds of messages. Messages that may need callback or error script executions are handled by the remote message monitor batch process.

NOTE: It is important to schedule the **Remote Message Monitor** batch process to retry failed messages periodically. For example, it should run every 5 minutes or so. Refer to [MI-RMMSG](#) batch control for more information.

Maintaining Remote Messages

This portal is used to view a messages sent to the crew or sent from the crew.

Refer to [About Remote Messages](#) for more information.

You can access this portal from the **Main Menu > Service Management > Remote Message**. You are brought to a query portal with options for searching for a specific remote message. Once a remote message has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Remote Message** zone on the portal's **Main** tab page provides information about the remote message, including the various maintenance actions applicable to its current state.

To Do Processing

There are numerous events that trigger the creation of email-like messages describing work that requires human attention. For example, if a bill segment has an error, a message is created to highlight such. We refer to each message as a **To Do Entry**. In this section, we describe how to view, manage and work To Do entries.

FASTPATH: We recommend familiarizing yourself with the topics described in [The Big Picture of To Do Lists](#) to fully appreciate how To Do entries are managed by the system.

The Big Picture of To Do Processing

The topics in this section provide background information on To Do processing functionality.

To Do Role

Each To Do entry is assigned a specific To Do Role. The role defines the users who may work on the entry. The permissible roles that may be assigned to a To Do entry are defined on the entry's To Do type. Refer to [To Do Entries Reference A Role](#) to learn more about initial role assignment.

After an entry is created, its role may be changed to any role defined as valid for the entry's To Do type. Manual changes of a To Do entry's role are recorded on the [To Do entry log](#).

To Do Priority

A To Do entry is associated with a priority level representing its processing order compared to other entries. Refer to [The Priority Of A To Do Entry](#) for more information on how an entry's priority is initially assigned.

A user may manually override a To Do entry's priority at any time. Once overridden, automatic recalculation of the priority is no longer performed so as not to override the value explicitly set by the user. Manual overrides of a To Do entry's priority are recorded on the [To Do entry log](#).

Work Distribution

To Do entries may be manually assigned to users either by their supervisors or picked up by the users themselves. The system can also be configured to dynamically assign the highest priority work to users on demand whenever they are available for fixing errors.

NOTE: Assignment on demand is in a plug-in. You can customize the [next assignment](#) plug-in on the installation options to follow your organization's work distribution rules. For example, when a user asks for the next assignment, business rules may be used to only consider To Do entries the user has the proper [skills](#) to resolve, prioritizing them based age, impacted dollar amount, etc.

One of the main benefits of the assignment on demand approach is that work is allocated real-time based on most current information such as skills, priority, user availability, etc. This also means that work remains unassigned until a user is actually going to work on it making it available to any qualified user.

Manual assignments of To Do entries are possible as follows:

- Supervisors may use the [supervisor function](#) pages to review and assign To Dos to users.
- A user may use the [To Do list](#) pages to review and assign To Dos to themselves or forward any of their To Dos to someone else.
- All of the above assignment functions may also be performed using the [To Do Search](#) page.


Assignment "on demand" is possible from the [Current To Do](#) zone on the Dashboard portal. Refer to this zone to learn about all the functions it provides.

A User's Current To Do

Users have the option of drilling down on an entry from various list queries in the system. Refer to [Working On A To Do Entry](#) for more information on drilling into an entry.

Drilling down on an entry from the following To Do list queries (using the entry's message description only) also sets the entry as the user's **Current** To Do entry:

- [To Do List](#)
- [To Do Search](#)

Additionally, users may have the option to click on the Current To Do icon, , on the Current To Do dashboard zone to request their next task. The returned To Do entry automatically becomes the user's **Current** To Do.

Regardless of how a To Do entry became current its information is displayed on the Current To Do zone.

FASTPATH: Refer to [Current To Do](#) zone to learn about all the functions it provides to efficiently resolve the current To Do and other entries that may potentially be [related](#) to it.

If a To Do entry becomes current as a result of drilling down from a list query, the system supports the ability to drill down to the **Next** and **Previous** entries respective to the **Current** To Do entry in that list.

NOTE: The **Previous To Do** button (Alt+Y) and **Next To Do** button (Alt+Z) in the Current To Do zone allow you to scroll through additional To Do items in your current work list without returning to the list query transaction. These buttons set the respective To Do entry to now being current and take you to its related transaction or to a script associated with the To Do Entry's message (not to the To Do entry itself). This is yet another way of drilling down on an entry.

If a To Do entry becomes current using the Current To Do zone's next assignment functionality, that entry is not associated with any work list to scroll through. The Next and Previous To Do buttons are therefore disabled in this case.

NOTE: Current Work List. Current work list refers to the list from which the current To Do entry was selected. Each new election of a current To Do overrides the previous work list with the new list the entry was selected from. If not selected from a list then the current work list is considered empty.

Related To Do Entries

A user working on a To Do may want to know of other To Dos that may be related to the same problem. If can be worked by the same user they may get resolved faster. It seems almost impossible to programmatically determine whether two To Dos really stem from the same problem. The user should therefore be presented with a list of potentially related To Dos from which he/she can select the truly relevant ones.

A To Do entry is considered potentially related to a given To Do if they reference a common entity. For example, if a given To Do references a certain account then any To Do referencing that same account is considered related to it.

NOTE: The system considers any To Do characteristic of **Foreign Key** type to represent a reference to an "entity". In other words, if a To Do has at least one common **Foreign Key** characteristic type / value with a given to do it is considered related to it.

The To Do Search page supports a **Related To Do filter** that lists all related To Dos for a given entry. Various pages showing To Do information often provide an indication as to how many non-complete entries may be related to a given To Do entry. Clicking on the indication's hyperlink transfers the user to the To Do Search query pre-populating it to list the related entries for the specific To Do entry.

FASTPATH: Refer to the [To Do Search](#) page for more information on all the functions it supports.

Complete A To Do

After you finish working on an entry, you can mark it as **Complete**. At this point, you may want to review potentially [related](#) To Do entries and complete those that have been addressed by resolving the problem associated with the current entry.

If you cannot resolve the problem, you can forward the To Do to another user, thus assigning the To Do to another user.

FASTPATH: Refer to [How To Complete To Dos](#) for more information.

NOTE: It is possible for the system to automatically mark an entry as **Complete**. Refer to [The Lifecycle Of A To Do Entry](#) for more information.

Managing To Do Entries

This section describes the various pages available to view and work on To Do entries.

To Do Supervisor Functions

This section describes To Do transactions that aid supervisors in the monitoring and assignment of To Do entries.

Supervisor To Do Summary

This Supervisor To Do Summary page summarizes To Do entries by To Do Type. Only entries that are **Open** (i.e., not assigned to a specific user) or **Being Worked On** are summarized.

NOTE: Only To Do entries that are assigned to a role to which you belong are summarized on this page. Refer to [To Do Entries Reference A Role](#) for more information.

Use **Menu > To Do > Supervisor To Do Summary** to open this page.

Description of Page

This page summarizes all To Do entries that are **Open** or **Being Worked On**.

- **To Do Type** contains the description of the To Do type.
- **Priority** provides a total count of To Do entries of a given type by priority. For each distinct priority the description includes the priority value followed by the corresponding number of entries. If all entries are of the same priority no additional count is displayed.
- **Total** contains the total of **Open** and **Being Worked On** entries. The hyperlink navigates to the [Supervisor To Do Assignment](#) transaction, setting it to display all **Open** and **Being Worked On** entries for the given To Do Type.
- **Open** contains the number of To Do Entries whose status is **Open**. The hyperlink navigates to the [Supervisor To Do Assignment](#) transaction, setting it to display all **Open** entries for the given To Do Type.
- **Being Worked On** contains the number of To Do entries whose status is **Being Worked On**. The hyperlink navigates to the [Supervisor To Do Assignment](#) transaction, setting it to display all **Being Worked On** entries for the given To Do Type.

The "age bars" that appear to the right of the totals highlight the age of the To Do entries. The colors red, yellow and green are used to highlight the relative number of old, middle-aged and new To Do entries. The age associated with each of these colors is [user-configurable](#).

Supervisor User Summary

The Supervisor User Summary page summarizes To Do entries by the user who is assigned to work on the entries. Only entries that are **Being Worked On** are summarized.

NOTE: Only To Do entries that are assigned to a role to which you belong are summarized on this page. Refer to [To Do Entries Reference A Role](#) for more information.

Use **Menu > To Do > Supervisor User Summary** to open this page.

Description of Page

This page contains a grid that displays all assigned To Do entries by User ID and To Do type. It indicates how many To Do entries of the displayed Type are assigned to the user (Assigned this Type) and how many To Do entries in total are assigned to the user (Assigned Total).

Clicking the hyperlinks transfer you to the [Supervisor To Do Assignment](#) transaction. On this transaction, you can assign / reassign To Do entries to a specific user. The difference is that clicking the To Do Type hyperlink causes only **Open** entries to be displayed on the Assignment transaction. Clicking the count in the Assigned this Type column causes **Being Worked On** entries assigned to the displayed user to be shown.

Supervisor To Do Assignment

This page lists To Do entries of a given type and allows a supervisor to assign / reassign them to a designated user. Use **Menu > To Do > Supervisor To Do Assignment** to open this page.

Description of Page

This page contains **Open** or **Being Worked On** To Do entries of a given type. These entries can be assigned to a specified user one-by-one or several at a time. The following points describe this functionality:

- You can select a different **To Do Type** by typing in a value or pressing the search button in the upper right corner. An **Info** description of the To Do type appears on the left.
- You can change the list of entries displayed according to their status. Select the **Filter by** status to be All (blank), **Open**, or **Being Worked On**.
- For **Being Worked On** entries, you may also **Filter by User** to limit the search to a specific user assigned to the To Do entries.
- If you want to re-sort the entries by a different value, click the search button adjacent to **Extra Information**. Each row's respective sort value is displayed in the **Extra Information** column.

NOTE: Extra Information. Each To Do list's default **Extra Information** is defined on the list's [To Do Type](#). Also note that each To Do list's available **Extra Information** differs.

- Click the **Select All** button to select all entries displayed in the grid. Click the **Clear All** to remove the selections in the grid.
- Use the checkbox if you want to assign specific entries.
- Entries are further sorted by their **Priority**. Highest in priority are listed first.
- **Create Date/Time** contains the date and time the To Do entry was created by the system. Click the Create Date/Time hyperlink to:
 - View more detailed information about the entry.
 - Add comments to the To Do entry.
 - Add a log entry of type **User Details** to record information about the To Do entry.
 - Forward the To Do entry to another user.
 - Send Back a To Do entry that was forwarded to you by another user.
- Click the number of **Related** entries hyperlink to navigate to the [To Do Search](#) query to list potentially related entries to the current entry.
- **Worked By** shows the user currently assigned to the To Do entry.
- **To Do Entry Status** is the current status of the To Do entry.
- **Message** contains a description of the To Do entry. Use the message description hyperlink to drill into the page on which the entry's respective object is displayed. If the To Do type has been configured to associate a script with this To Do

entry's message number, then instead of being taken to the respective page, the associated script is launched. Refer to [Working On A To Do Entry](#) for more information.

- You can assign entries to a user:
 - Enter a user ID or search for a user using the **Assign To User** field. You can only select a user that is associated with a role that is associated with this **To Do Type**. Note that the current number of assignments is displayed adjacent.
 - Select one or more entries to assign to this user and click the **Assign** button. The entries' status becomes **Being Worked On**.

To Do List Functions

This section describes To Do pages that aid users in the assignment and working of entries in To Do lists.

To Do Summary

This page summarizes To Do entries that meet the following qualifications:

- The entry is **Open** and you belong to the entry's To Do role. Refer to [To Do Entries Reference A Role](#) for more information about roles.
- The entry is **Being Worked On** and you are the user to which it is assigned.

Use **Menu > To Do > To Do Summary** or press **Alt+X** to open this page.

FASTPATH: We recommend familiarizing yourself with the topics described in [The Big Picture of To Do Lists](#) to fully appreciate how To Do entries are managed by the system.

Description of Page

This page is dedicated to a grid that summarizes the To Do entries on which you can work. The following information is displayed:

- **To Do Type** contains the description of the To Do type.
- **Priority** provides a total count of To Do entries of a given type by priority. For each distinct priority the description includes the priority value followed by the corresponding number of entries. If all entries are of the same priority no additional count is displayed.
- **Total** contains the total of **Open** and **Assigned to You** entries. The hyperlink navigates to the [To Do List](#) transaction, setting it to display all **Open** and **Being Worked On** entries assigned to you for the given To Do Type.
- **Open** contains the number of To Do Entries whose status is **Open**. The hyperlink navigates to the [To Do List](#) transaction, setting it to display all **Open** entries for the given To Do Type.
- **Assigned to You** contains the number of To Do entries that are assigned to you whose status is **Being Worked On**. The hyperlink navigates to the [Supervisor To Do Assignment](#) transaction, setting it to display all **Being Worked On** entries for the given To Do Type.

The "age bars" that appear to the right of the totals highlight the age of the To Do entries. The colors red, yellow and green are used to highlight the relative number of old, middle-aged and new To Do entries. The age associated with each of these colors is [user-configurable](#).

To Do List

This page lists To Do entries of a given type that are either **Open** (e.g., not assigned to a specific user) or are assigned to you.

NOTE: Restricted by role. Only those entries assigned to a role to which you belong may be seen on this page. Refer to [To Do Entries Reference A Role](#) for more information.

FASTPATH: We recommend familiarizing yourself with the topics described in [The Big Picture of To Do Lists](#) to fully appreciate how To Do entries are managed by the system.

Use **Menu > To Do > To Do List** to open this page.

Description of Page

This page contains a grid that summarizes the To Do entries of a given type on which you can work.

The **Info** section contains a concatenation of the **To Do Type's** description and the user's name.

The area beneath **Info** provides you with options that control which To Do entries appear and the order in which they are displayed. The following points describe the various options:



- Use **Sort By** to define how you want the To Do entries sorted. The following options are available:
 - **Assigned Date - On or After.** This option sorts the To Do entries in chronological order (based on the date the entry was assigned to you) starting from the input **Date**. If you choose this option, the system assumes you want to start from the earliest To Do entry. If you want to start from a date other than the earliest To Do entry, enter the desired **Date** and click the adjacent search button.
 - **Assigned Date - On or Before.** This option sorts the To Do entries in reverse chronological order (based on the date the entry was assigned to you) starting from the input **Date**. If you choose this option, the system assumes you want to start from the today's date. If you want to start from a date other than today's date, enter the desired **Date** and click the adjacent search button.
 - **Create Date - On or After.** This option sorts the To Do entries in chronological order (based on the date the entry was created) starting from the input **Date**. If you choose this option, the system assumes you want to start from the earliest To Do entry. If you want to start from a date other than the earliest To Do entry, enter the desired **Date** and click the adjacent search button.
 - **Create Date - On or Before.** This option sorts the To Do entries in reverse chronological order (based on the date the entry was created) starting from the input **Date**. If you choose this option, the system assumes you want to start from the today's date. If you want to start from a date other than today's date, enter the desired **Date** and click the adjacent search button.
 - **Extra Information.** This option sorts the To Do entries by the value of the **Extra Info**.
- Use **Extra Info** to choose data related to the To Do Entry that should be used to sort the To Do entries. This information is displayed in each entry's **Extra Information** column.

NOTE: Extra Info. Each To Do list's available **Extra Info** is defined on the list's [To Do Type Sort Keys](#). Also note that each To Do list's available **Extra Info** differs based on the configuration of the To Do Type.

- Use **Filter by** to restrict the To Do entries based on their status. The following options are available:
 - **All.** This option shows all entries that are **Open** or **Being Worked On**.
 - **Being Worked On.** This option shows all entries that are **Being Worked On** (and assigned to you).
 - **Open.** This option shows all entries that are **Open**.

The grid that follows contains the To Do entries that match your criteria. The following information appears:

- Click the **Select All** button to select all entries displayed in the grid. Click the **Clear All** button to remove the selections in the grid.
- Use the checkbox if you want to mark specific entries as **Completed** or **Being Worked On** using the **Complete** or **Assign** buttons at the bottom.

- A Comments icon  appears if there are comments on the To Do item and a Log Entry icon  appears if there is a log entry of type **Forwarded**, **Sent Back** or **User Details** on the To Do item. This column appears if at least one To Do has either comments or a special log entry.
- Entries are further sorted by their **Priority**. Highest in priority are listed first.
- **Create Date/Time** contains the date and time the To Do entry was created by the system.
- Click the Create Date/Time hyperlink or either of the special icons to navigate to the [To Do entry](#) page.
- Click the number of **Related** entries hyperlink to navigate to the [To Do Search](#) query to list potentially related entries to the current entry.
- **Assigned Date/Time** contains the date and time the To Do entry was assigned to a user. This field is only populated on entries that are **Being Worked On**. If a user assigned the To Do entry, the user ID and name of the user who assigned the To Do entry are displayed beneath the Date/Time.
- **Message** contains a description of the To Do entry. Use the message description hyperlink to drill into the page on which the entry's respective object is displayed. If the To Do type has been configured to associate a script with this To Do entry's message number, then instead of being taken to the respective page, the associated script is launched. Refer to [Working On A To Do Entry](#) for more information.

NOTE: Current To Do. Drilling into an entry from this page sets it to be the user's current To Do. Refer to [A User's Current To Do](#) for more information on how to efficiently scroll through additional entries in your current work list.

CAUTION: Automatic Assignment! When you drill into an entry using the message description hyperlink from this page, the system changes the status of the entry to **Being Worked On** and assigns it to you.

- **To Do Entry Status** is the current status of the To Do entry.
- **Extra Information** contains the data for the entry that corresponds to the Extra Info chosen in the sort options above.
- **To Do ID** contains the unique identifier of the To Do entry.
- If you want to mark one or more entries as **Being Worked On**, check the desired entries and then click the **Assign** button.
- If you want to mark one or more entries as **Completed**, check the desired entries and then click the **Complete** button. Completed entries do not appear on the To Do list (but they are retained on the database for audit purposes).

To Do Entry

The topics in this section describe the pages on which a To Do entry and its associated To Do log are maintained.

To Do Entry - Add Dialog

The To Do Entry transaction features a dialog that can add new To Dos (if the system is configured with [manual To Do types](#)). This dialog appears if you open the **Menu > To Do > To Do Entry > Add** (it also appears if you click the clear button when on the To Do Entry page).

Description of Page

The **To Do Type** is the type of To Do entry that you want to create. Only To Do types that are set up for manual usage appear in the list.

The default **To Do Type Priority** is displayed adjacent. Use **Override Priority** to override the default [priority](#) assigned by the system.

Type a **Subject** for your To Do entry. Depending on how the To Do type is set up, the subject may appear as the message in the To Do entry and To Do lists.

Enter a **Comment** to provide a longer description of and more details about the To Do entry.

Send To indicates whether the new To Do will be assigned to a **Role** (the default) or **User**.

The **Role** indicates the role to which this To Do entry will be assigned. Only users of the specified role will be able to view and work on the To Do entry. The role is defaulted from the To Do type. If you want to assign the To Do to a different role, change the specified role. You can only assign the To Do entry to one of the roles specified on the To Do type.

User appears if **Send To** is set to **User**. Use this option if you want to assign the To Do to a specific user. The user must be a member of the selected role.

If the **To Do type** defines valid characteristic types, the **Characteristics** collection is visible. If any of the characteristic types are flagged as Default on the To Do type, the characteristic type with its default value (if applicable) are displayed. If applicable use the characteristics to define other information related to this new To Do entry. In Oracle Utilities Customer Care and Billing, for example, you can link the account and premise that you are attempting to start service for.

Click **OK** to add the To Do Entry.

To Do Entry - Main

You use this page to perform various actions or modify details about a To Do entry. Use **Menu > To Do > To Do Entry > Search** to open this page.

NOTE: Restricted by role. You may only change To Do entries that are assigned to you or ones that are **Open** and assigned to a role to which you belong. Refer to [To Do Entries Reference A Role](#) for more information about roles.

Description of Page

To Do contains a concatenation of important information about the To Do entry. **To Do ID** is the system-assigned unique identifier of the entry. These values only appear after the case is added to the database.

NOTE: Formatting may be performed by a plug-in. Refer to [To Do Info May Be Formatted By An Algorithm](#) for more information.

To Do Type displays the entry's To Do Type and its description. Refer to [To Do Entries Reference A To Do Type](#) for more information about To Do types.

Role displays the role of the users who may be assigned to the entry along with a description of the role. Refer to [To Do Entries Reference A Role](#) for more information about roles.

Message is a display-only field that describes why the entry exists. If the To Do entry is in the status of **Open** or **Being Worked On**, the message description allows you to drill into the page on which the entry's respective object is displayed. If the To Do type has been configured to associate a script with this To Do entry's message number, then instead of being taken to the respective page, the associated script is launched. Refer to [Working On A To Do Entry](#) for more information.

Status indicates if the entry is **Open**, **Being Worked On**, or **Complete**. Refer to [The Lifecycle Of A To Do Entry](#) for more information status.

Use **Priority** to manually override the To Do entry's current **priority**. Any manual change to the priority is recorded on the [To Do entry log](#).

Assigned To displays the user who is working the entry.

Sent By displays the user who assigned this To Do to the current user.

Created By only appears if a background process that uses To Do entries to report object-specific errors created the To Do entry. The field displays the background process description and batch run number in which the entry was created. Refer to [To Dos Created for Object-Specific Error Conditions](#) for more information.

Extract Info only appears if the To Do is being sent out of the system. Extract Info describes the status of the extract (for example whether it has been processed or not). Refer to [To Do Entries May Be Routed Out Of The System](#) for more information.

Use **Comments** to describe anything interesting or unusual about the To Do entry. The **Comments** field is disabled if the To Do entry is complete.

If the To Do entry has non-complete [related entries](#) a summary description appears below the comments line. The description states the total number of related entries, how many are still **Open**, how many are assigned to you and how many assigned to others. Use this hyperlink to navigate to the [To Do Search](#) query to list these entries.

The **Actions** buttons allow you to **Complete** the To Do, **Forward** the To Do to another user or Role, or **Send Back** a To Do that has been forwarded to you by another user. All buttons are disabled once a To Do is complete. Refer to [To Do Entry Actions](#) and the [How To](#) section for more information on using these actions.

The **Characteristics** collection is visible for **manual** To Do entries or for **automatic** To Do entries that have characteristics. **Characteristic Type** indicates the type of characteristic. Specify a valid **Characteristic Value** for the characteristic type.

NOTE: Only characteristic types defined as permissible on a To Do entry may be linked to a To Do. Refer to [Setting Up Characteristic Types & Their Values](#) for more information about characteristic types. In addition, for **manual** To Do types, the characteristic type must also be defined as valid for the To Do entry's [To Do type](#).

The **Log** grid displays log entries (in reverse chronological order) that audit the progress of a To Do entry. You cannot modify or delete log entries that have been created by the system. You can manually add a log entry of type **User Details** and you may remove a log entry before it is saved. Once a **User Details** log entry is saved, you cannot modify or delete it.

The Log grid displays the following information:

Log Date/Time displays the date and time when the log entry was created.

Log Type indicates the To Do entry event for which the log entry was created. For example, when the To Do entry is created, the Log Type indicates **Created**. For more information refer to the [Log Entry Events](#) table below.

The **Details** column contains user-specified or system-generated information about the log entry. This field is blank for some types of log entries.

Log User displays the user who caused the log entry to be created.

Assigned To displays the user to whom the To Do entry was assigned (if any).

To Do Entry - Keys/Values

The Keys / Values page contains detailed information about the sort keys and drill keys associated with a To Do entry. This information may not be modified. It is shown to assist support personnel.

Use **Menu > To Do > To Do Entry > Search** to search for the desired To Do entry, then navigate to the **Keys/Values** page.

Description of Page

The **Sort Keys** grid contains an entry's sort keys and values. This information is display-only.

FASTPATH: Refer to [To Do Type - Sort Keys](#) for more information.

The **Drill Keys** grid contains an entry's drill keys and values. This information is display-only.

FASTPATH: Refer to [To Do Type - Drill Keys](#) for more information.

To Do Entry Actions

You can perform the following actions on a To Do Entry.

Complete

Clicking **Complete** causes the To Do entry's status to be **Completed**. When you complete a To Do, a [log entry](#) is created that indicates when the To Do was completed and by whom. This button is enabled for To Do entries with a status of **Open** or **Being Worked On**.

FASTPATH: Refer to [The Lifecycle Of A To Do Entry](#) for more information.

Forward

Clicking **Forward** causes the Forward To Do Entry dialog to appear, allowing you to forward the To Do to another user or role. This button is enabled for **Open** To Do entries assigned to a role to which you belong or **Being Worked On** To Do entries assigned to you.

Send To indicates whether the To Do will be forwarded to a **Role** or **User** (the default).

User appears if **Send To** is set to **User**. The **User** indicates the user to which this To Do entry will be forwarded. The user must be a member of one of the roles specified for the To Do type.

Role is enabled if **Send To** is set to **Role** or if set to **User** and a User is specified. It indicates the role to which this To Do entry will be forwarded. Only users of the specified role will be able to view and work on the To Do entry. The role is defaulted from the To Do type. You can only forward the To Do entry to one of the roles specified on the To Do type. If forwarding to a user the role must also be valid for the specified user.

The **Details** field provides an explanation of why a user is forwarding the To Do. This information appears in the To Do log.

Click **OK** to forward the To Do entry.

Send Back

Clicking **Send Back** causes the Send Back To Do Entry dialog to appear, allowing you to send back the To Do to the user who forwarded it to you. This button is enabled if the To Do entry's status is **Being Worked On** and it was forwarded to you by another user.

User is a display only field that indicates the user who assigned you the To Do.

Role may be used to also change the entry's To Do Role. The role must be a permissible role specified on the To Do type that is also valid for the user this entry is sent back to. By default, if the sent to user is not a member of the entry's current role, the system automatically defaults the role to the first permissible role that is also valid for the user.

The **Details** field provides an explanation of a why the user is sending back the To Do or any other desired information. This information appears in the To Do log.

Click **OK** to send the To Do back to the assigning user.

Log Entry Events

The following table contains the events that cause a log entry to be created by the system.

Event	Log Type	To Do Entry Status	Content of Details	Content of Assigned To (on To Do entry and To Do Log)	Log User ID	Contents of Sent By (on To Do entry)
A To Do is created (by the	Created	Open			ID of user who created the To Do	

Event	Log Type	To Do Entry Status	Content of Details	Content of Assigned To (on To Do entry and To Do Log)	Log User ID	Contents of Sent By (on To Do entry)
system or by a user)						
A To Do is created (by the system or by a user) in Being Worked On status	Created	Being Worked On		User ID of user who was assigned this To Do	ID of user who created the To Do	Creating user's ID
A To Do is completed (by the system or by a user)	Completed	Complete			ID of user who completed the To Do	
A user takes an open To Do	Assigned	Being Worked On	Selected by user	Assigned To User ID	Assigned To User ID	ID of user who assigned the To Do
A supervisor assigns a To Do to a user (also happens if they reassign)	Assigned	Being Worked On	Assigned to user	Assigned To User ID	ID of user who assigned the To Do	ID of user who assigned the To Do
A user forwards an entry to a user	Forwarded	Being Worked On	<i>Log details entered by user</i>	Forwarded To User ID	ID of user who forwarded the To Do	ID of user who forwarded the To Do
A user forwards an entry to a role	Forwarded	Open	Forwarded to role <i>Role Description (Role ID) + Log details entered by user</i>		ID of user who forwarded the To Do	ID of user who forwarded the To Do
A user sends back a To Do	Sent back	Being Worked On	<i>Log details entered by user</i>	Sent Back To User ID	ID of user who sent back the To Do	ID of user who sent back the To Do
A user adds a log entry	User Details	Status remains the same	<i>Log details entered by user</i>		ID of user who added the entry	
A user manually overrides priority	Override Priority	Status remains the same	Priority changed from < <i>old value</i> > to < <i>new value</i> >	Assigned To User ID	ID of user who made the change	

To Do Search

This page allows you to search for To Do entries using various search criteria, and perform various actions on selected ones.

NOTE: No restriction by role. This page returns all To Do entries that meet the search criteria, regardless of the To Do entry's role. Using the search, a user may drill into the details of a To Do entry even if the To Do is not assigned to the user. Refer to [To Do Entries Reference A Role](#) for more information about roles.

FASTPATH: We recommend familiarizing yourself with the topics described in [The Big Picture of To Do Lists](#) to fully appreciate how To Do entries are managed by the system.

To Do Search - Main

Use **Menu > To Do > To Do Search** to open this page.

The top half of the page is where you enter the criteria used to search for To Do entries.

NOTE: Multiple search criteria may be specified. You can search for To Do entries using a combination of search criteria. For example, if you enter both a **To Do Type** of **TD-NOMR** and a **Status** of **Being Worked On**, only those To Do entries for To Do type **TD-NOMR** that are currently **Being Worked On** are displayed.

CAUTION: Try to be as specific as possible when entering search criteria. Why? Because entering open-ended search criteria may have a severe impact on response times.

The following table describes each of the different search filters.

Search Filter	Description
Related To Do	<p>If you're searching for potentially related entries for a given To Do entry, choose Specific To Do for the Related To Do Filter. Enter the To Do ID.</p> <p>A filter value of Not Applicable defaults.</p>
To Do Type	<p>If you're searching for a To Do entry and you know the value of a To Do Type linked to the To Do entry, choose Specific To Do Type for the To Do Type Filter. Enter the To Do Type.</p> <p>A filter value of All defaults.</p>
Char Type	<p>If you're searching for a To Do entry and you know the value of a characteristic linked to the To Do entry, choose Specific Char Type for the Char Type Filter. Enter the Characteristic Type and Characteristic Value.</p> <p>A filter value of Not Applicable defaults.</p> <p>Only characteristic types that include a characteristic entity of To Do Entry and that have been marked to allow searching by characteristic value are available.</p>
User	<p>Use this search method if you know the Created By User, Assigned To User or Completed By User for the To Do entry. Enter the appropriate User Id.</p> <p>A filter value of Not Applicable defaults.</p>
Date Range	<p>Use this field if you're searching for a To Do entry and know the Date Created or Date Completed.</p> <p>A filter value of Date Created defaults. The Start Date/Time defaults to the current date, 12:00AM and the End Date/Time defaults to the current date, 11:59PM.</p> <p>If you enter a Date Range Filter of All, the system ignores this search method.</p>
Status	<p>Use this search method to restrict the search to To Do entries with a specific status value. You may search for To Do entries in the status of Open, Being Worked On or Completed. You may also perform a combination search for To Do entries that have a status value of either Open or Being Worked On.</p> <p>A filter value of All defaults.</p>



NOTE: The system allows any combination of values. The system does not try to validate your search combinations, even if they do not make sense. For example, if you enter a **User Filter** value of **Completed By User** and a **Status Filter** value of **Open**, you will not get an error. In this case, the system will find no records that match your criteria (because To Do entries that are Open do not have a Completed By User).

The system shows the total number of To Do entries that satisfy your search results immediately below the grid.

The first group of To Do entries is displayed in the grid at the bottom of the page.

- Click the **Select All** button to select all entries displayed in the grid. Click the **Clear All** to remove the selections in the grid.
- Use the checkbox if you want to assign, reopen or complete specific entries. Only **Open** or **Being Worked On** entries may be selected.

FASTPATH: Refer to [To Do Search Actions](#) and the [How To](#) section for more information on using these actions.

- A Comment icon  appears if there are comments on the To Do item and a Log Entry icon  appears if there is a log entry of type **Forwarded**, **Sent Back** or **User Details** on the To Do item. This column appears if at least one To Do item has either comments or a special log entry. Use either icon to navigate to the To Do entry page.
- Entries are sorted by their **Priority**. Highest in priority are listed first.
- **Create Date/Time** contains the date and time the To Do entry was created by the system. Use the hyperlink to navigate to the To Do entry page.
- Click the number of **Related** entries hyperlink to redisplay this page listing potentially related entries to the current entry.
- **Last Assigned Date/Time** contains the date and time the To Do entry was most recently assigned to a user, if applicable, along with the Assigned To user.
- **Message** contains a description of the To Do entry. If the To Do entry is in the status of **Open** or **Being Worked On**, the message description allows you to drill into the page on which the entry's respective object is displayed. If the To Do type has been configured to associate a script with this To Do entry's message number, then instead of being taken to the respective page, the associated script is launched. Refer to [Working On A To Do Entry](#) for more information.

NOTE: Current To Do. Drilling into the page by clicking the Message hyperlink does not automatically assign **Open** To Do entries to the user (as the To Do List page does when clicking on the Message). But it does treat this as the user's current to do, enabling To Do related toolbar icons. Refer to [A User's Current To Do](#) for more information.

- **Status** contains the current status of the To Do entry.
- **Completion Date/Time** contains the date and time the To Do entry was completed and the user who completed the entry, if applicable.
- **To Do Type** contains a description of the To Do type for the To Do entry.

To Do Search Actions

You can perform the following actions on selected To Do Entries.

Assign Selected Entries

Use the **Assign** button to assign or reassign the selected entries to a user. Enter a user ID or search for a user using the **Assign To User** field. By default this field is populated with the current user's ID. Note that the user's current number of assignments is displayed adjacent.

Upon successful processing, the selected entries are assigned to the specified user and the status of **Open** entries becomes **Being Worked On**. Additionally, a corresponding [log entry](#) is added to each To Do.

NOTE: Only users with **Supervisor Assignment** access rights may reassign entries that are not currently assigned to them.

Reopen Selected Entries

Use the **Reopen** button to unassign the selected entries from their users forwarding them back to their role.

Upon successful processing, the selected entries become **Open** and a corresponding [log entry](#) is added to each To Do.

Complete Selected Entries

Use the **Complete** button to complete the selected entries.

Upon successful processing, the selected entries become **Completed** and a corresponding [log entry](#) is added to each To Do.

FASTPATH: Refer to [The Lifecycle Of A To Do Entry](#) for more information.

How To

The following sections describe how to perform common tasks for To Dos.

How To Assign a To Do to a User

There are several ways to assign a To Do entry to a user:

- A supervisor can use the [Supervisor Functions](#) pages to review, assign and reassign To Dos to users.
 - The [Supervisor To Do Summary](#) page summarizes by type the total, **Open** (unassigned) and **Being Worked On** (assigned) entries for roles to which you belong.
 - The **Total**, **Open** or **Being Worked On** column hyperlinks transfers you to the [Supervisor To Do Assignment](#) page where you can assign / reassign one or more of the corresponding entries to a specific user.
- Use the [To Do List Functions](#) pages to review and assign To Dos to yourself.
 - The [To Do Summary](#) page summarizes by type the total, **Open** (unassigned) and **Being Worked On** (assigned to you) entries for roles to which you belong.
 - The **Open** column hyperlink transfers you to the [To Do List](#) page where you can assign one or more of the **Open** entries to yourself.
- You may also use the [To Do Search](#) page to find certain To Do entries based on various criteria and assign selected entries to a user.
- Forwarding an entry to a user using the [To Do Entry](#) page is yet another way of assigning an entry to a user.
- The [Current To Do](#) zone next assignment functionality can be configured to dynamically assign an **Open** To Do to the requesting user based on your organization's work assignment rules. Refer to this zone for more information.

How To Work on To Do Entries


Depending on your organization's [work distribution](#) practice, there are several ways to work on To Do entries.

When work is manually assigned, a typical approach may be to review the entries in a list and work them one entry at a time.

- You can use the [To Do List](#) page to review and work on To Do entries by list type.
- You may also use the [To Do Search](#) page to find certain To Do entries based on various criteria and work on selected entries. For example, you may want to review and work on entries related to a given To Do.

Regardless of the above methods to define your work list, when ready to work on a specific entry, drill into its respective page using the message description hyperlink. Drilling into an entry from the above work lists makes it your current To Do entry. This entry's key information is displayed on the [Current To Do](#) dashboard zone and you may use special action bar buttons to scroll through your work list to the next and previous entries respective to your current To Do.

FASTPATH: Refer to [Working On A To Do Entry](#) for more information.

When work is assigned "on demand", click the Current To Do icon, , on the Current To Do dashboard zone to request your next task. The returned To Do entry automatically becomes your current To Do. The entry's key information is displayed on the zone but since the entry was not selected from a work list the next and previous action bar buttons are disabled.

FASTPATH: Refer to [Current To Do](#) zone to learn about all the functions it provides to efficiently resolve the current To Do and potentially other To Do entries that may be [related](#) to it.

Resolve the current To Do by fixing the problem indicated in the message and mark the entry as **Complete**.

FASTPATH: Refer to [How To Complete To Dos](#) for more information.

How To Add a New To Do

If your system is configured with [manual To Do types](#), you can add a To Do entry online. Online To Do entries may be used like electronic help tickets in the system. In Oracle Utilities Customer Care and Billing, for example, if you are having a problem starting service, you can create a To Do that describes the problem. The To Do can be assigned to a help resolution group that could either resolve the problem or send the To Do back to you with information describing how to resolve the problem.

FASTPATH: Refer to [To Do Entry - Add Dialog](#) for more information.

How To Complete To Dos

When you complete a To Do, a log entry is added that indicates when the To Do was completed and by whom.

There are multiple ways in which you can complete a To Do.

- The [To Do List](#) page allows you to complete one or more selected entries of a specific To Do Type.
- The [To Do Search](#) page allows you to search for To Do entries using various search criteria and complete selected ones.
- Use the [To Do Entry](#) page to complete a single To Do Entry.
- You may also use the [Current To Do](#) zone to complete the current entry you are working on and/or all [related](#) entries assigned to you.
- The system automatically completes some To Do entries. For To Do entries that are logically associated with the state of some object, the system automatically marks the entry **Complete** when the object is no longer in the respective state. In Oracle Utilities Customer Care and Billing for example, an entry that's created when an account doesn't have a bill cycle is completed when the account has a bill cycle.

NOTE: The automatic completion of To Do entries occurs when the To Do entry process responsible for creating entries of a given type is executed. Therefore, if these processes run only once per day, these entries remain **Being Worked On** even if the object is no longer in the respective state.

How To Forward a To Do

If you have been assigned a To Do that you are unable to resolve, you can forward it to another user or role for help or additional instructions. When a To Do is forwarded, a log entry indicates who forwarded the To Do, to whom it was forwarded, and the specified reason.

- Use the **Forward** button on the [To Do Entry](#) page to forward a To Do entry to a role or a user.
- You may also use the **Reopen** button on the [To Do Search](#) page to forward selected entries back to their current role.

How To Send Back a To Do

If you have been forwarded a To Do by another user, you may send back the To Do with instructions to the forwarding user. When a To Do is sent back, a log entry is created that indicates when the To Do was sent back and by whom.

Use the **Send Back** button on the [To Do Entry](#) page to send back a To Do entry.

How To Add a Log Entry to a To Do

You can use the **Comments** section of a [To Do entry](#) to add a comment to the To Do as long as the To Do is **Open** or **Being Worked On**. However, the Comments field is disabled if the To Do entry is complete.

Alternatively, you can add a **User Details** log entry to a To Do at any time (even after it's completed). Refer to [To Do Entry](#) page for more information.

Reports

The following section describes the functionality provided for submitting reports on-line and for viewing reports submitted through a scheduler.

FASTPATH: For information about this functionality, see [The Big Picture of Reports](#).

On-line Report Submission

The on-line report submission page enables you to request a specific report to be run. This is only available if your system is able to integrate with your third party reporting system on-line. The system comes configured to integrate with BI Publisher and Business Objects Enterprise.

Report Submission - Main

This page allows you to submit a report on-line. Navigate to this page using **Menu > Batch > Report Submission..**

Description of Page

To submit a report, choose the **Report Code** for the process you wish to submit. Refer to [Defining Report Definitions](#) for information about defining reports in the system.

A description of the report is displayed.

The parameters defined for the report on the [report definition](#) page are displayed in the grid. To see a description of a parameter, use the search button adjacent to the parameter label.

NOTE: Default note. The default values displayed for the parameters are defined on the [report definition](#) page.

When you have entered the appropriate parameter values, click the **Submit** button to submit the report. Refer to [How To Request Reports](#) for more information.

NOTE: Reports submitted via the on-line report submission page are NOT logged in the report history repository and are therefore not visible from the [report history](#) page. Only reports submitted through your third party reporting tool's batch scheduler are logged in the report history.

Viewing Historic Reports

If your third party reporting tool stores historic copies of reports in a repository, you may configure the system to allow a user to view these historic reports on-line.

NOTE: The system comes with the ability to display reports stored in the Business Objects Enterprise report repository. Only reports scheduled through Business Objects Enterprise's batch schedule are stored in the repository.

Report History - Main

This page allows you to view a report stored in your third party reporting tool's report repository. Navigate to this page using **Menu > Batch > Report History**.

Description of Page

The top part of the page shows the **Report Code**, **Description** and **Long Description** for the selected report.

To select a report, search by either **Report Code** or **Description** and click the adjacent search button. The reports matching your search criteria are displayed. Choose the desired report from the selection list.

NOTE: Displaying Report History. When you choose a report, the system interfaces with your reporting software to display a list of historical occurrences of this report. Refer to [Interface with Business Objects Enterprise's Batch Scheduler](#) for more information.

Chapter 2

Administrative User Guides

This section describes administrative features and functionality in Oracle Utilities Application Framework and OracleReal-Time Scheduler.

Framework Administrative User Guide

The topics in this section describe how to administer the Oracle Utilities Application Framework.

Defining General Options

This section describes control tables that are used throughout your product.

Defining Installation Options

The topics in this section describe the various installation options that control various aspects of the system.

Installation Options - Main

Select **Admin > General > Installation Options - Framework** to define system wide installation options.

Description of Page

The **Environment ID** is a unique universal identifier of this instance of the system. When the system is installed, the environment id is populated with a six digit random number. While it is highly unlikely that multiple installs of the system at a given implementation would have the same environment ID, it is the obligation of the implementers to ensure that the environment ID is unique across all installed product environments.

System Owner will be **Customer Modification**.

The **Admin Menu Order** controls how the various control tables are grouped on Admin.

- If you choose **Alphabetical**, each control table appears under a menu item that corresponds with its first letter, using a Roman alphabet. For example, the Language control table will appear under the L menu item entry.
- If you choose **Functional**, each control table appears under a menu item that corresponds with its functional area. Note, the [menu](#) that is used when this option is chosen is the one identified with a menu type of **Admin**.

NOTE: The **Alphabetical** option only supports the Roman alphabet. For languages that do not use the Roman alphabet, the recommendation is to configure the system for the **Functional** setting.

CAUTION: In order to improve response times, installation options are cached the first time they are used after a web server is started. If you change the Admin Menu Order and you don't want to wait for the cache to rebuild, you must clear the cached information so it will be immediately rebuilt using current information. Refer to [Caching Overview](#) for information on how to clear the system login cache (this is the cache in which installation options are stored).

The **Language** should be set to the primary language used by the installation. Note that if multiple languages are supported, each user may define their preferred language.

The **Currency Code** is the default currency code for transactions in the product.

If your product supports effective dated characteristics on any of its objects, define the date to be used as the **Characteristic Default Date** on objects without an implicit start date. The date you enter in this field will default when new characteristics are added to these objects (and the default date can be overridden by the user).

Active Owner displays the owner of newly added system data (system data is data like algorithm types, zone types, To Do types, etc.). This will be **Customer Modification** unless you are working within a development region.

Country and **Time Zone** represent the default country and time zone that should be used throughout the application.

Turn on **Seasonal Time Shift** if your company requires seasonal time shift information to be defined. Note that this is currently only applicable to Oracle Customer Care and Billing > Interval Billing functionality.

Installation Options - Messages

Select **Admin > General > Installation Options - Framework** and the **Messages** tab to review or enter messages that will appear throughout the application when a given event occurs.

The **Message** collection contains messages that are used in various parts of the system. For each message, define the **Installation Message Type** and **Installation Message Text**. The following table describes the **Message Types** provided by the framework product and how they are used in the system. Your specific product may have introduced addition message types.

Message Type	How The Message Is Used
Company Title for Reports	This message appears as a title line on the sample reports provided with the system. Generally it is your company name. It is only used if you have installed reporting functionality and are using the sample reports (or have designed your reports to use this message).

Installation Options - Algorithms

Select **Admin > General > Installation Options - Framework** and the **Algorithms** tab to review or enter the algorithms that should be evoked when a given event occurs.

The grid contains **Algorithms** that control important functions in the system. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).

- Specify the **Sequence Number** and **Algorithm** for each system event. You can set the **Sequence Number** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

CAUTION: These algorithms are typically significant processes. The absence of an algorithm might prevent the system from operating correctly.

The following table describes each **System Event**.

System Event	Optional / Required	Description
Validate Email Attachment	Optional	Algorithms of this type are used to validate the attachments for size and total count while sending attachments using the Email service. Refer to Sending Email for more information. Click here to see the algorithm types available for this system event.
Geocoding Service	Optional	Algorithms of this type use Oracle Locator to retrieve latitude and longitude coordinates using address information. Click here to see the algorithm types available for this system event.
Global Context	Optional	Algorithms of this type are called whenever the value of one of the global context fields is changed. Algorithms of this type are responsible for populating other global context values based on the new value of the field that was changed. Refer to Global Context Overview for more information. Click here to see the algorithm types available for this system event.
Guaranteed Delivery	Optional	Algorithms of this type may be called by processes that receive incoming messages that should 'guarantee delivery'. Refer to Guaranteed Delivery for more information. The business service F1-GuaranteedDelivery may be used to invoke this plug-in spot. Click here to see the algorithm types available for this system event.
Ldap Import	Optional	Algorithms of this type are called for operations on users, groups, and group memberships after they have been processed. Click here to see the algorithm types available for this system event.
Ldap Import Preprocess	Optional	Algorithms of this type are called to preprocess data retrieved from LDAP. Click here to see the algorithm types available for this system event.
Next To Do Assignment	Optional	This type of algorithm is used to find the next To Do entry a user should work on. It is called from the Current To Do dashboard zone when the user ask for the next assignment. Click here to see the algorithm types available for this system event.
Reporting Tool	Optional	If your installation has integrated with a third party reporting tool, you may wish to allow your users to submit reports on-line using report submission or to review report history online. This algorithm is used by the two on-

System Event	Optional / Required	Description
		line reporting pages to properly invoke the reporting tool from within the system. Click here to see the algorithm types available for this system event.
SMS Receive Service	Optional	This type of algorithm is used to provide SMS receive service. Only one algorithm of this type should be plugged in. Click here to see the algorithm types available for this system event.
SMS Send Service	Optional	This type of algorithm is used to provide SMS send service. If your installation uses the base algorithm that uses BPEL, you will need to create a feature configuration with the SMS Send Configuration feature type to define your Oracle BPEL server and service call details. If your installation has integrated with a third-party SMS service, you may want to override this algorithm type with your own implementation. Only one algorithm of this type should be plugged in. Click here to see the algorithm types available for this system event.
To Do Information	Optional	We use the term To Do information to describe the basic information that appears throughout the system to describe a To Do entry . Plug an algorithm into this spot to override the system default "To Do information". Click here to see the algorithm types available for this system event.
To Do Pre-creation	Optional	These types of algorithms are called when a To Do entry is being added. Click here to see the algorithm types available for this system event.

Installation Options - Accessible Modules

Select **Admin > General > Installation Options - Framework** and the **Accessible Modules** tab to view the list of accessible modules.

Description of Page

This page displays the full list of the application's function modules. A **Turned Off** indication appears adjacent to a module that is not accessible based on your system's module configuration setup.

FASTPATH: Refer to [Module Configuration](#) for more information on function modules and how to turn off modules that are not applicable to your organization.

Installation Options - Installed Products

Select **Admin > General > Installation Options - Framework** and the **Installed Products** tab to view a read only summary of the products that are installed in the application version that you are logged into.

Description of Page

The **Product Name** indicates the name of the "products" that are installed. The collection should include **Framework**, an entry for your specific product and an entry for **Customer Release**.

Release ID shows the current release of the application that is installed. This field is used by the system to ensure that the software that executes on your application server is consistent with the release level of the database. If your implementation of the product has developed implementation-specific transactions, you can populate the Release Id for the **Customer Release** entry to define the latest release of implementation-specific logic that has been applied to this environment. In order for this to work, your implementation team should populate this field as part of their upgrade scripts.

The **Release ID Suffix**, **Build Number** and **Patch Number** further describe the details of your specific product release.

The **Display** column indicates the product whose name and release information should be displayed in the title bar. Only one product sets this value to **Yes**.

Owner indicates if this entry is owned by the base package or by your implementation (**Customer Modification**).

Product Type indicates if the product is a Parallel Application. A parallel application is one that is independent of, and does not conflict with, other parallel applications. Multiple parallel applications can be installed in the same database and application server.

NOTE: About Information. The information on this tab is used to populate the information displayed in the [About](#) information for your product.

Support For Different Languages

User Language

The system provides support for multiple languages in a single environment. System users can use the system in their preferred language, as long as a translation into that language has been provided. A user sees the system in the language defined on their [user record](#). If enabled, users can use the [Switch Language](#) zone to switch to another supported language real time.

NOTE: Normally, setting up the system for another language is an implementation issue, not an administrative setup issue. However, there are several online administrative features that are used to set up a new language, and these are described here.

The following steps are required to support a new language:

1. **Define a language code and indicate that it is enabled.** For details on this procedure, see [Defining Languages](#).
2. **Copy descriptions of all language-enabled tables from an existing translation (e.g., English).** The copied values act merely as placeholders while the strings are translated into the new language. It is necessary to do this as a first step in order to create records using the new language code created in the previous step. Language-based descriptions can be copied using a supplied batch process, [FI-LANG](#). The batch copies all English labels in the system.
3. **Apply the language pack.** If the product supplies a language pack with translations for the system metadata descriptions, follow the instructions provided with the language pack to add the translated text.
4. **Translate additional content.** Translatable descriptions and labels for implementation data may be updated / entered in the application. First the user record must be updated to reference the new language. This may be done in one of the following ways:
 - a. Switch to the new language using the [Switch Language](#) zone.
 - b. If that zone is not available, navigate to the user page, assign the new language code to your User ID, sign out, and sign back in again.

Any online functions that you access will use your new language code. You can change the language code for all users who plan to use/modify the new language.

Additional Topics

Your product may support additional uses for language. For example, in Oracle Utilities Customer Care and Billing, you can define each customer's language. This allows you to send bills and other correspondence in each customer's preferred language. For more information, navigate to the index entry **Languages, Customer Language**.

Defining Languages

A language code exists for every language spoken by your users. The system uses this code to supply information to users in their respective language. Select **Admin > General > Language** to define a language.

Description of Page

Enter a unique **Language Code**. If you are applying a language pack provided by the product, use the language code designed by the language pack.

Enter the **Description** for the language. Typically this should be the name of the language in that language.

Turn on **Language Enable** if the system should add a row for this language whenever a row is added in another language. For example, if you add a new currency code, the system will create language specific record for each language that has been enabled. You would only enable multiple languages if you have users who work in multiple languages. Languages that are configured as enabled, appear in the [Switch Language](#) dashboard zone. In addition, the login page for the application displays all the languages that are enabled, allowing the user to toggle the login instructions in that language.

NOTE: The list of enabled languages is captured on the server at startup time. If a new language is enabled, contact your server administrator to refresh the server in order to see the new language displayed in the login page.

The following two fields control how the contents of grids and search results are sorted by the Java virtual machine (JVM) on your web server:

- The **Locale** is a string containing three portions:
 - ISO language code (lower case, required)
 - ISO country code (upper case, optional)
 - Variant (optional).
- Underscores separate the various portions, and the variant can include further underscores to designate multiple variants. The specific JVM in use by your particular hardware/OS configuration constrains the available **Locales**. Validating the **Locale** against the JVM is outside the scope of this transaction. This means you are responsible for choosing valid **Locales**.

The following are examples of valid locales:

- en_US (this is for American English)
- en_AU (this is for Australian English)
- pt_BR (this is for Brazilian Portuguese)
- fr_FR_EURO (this is for European French)
- ja_JP (this if for Japanese)

In addition, the Java collation API can take a **Collator Strength** parameter. This parameter controls whether, for example, upper and lower-case characters are considered equivalent, or how accented characters are sorted. Valid values for collator strength are **PRIMARY**, **SECONDARY**, **TERTIARY**, and **IDENTICAL**. If you leave this field blank, Java will use its default value for the language. We'd like to stress that the impact of each value depends on the language.

Please see <http://java.sun.com/j2se/1.3/docs/guide/intl/locale.doc.html> for more information about the collator strength for your language.

Display Order indicates if this language is written **Left to Right** or **Right to Left**.

Owner indicates if this language is owned by the base package or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add a language. This information is display-only.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_LANGUAGE](#).

Note that all administrative control tables and system metadata that contain language-specific columns (e.g., a description) reference a language code.

In addition, other tables may reference the language as a specific column. For example, on the User record you indicate the preferred language of the user.

Defining Countries

The topics in this section describe how to maintain countries.

Country - Main

To add or review Country definitions choose **Admin > General > Country > Search**.

The **Main** page is used to customize the fields and field descriptions that will be displayed everywhere addresses are used in the system. This ensures that the all addresses conform to the customary address format and conventions of the particular country you have defined.

Description of Page

Enter a unique **Country** and **Description** for the country.

The address fields that appear in the **Main** page are localization options that are used to customize address formats so that they conform to address requirements around the world. By turning on an address field, you make that field available everywhere addresses for this country are used in the system. You can enter your own descriptions for the labels that suffix each switch; these labels will appear wherever addresses are maintained in the system.

NOTE: For any country where the **State** switch is checked, the valid states for the country must be entered on the **Country - State** tab. When entering address constituents on a record that captures this detail, the value for State is verified against the data in the State table. For any country where there is a component of the address that represents a "state" but your implementation does not want to populate the valid states for that country, choose a different field such as County for this constituent (and define an appropriate label). When entering address constituents on a record that captures this detail, no validation is done for the County column.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_COUNTRY](#).

Country - States

To maintain the states located in a country, choose **Admin > Country > Search** and navigate to the **State** page.

Description of Page

For any country where you have enabled the State switch, use the **State** collection to define the valid states in the **Country**.

- Enter the standard postal abbreviation for the **State** or province.

- Enter a **Description** for this state or province.

Defining Currency Codes

The currency page allows you to define display options related to currency codes that are used by your system. Use **Admin** > **Financial** > **Currency** to define the currency codes in which financial information is denominated.

Description of Page

Enter a unique **Currency** and **Description** for the currency.

Use Currency **Symbol** to define the character that prefixes currency amounts in the system (e.g., \$ for U.S. dollars).

Enter the number of **Decimals** that will appear in the notation for the currency.

NOTE: Please contact your specific product to verify whether it supports a currency with more than 2 decimals.

The **Currency Position** indicates whether the currency symbol should be displayed as a **Prefix** or a **Suffix** to the currency amount.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_CURRENCY_CD](#).

Defining Time Zones

The following topics describe how to design and set up time zones.

Designing Time Zones

NOTE: Oracle Utilities Customer Care and Billing - Interval Billing applications customers should consult the topic *Time Issues* (search the Help index for "time issues") for specific information relating to that product's interval billing time related functionality.

It is recommended that all time sensitive data is stored in the standard time (also called 'physical time') of the base time zone as defined on the installation options. This will prevent any confusion when analyzing data and will ensure that your algorithms do not have to perform any shifting of data that may be stored in different time zones.

The Time Zone entity is used to define all the time zones where your customers may operate. Each time zone should define an appropriate Time Zone Name. This is a reference to an external source that defines time zones, their relationship to Greenwich Mean Time, whether the time zone follows any shifting for summer / winter time (daylight savings time) and when this shift occurs.

When designing your time zones, the first thing to determine is the base time zone. You may choose the time zone where the company's main office resides. Once this is done you can link the time zone code to the installation option as the base time zone. Refer to [Installation Options - Main](#) for more information.

If your company does business beyond your main office's time zone, define the other time zones where you may have customers or other systems with which you exchange data. At this point, your specific product may include configuration tables to capture default time zones, for example based on a postal code or geographic location.

NOTE: Date and time in business object schemas. When defining date / time fields in a BO schema, schema attributes can be used to define whether or not data should be stored in standard time for the base time zone or if it should be stored in the standard time of another time zone (related to the data). In addition, schema attributes can be used to indicate if the display of the time should be shifted to represent the "local time". This is used to adjust for seasonal time differences. For example, if the data is stored in the appropriate time zone, but currently daylight savings time is being

observed, the data will be shifted and shown in the “local” time. In addition, if the data is stored in the base time zone but the data is related to a different time zone, the data will be shown in the time zone appropriate for the data (including the appropriate seasonal adjustment). Refer to Schema Tips on the business object page for more information.

Setting Up Time Zones

Refer to [Designing Time Zones](#) for background information about defining time zones.

Open **Admin > General > Time Zone > Search** to define the time zones and their relation to the base time.

Description of Page

Enter a unique **Time Zone** and **Description** for the time zone.

Select the **Time Zone Name** from the list of Olson time zone values. This value is a reference to an external definition that allows the system to know how the time zone relates to Greenwich Mean Time and information about whether the time zone shifts for summer / winter time and when.

Indicate the **Shift in Minutes** that this time zone differs from the base time zone defined on the Installation Options. This is only applicable for the *Oracle Utility Customer Care and Billing - Interval Billing* application.

Indicate the **Seasonal Time Shift** applicable for this time zone. This is only applicable for the *Oracle Utility Customer Care and Billing - Interval Billing* application.

Default Time Zone Label and **Shifted Time Zone Label** are used for data that is sensitive to time zones and time shifting. It indicates whether the data displayed or data to be input is related to the “standard” time or the “shifted” time. For example, on a day when clocks are turned back one hour, a time entry of 1:30 a.m. needs to be labeled as either 1:30 a.m. standard time or 1:30 a.m. daylight savings time.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_TIME_ZONE](#).

Setting Up Seasonal Time Shift

NOTE: The information in this topic applies only to Oracle Utilities Customer Care and Billing - Interval Billing applications.

Open **Admin > General > Seasonal Time Shift > Search** to define the seasonal time shift schedule.

Description of Page

Enter a unique **Seasonal Time Shift** code and **Description** for the seasonal time shift.

The Collection defines the **Effective Date/Time** (in standard time) that a time zone may shift in and out of standard time. If time is changed from standard time on the effective date/time, enter the **Shift in Minutes** that the time changes from standard time (usually **60**). If the time is changed back to standard time on the effective date/time, enter a **Shift in Minutes** of **0**.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_SEAS_TM_SHIFT](#).

Defining Geographic Types

If your company uses geographic coordinates for dispatching or geographic information system integration, you need to setup a geographic (coordinate) type for each type of geographic coordinate you capture on your premises and/or service points (geographic coordinates can be defined on both premises and service points).

To define geographic types, open **Admin > Geographic > Geographic Type**.

NOTE: Find a customer / premise using a geographic coordinate. In Oracle Utilities Customer Care and Billing there are several queries that allow you to search for customers and premises by geographic type. For example, Control Central and Meter Search allow you to search using the premise or service point geographic type. Refer to the product documentation for more information.

Description of Page

Enter an easily recognizable **Geographic Type** code and **Description**.

Define the algorithm used to validate the **Validation Format Algorithm**. If an algorithm is specified, the system will validate that the geographic location entered on the premise and/or service point for the geographic type is in the format as defined in the algorithm. If you require validation, you must set up this [algorithm](#) in the system.

Click [here](#) to see the algorithm types available for this plug-in spot.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_GEO_TYPE](#).

Defining Work Calendar

Workday calendars are used to ensure system-calculated dates fall on a workday. Select **Admin > General > Work Calendar > Search** to define a workday calendar.

Description of Page

The information on this transaction is used to define the days of the week on which your organization works.

Enter a unique **Work Calendar** and **Description**.

Turn on (check) the days of the week that are considered normal business days for your organization.

Use the collection to define the **Holiday Date**, **Holiday Start Date**, **Holiday End Date**, and **Holiday Name** for each company holiday. **Holiday Start Date** and **Holiday End Date** define the date and time that the holiday begins and ends. For example, your organization might begin a holiday at 5:00 p.m. on the day before the actual holiday.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_CAL_WORK](#).

Defining Display Profiles

When you set up your [users](#), you reference a display profile. A user's display profile controls how dates, times, and numbers displayed. Choose **Admin > General > Display Profile > Search** to maintain display profiles.

Description of Page

Enter a unique **Display Profile ID** and **Description** to identify the profile.

Enter a **Date Format**. This affects how users view dates and how entered dates are parsed. This is a "free format" field with some rules.

- **dd** or **d** is interpreted as the day of the month. The **d** option suppresses a leading 0.
- **MM** or **M** is interpreted as the month number. The **M** option suppresses a leading 0.
- **yyyy**, **yy**, or **y** is interpreted as the year. The year can be 4 or 2 digits. The **y** option allows entry in either 2 or 4-digit form and is displayed in 2-digit form.
- Other characters are displayed as entered. Typically, these other characters should be separators, such as "-", ".", or "/". Separators are optional; a blank space cannot be use.

Here are some examples of date formats.

Format	Displayed / entered as
MM-dd-yyyy	04-09-2001
d/M/yyyy	9/4/2001
yy.MM.dd	01.04.09
MM-dd-y	04-09-01 - in this case you could also enter the date as 04-09-2001

NOTE: For centuries, the default pivot for 2-digit years is **80**. Entry of a 2-digit year greater than or equal to **80** results in the year being interpreted as 19xx. Entry of a 2-digit year less than **80** results in the year being interpreted as 20xx.

In addition, the following date localization functionality is supported. Note that in every case, the date is stored in the database using the Gregorian format. The settings below result in a conversion of the date for the user interface.

- **Hijri Dates**

Entering **iiii** for the year is interpreted as a year entered and displayed in Hijri format. For example, the Gregorian date 2014–05–30 may be entered / displayed as 1435/07/30 for a user whose display profile date format is **iiii/MM/dd**. Note that this functionality relies on date mapping to be defined in the Hijri to Gregorian Date Mapping entry. Refer to [Additional Hijri Date Configuration](#) for more information.

- **Taiwanese Dates**

Entering **tttt** for the year is interpreted as a year entered and displayed in Taiwanese format where year 1911 is considered year 0000. For example, if the Gregorian date is 01-01-2005, it is displayed as 01-01-0094 for a user whose display profile date format is **dd-mm-tttt**.

- **Japanese Dates**

There are two options available for configuring Japanese Era date support. The setting **Gyy** for the year is interpreted as a year entered and displayed using an English character for the era followed by the era number. The letter 'T' is used for dates that fall within the *Taisho* era. The letter 'S' is used for dates that fall within the *Showa* era and the letter 'H' is used for dates that fall within the *Heisei* era. For example, for a user whose display profile date format is **Gyy/mm/dd** the Gregorian date 2008/01/01 is shown as **H20/01/01** ; the Gregorian date 1986/03/15 is shown as **S61/03/15**. The setting **GGGGyy** is interpreted as a year entered and displayed using Japanese characters for the era followed by the era number.

Japanese date limitations are as follows:

- The years 1912 through the current date are supported.
- Any functionality that displays Month and Year does not support Japanese Era dates. These dates are shown in Gregorian format.
- Graphs that display dates do not support the GGGGyy format.

Enter a **Time Format**. This is a "free format" display with some rules.

- **hh** or **h** is interpreted as the hour, 1-12; **KK** or **K** is interpreted as the hour, 0-11; **HH** or **H** is interpreted as the hour, 0-23; **kk** or **k** is interpreted as the hour, 1-24. The **h**, **K**, **H**, and **k** options suppress a leading 0.
- **mm** or **m** is interpreted as the minutes. The **m** option suppresses a leading 0.
- **ss** or **s** is interpreted as the seconds. The **s** option suppresses a leading 0.
- **a** is interpreted to mean display **am** or **pm** (only needed when the hour is entered in **hh**, **h**, **KK** or **K** formats). If an **am** or **pm** is not entered, it defaults to **am**.

Here are some examples of time formats.

Format	Displayed / entered as
hh:mma	09:34PM (can be entered as 09:34p)
hh:mm:ss	21:34:00
h:m:s	9:34:0

There are several options for displaying Numbers.

Decimal Symbol defines the separator between the integer and decimal parts of a number. Valid values are "." (a period) or "," (a comma),

Group Symbol defines the means to separate groups of bigger numbers. Valid values are

- "," (comma). Large numbers group by threes separated by a comma, for example 1,000,000.
- "." (period). Large numbers group by threes separated by a period, for example 1.000.000.
- **None**. Large numbers do not have any separator, for example 1000000.
- **South Asian**. This option uses a comma for its separator but will group large numbers as follows: the first comma is used for the thousands separation and numbers over 9,999 are grouped with 2 units, for example 10,00,000.
- **Space**. Large numbers group by threes separated by a space, for example 1 000 000.

Negative Format defines how negative values are displayed. Valid values are **-9.9**, **(9.9)**, or **9.9-**.

Currency values can have a different **Negative Format** from other numbers. Valid values are **-S9.9**, **(S9.9)**, or **S9.9-**, where the "S" represents the currency symbol.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_DISP_PROF](#).

Additional Hijri Date Configuration

For implementations that wish to support displaying dates according to the Hijri calendar, besides appropriate configuration in the [Display Profile](#), the mapping between the Hijri dates and the Gregorian dates must be entered. This mapping is defined in a master configuration record.

Navigate using **Admin > System > Master Configuration**. You are shown a list of master configuration business objects. Find the entry for **Hijri to Gregorian Date Mapping**. Click the add icon if there is no record yet. Click the broadcast icon to view an existing record or the edit icon to modify an existing record.

The mapping record contains a collection of entries for each year in the Islamic calendar.

For each year, clicking the Expand Zone icon shows the mapping collection with the first date of each month of the Hijri calendar. The corresponding date in the Gregorian calendar should be entered for each row.

Defining Phone Types

Phone types define the format for entering and displaying phone numbers.

To add or review phone types, choose **Admin > General > Phone Type**.

Description of Page

Enter a unique **Phone Type** and **Description** for each type of phone number you support.

Select an appropriate **Phone Number Format Algorithm** for each **Phone Type**. This algorithm controls the format for entry and display of phone numbers. Click [here](#) to see the algorithm types available for this plug-in spot.

Use **Phone Type Flag** to define if this type of phone number is a **Fax** number. Defining which phone type is used for facsimile transmittal is only pertinent if your product supports routing of information via fax. For example, in Oracle Utilities Customer Care and Billing, the system may be configured to fax a bill to a customer.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_PHONE_TYPE](#).

Setting Up Characteristic Types & Values

If you need to introduce additional fields to objects that were delivered with minimal fields, you can add a characteristic type for each field you want to capture. Using the characteristic type approach allows you to add new fields to objects without changing the database. There are some pages in the system that support a generic list of characteristics. For those pages, no changes are required for users to view and maintain characteristics. Other pages are business object oriented pages and their maintenance and display are controlled by the UI maps defined for the business objects. In those pages the display / maintenance of the characteristics are driven by the design of the business object and its maps.

The topics in this section describe how to setup a characteristic type.

There Are Four Types Of Characteristics

Every characteristic referenced on an object references a characteristic type. The characteristic type controls the validity of the information entered by a user when they enter the characteristic's values. For example, if you have a characteristic type on user called "skills", the information you setup on this characteristic type controls the valid values that may be specified by a user when defining another user's skills.

When you setup a characteristic type, you must classify it as one of the following categories:

- **Predefined value.** When you setup a characteristic of this type, you define the individual valid values that may be entered by a user. A good example of such a characteristic type would be one on User to define one or more predefined skills for that user. The valid values for this characteristic type would be defined in a discreet list.
- **Ad hoc value.** Characteristics of this type do not have their valid values defined in a discreet list because the possible values are infinite. Good examples of such a characteristic type would be ones used to define a user's birth date or their mother's maiden name. Optionally, you can plug-in an algorithm on such a characteristic type to validate the value entered by the user. For example, you can plug-in an algorithm on a characteristic type to ensure the value entered is a date.
- **Foreign key value.** Characteristics of this type have their valid values defined in another table. For example perhaps you want to link a user to a table where User is not already a foreign key. Valid values for this type of characteristic would be defined on the user table. Please be aware of the following in respect of characteristics of this type:
 - Before you can create a characteristic of this type, information about the table that contains the valid values must be defined on the [foreign key reference table](#).
 - The referenced table does not have to be a table within the system.
 - Not all entities that support characteristics support foreign key characteristics. Refer to the data dictionary to identify the entities that include the foreign key characteristic columns.
- **File Location.** Characteristics of this type contain a URL. The URL can point to a file or any web site. Characteristics of this type might be useful to hold references to documentation / images associated with a given entity. For example, the image of a letter sent to you by one of your customers could be referenced as a file location characteristic on a customer contact entry. When such a characteristic is defined on an entity, a button can be used to open the URL in a separate browser window. Note that for references to a file, the recommendation is to use the Attachment functionality to link documents to an object rather than a characteristic type of File Location. Refer to [Attachment Overview](#) for more information. The documentation related to file location remains for upgrade purposes.

Searching By Characteristic Values

For certain entities in the system that have characteristics, you may search for a record linked to a given characteristic value. The search may be done in one of the following ways:

- Some base searches provide an option to search for an object by entering Characteristic Type and Characteristic Value.

- Your implementation may define a customized search for an entity by a characteristic value for a specific characteristic type using a query data explorer.
- Your implementation may require a business service to find a record via a given characteristic value. For example, maybe an upload of user information attempts to find the user via an Employee ID, defined as a characteristic.

Not all entities that support characteristics support searching by characteristics. Refer to the data dictionary to identify the characteristic collections that include the search characteristic column.

CAUTION: For ad-hoc characteristics, only the first 50 bytes are searchable. For foreign key characteristics, the search value is populated by concatenating the values of each foreign key column to a maximum of 50 bytes.

For the base searches that provide a generic option to search by characteristic type and value, you can restrict the characteristic types that can be used to search for an entity. For example, imagine you use a characteristic to define a "jurisdiction" associated with a To Do for reporting purposes. If your company operates within a very small number of jurisdictions, you wouldn't want to allow searching for a To Do by jurisdiction, as a large number of To Do entries would be returned.

A flag on the [characteristic type](#) allows an administrator to indicate if searching by this characteristic type is **allowed** or **not allowed**.

Characteristic Type - Main

To define a characteristic type, open **Admin > System > Characteristic Type > Search**.

Description of Page

Enter an easily recognizable **Characteristic Type** and **Description** for the characteristic type. **Owner** indicates if this characteristic type is owned by the base package or by your implementation (**Customer Modification**).

CAUTION: Important! If you introduce a new characteristic type, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

Use **Type of Char Value** to classify this characteristic type using one of the following options (refer to [There Are Four Types Of Characteristics](#) for more information):

- **Predefined value.** Characteristics of this type have their valid values defined in the **Characteristic Value** scroll, below. For each valid value, enter an easily recognizable **Characteristic Value** and **Description**.
- **Ad hoc value.** Characteristics of this type capture free form text. If you use this option, you can optionally define the **Validation Rule** used to validate the user-entered characteristic value. Click [here](#) to see the algorithm types available for this plug-in spot.
- **File location value.** Characteristics of this type contain a URL. The URL can point to a file or any web site. Note that for references to a file, the recommendation is to use the Attachment functionality to link documents to an object rather than a characteristic type of File Location. Refer to [Attachment Overview](#) for more information. The documentation related to file location remains for upgrade purposes.

File location characteristic values must be entered in a "non-relative" format. For example, if you want to define a characteristic value of *www.msn.com*, enter the characteristic value as `http://www.msn.com`. If you omit the `http://` prefix, the system will suffix the characteristic value to the current URL in your browser and attempt to navigate to this location when the launch button is pressed. This may or may not be the desired result.

NOTE:

Due to browser security restrictions, opening URLs using the file protocol ("file://") from pages retrieved using http does not work for Internet Explorer version 7 or later, or in Firefox. If the file protocol is used, the browser either does not return properly or an error is thrown (e.g., "Access Denied", which usually results from cross site scripting features added for security reasons).

This issue has no known workaround. To comply with browser security standards, the recommendation is to move the target files to an FTP or HTTP server location to avoid protocols that are subject to browser security restrictions.

- **Foreign key reference.** Characteristics of this type have their valid values defined in another table. If you choose this option, you must use **FK Reference** to define the table that controls the valid values of this characteristic type. Refer to [Setting Up Foreign Key References](#) for more information.

Use the **Allow Search by Char Val** to indicate if searching for an entity by this characteristic type is **Allowed** or **Not Allowed**. Refer to [Searching by Characteristic Values](#) for more information.

The **Custom** switch is only applicable to **Predefined value** types. It indicates whether or not an implementation is allowed to add values for a characteristic type whose owner is not **Customer Modification**

- If this switch is turned on, an implementation may add characteristic values to the grid for system owned characteristic types.
- If this switch is turned off, an implementation may not add characteristic values to the grid for system owned characteristic types.

NOTE: Regardless of the value of the Custom switch, an implementation may not update or remove system owned characteristic values.

The **Characteristic Value** grid defines the valid values for a **Predefined value** type of characteristic.

The **Characteristic Value** is the unique identifier of the value.

Description is the text that is visible in the dropdowns and display when viewing this characteristic value.

Owner indicates if this characteristic value is owned by the system or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add characteristic values to a characteristic type. This information is display-only.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_CHAR_TYPE](#) in the data dictionary schema viewer.

Characteristic Type - Characteristic Entities

To define the entities (objects) on which a given characteristic type can be defined, open **Admin > System > Characteristic Type > Search** and navigate to the **Characteristic Entities** tab.

Description of Page

Use the **Characteristic Entity** collection to define the entities on which the characteristic type can be used. **Owner** indicates if this is owned by the base package or by your implementation (**Customer Modification**).

NOTE: The values for this field are customizable using the Lookup table. This field name is **CHAR_ENTITY_FLG**.

NOTE: For some entities in the system, the valid characteristics for a record are defined on a related "type" entity. For example, in Oracle Utilities Customer Care and Billing the meter type defines valid characteristics for meters of that type. When configuring your system, in addition to defining the appropriate entity for a characteristic type, you may also need to link the characteristic type to an appropriate entity "type". This technique is typically not followed for business object driven maintenance objects, where the business objects can be configured with the appropriate "flattened" characteristic types in the schema.

Setting Up Foreign Key Reference Information

A Foreign Key Reference defines the necessary information needed to reference an entity in certain table.

You need to set up this control table if you need to validate a foreign key value against a corresponding table. For example, if a schema element is associated with an FK Reference the system validates the element's value against the corresponding table. Refer to [Configuration Tools](#) to learn more about schema-based objects. Another example is characteristics whose valid values are defined in another table (i.e., you use "foreign key reference" characteristic types). Refer to [There Are Four Types Of Characteristics](#) for a description of characteristics of this type.

A FK Reference is used not just for validation purposes. It also used to display the standard information description of the reference entity as well as provide navigation information to its maintenance transaction. Info descriptions appear throughout the UI, for example, whenever an account is displayed on a page, a description of the account appears. The product provides base product FK references for many of its entities as they are used for validation and display of elements in both fixed page user interfaces as well as portal based user interfaces.

An implementation may also see the need to define a foreign key reference. The following points describe what you should know before you can setup a foreign key reference for a table.

- The physical name of the table. Typically this is the primary table of a maintenance object.
- The program used by default to construct the referenced entity's info description. Refer to [Information Description Is Dynamically Derived](#) for more information on how this is used.
- The transaction used to maintain the referenced entity. This is where the user navigates to when using the "go to" button or hyperlink associated with the entity. Refer to [Navigation Information Is Dynamically Derived](#) for more information on how this is used.
- The name of the search page used to look for a valid entity. Refer to [Search Options](#) for more information.

Information Description Is Dynamically Derived

Typically a FK Reference is defined for a maintenance object's primary table. In this case the system dynamically derives the standard information associated with a specific referenced entity as follows:

- Attempt to determine the business object associated with the referenced entity. Refer to the [Determine BO](#) maintenance object algorithm system event for more information. If a business object has been determined, the system lets the business object's [Information](#) plug-in, if any, format the description.
- If a business object has not been determined or the business object has no such plug-in, the system lets the maintenance object's [information](#) plug-in, if any, format the description.
- If the maintenance object has no such plug-in, the system uses the info program specified on the FK Reference to format the description.

NOTE: Technical note. The class that returns the information displayed adjacent to the referenced entity is generated specifically for use as an info routine. Please speak to your support group if you need to generate such a class.

NOTE: Generic routine. The system provides a generic information routine that returns the description of control table objects from its associated language table. By "control table" we mean a table with an associated language table that contains a **DESCR** field. Refer to [Defining Table Options](#) for more information on tables and fields. The java class is `com.splwg.base.domain.common.foreignKeyReference.DescriptionRetriever`.

Navigation Information Is Dynamically Derived

Typically a FK Reference is defined for a maintenance object's primary table. In this case the system dynamically derives the actual transaction to navigate to for a given referenced entity as follows:

- Attempt to determine the business object associated with the referenced entity. Refer to the [Determine BO](#) maintenance object algorithm system event for more information. If a business object has been determined, use the maintenance portal defined as its **Portal Navigation Option** business object option.
- If a business object has not been determined or the business object defines no such option, the system uses the transaction specified on the FK Reference.

Search Options

The product provides two main metaphors for implementing a user interface. For input fields that are foreign keys, search options are dependent on the metaphor used by the page in question.

- A [fixed maintenance page](#) user interface is a page supplied by the base product where only minor enhancements, if any, can be introduced by implementations. The foreign key reference may be used in one of two ways.
 - The based product may use an FK reference to define a base element on one of these pages. If a search is available for such elements, the FK reference's Search Navigation Key is used to implement the search.
 - Entities that support characteristics typically include a generic characteristic collection UI metaphor on these types of pages. In this metaphor, a foreign key characteristic displays a search icon if the FK Reference has configured a Search Navigation Key.
- A [portal based](#) user interface is a more flexible user interface where an implementation has more options for customizing the look and feel. The base product uses UI maps or automatic UI rendering to display input fields. Elements that are foreign keys may display a search icon if the FK reference defines a Search Zone.

NOTE: Defining search zones directly. It's possible for elements on a UI map to define a specific search zone directly in the HTML, rather than using the search zone defined on an FK reference. Refer to the UI map tips for more information on implementing searches using zones.

Foreign Key Reference - Main

To setup a foreign key reference, open **Admin > Database > FK Reference > Search**.

CAUTION: Important! If you introduce a new foreign key reference, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

Description of Page

Enter an easily recognizable **FK** (foreign key) **Reference** code and **Description** for the table.

Enter the name of the **Table** whose primary key is referenced. After selecting a **Table**, the columns in the table's primary key are displayed adjacent to **Table PK Sequence**.

Use **Navigation Option** to define the page to which the user will be transferred when they press the go to button or hyperlink associated with the referenced entity. Refer to [Navigation Information Is Dynamically Derived](#) for more information on how this is used.

The **Info Program Type** indicates whether the default program that returns the standard information description is **Java** or **Java (Converted)**, meaning it was converted into Java.

NOTE: **Java (Converted)** program types are not applicable to all products.

Use **Info Program Name** to enter the Java class / program name.

Refer to [Information Description Is Dynamically Derived](#) for more information on the info program is used.

NOTE: View the source. If the program is shipped with the base package, you can use the adjacent button to display the source code of this program in the [Java docs viewer](#).

Use **Context Menu Name** to specify the context menu that appears to the left of the value.

NOTE: Context Menu Name is not applicable to user interface elements displaying a generic collection using a foreign key characteristic type. It is only applicable for pages utilizing the foreign key compound element type for fixed page user interface and for data displayed in a portal based user interface where the foreign key reference is defined as an attribute for an element. Report parameters that reference foreign key characteristics are an example of a user interface where a context menu is not displayed even if the foreign key reference defines one.

Use **Search Navigation Key** to define the search page that will be opened when a user searches for valid values on a user interface that is a fixed page. Refer to [Search Options](#) for more information.

Use **Search Type** to define the default set of search criteria used by the **Search Navigation Key**'s search page.

Use **Search Zone** to define the search zone that opens when a user searches for valid values when the foreign key reference is configured as an input field on a portal based page. Refer to [Search Options](#) for more information.

Use **Search Tooltip** to define a label that describes the **Search Navigation Key**'s search page.

NOTE: Search Type and Search Tooltip. These attributes are only applicable to user interface elements utilizing the foreign key compound element type on fixed page user interfaces. Report parameters that reference foreign key characteristics are an example of a user interface where this information is not used even if the foreign key reference defines them.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_FK_REF](#).

Defining Feature Configurations

Some system features are configured by populating options on a "feature configuration". Because various options throughout the system may be controlled by settings in feature configuration, this section does not document all the disparate possible options. The topics below simply describe how to use this transaction in a generic way.

For information about specific features:

- Refer to the detailed description of each option type.
- Use the index in the online help and search for 'feature configuration' to find any specific topics describing feature options in the administration guide.

You can create options to control features that you develop for your implementation. To do this:

- Review the lookup values for the lookup field **EXT_SYS_TYP_FLG**. If your new option can be logically categorized within an existing feature type, note the lookup value. If your new option warrants a new feature type, add a lookup value to this lookup field.
- Define the feature's option types. If you have identified an existing feature type to add the options to, find the lookup with the name **xxxx_OPT_TYP_FLG** where **xxxx** is the lookup value of **EXT_SYS_TYP_FLG** noted above. If you decided to create a new feature type (by adding a new lookup value to the **EXT_SYS_TYP_FLG** lookup, you must create a new lookup with the name **xxxx_OPT_TYP_FLG** where **xxxx** is the new value you defined above.

- Flush all caches.

Feature Configuration - Main

To define your feature configuration, open **Admin > System > Feature Configuration > Search**.

Description of Page

Enter an easily recognizable **Feature Name** code.

Indicate the **Feature Type** for this configuration. For example, if you were setting up the options for the batch scheduler, you'd select **Batch Scheduler**.

NOTE: You can add new Feature Types. Refer to the description of the page above for how you can add Feature Types to control features developed for your implementation.

NOTE: Multiple Feature Configurations for a Feature Type. Some Feature Types allow multiple feature configurations. The administration documentation for each feature will tell you when this is possible.

The **Options** grid allows you to configure the feature. To do this, select the **Option Type** and define its **Value**. Set the **Sequence** to **1** unless the option may have more than value. **Detailed Description** may display additional information on the option type.

NOTE: Each option is documented elsewhere. The administration documentation for each feature describes its options and whether an option supports multiple values.

NOTE: You can add new options to base-package features. Your implementation may want to add additional options to one of the base-package's feature types. For example, your implementation may have plug-in driven logic that would benefit from a new option. To do this, display the lookup field that holds the desired feature's options. The lookup field's name is **xxxx_OPT_TYP_FLG** where **xxxx** is the identifier of the feature on the **EXT_SYS_TYP_FLG** lookup value. For example, to add new batch scheduler options, display the lookup field **BS_OPT_TYP_FLG**.

Feature Configuration - Messages

If the feature exists to interface with an external system, you can use this page to define the mapping between error and warning codes in the external system and our system.

Open this page using **Admin > Feature Configuration > Search** and navigate to the **Messages** tab.

Description of Page

For each message that may be received from an external system, define the **Feature Message Category** and **Feature Message Code** to identify the message.

A corresponding message must be defined in the [system message](#) tables. For each message identify the **Message Category** and **Message Number**. For each new message, the Message Category defaults to **90000** (because an implementation's messages should be added into this category or greater so as to avoid collisions during upgrades).

Defining Master Configurations

A master configuration is an object that enables an implementation to define configuration for features in the system. It is an alternative to using feature configuration for defining options. A master configuration is defined using a business object. Only one master configuration may exist for a given business object.

The product provides one or more master configuration that may be used for configuration. Some examples, of base master configuration business objects are as follows

- **Hijri to Gregorian Date Mapping.** This allows an implementation that uses Hijri dates to define the mapping between those dates and Gregorian dates.
- **ILM Configuration.** For implementations that use Information Lifecycle Management, the ILM configuration record defines some parameters used by the process.
- **Migration Assistant Configuration.** For implementations that use the configuration migration assistant (CMA), the configuration record defines some parameters used by the process.

For a list of all the master configuration records provided by the product, navigate to the master configuration page in the application.

Setting Up Master Configurations

To set up a master configuration, open **Admin > System > Master Configuration**.

The topics in this section describe the base-package zones that appear on the Master Configuration portal.

Master Configuration

The Master Configuration List zone lists every category of master configuration.

The following functions are available:

- Click a broadcast button to open other zones that contain more information about the adjacent master configuration.
- Click the **Add/Edit** button to start a business process that updates the master configuration.

Master Configuration Details

The Master Configuration Details zone contains display-only information about a master configuration.

This zone appears when a master configuration has been broadcast from the Master Configuration zone.

Please see the zone's help text for information about this zone's fields.

Miscellaneous Topics

The following sections describe miscellaneous system wide topics.

Module Configuration

The system provides the ability to simplify the user interface based on functionality areas practiced by your organization.

Menu items and other user interface elements are associated with function modules. By default, all function modules are accessible. If a function module is not applicable to your business you may turn it off. Refer to [Turn Off A Function Module](#) for more information on how to turn off a module.

If a function module is made non-accessible, i.e. turned off, its related elements are suppressed from the user interface. In addition the system may validate that related functionality is not accessed. This also means that turning off the wrong module may cause any of the following to occur:

- Menu items may not appear. Refer to [Menu Item Suppression](#) to better understand how menu item suppression works.
- Entire menus may not appear. Refer to [Menu Suppression](#) to better understand how menu suppression works.
- Tabs on pages may not appear.
- Fields may not appear.
- The system may return an error message when you attempt to use a function (indicating the function is turned off).

To correct the above situation, simply remove the module from the turned off list thus making it accessible again. Your module configuration setup is displayed on the [installations](#) record.

Menu Item Suppression

The following points describe how your module configuration can suppress [menu items](#).

- Menu items that are owned by the base product (as opposed to those your implementation adds) are associated with one or more function modules. If your module configuration has turned off all of the menu item's modules, the menu item is suppressed. If at least one of the modules is accessible, i.e. turned on, the menu item is not suppressed.
- If a menu line doesn't contain any accessible items, the menu line is suppressed.
- If all lines on a menu are suppressed, the menu itself (**Menu** or **Admin menu**) is suppressed in the application toolbar.

Menu Suppression

In addition to the above Menu Item Suppression logic, the following points describe how your module configuration can suppress an entire menu.

- Menus that are owned by the base product (as opposed to those your implementation adds) are associated with one or more function modules.
- If your module configuration has turned off all of the menu's modules, the entire menu is suppressed. If at least one of the modules is accessible, i.e. turned on, the menu is not suppressed.

Turn Off A Function Module

The base package is provided with a **Module Configuration** [Feature Configuration](#) that allows your organization to turn off base package function modules.

To turn off any of the base package function modules add a **Turned Off** option to this feature configuration referencing that module. Refer to the **MODULE_FLG** lookup field for the complete list of the application's function modules.

Any module not referenced on this feature configuration is considered turned on, i.e. accessible. To turn on a module, simply remove its corresponding **Turned Off** option from this feature configuration.

You may view your module configuration setup on the [installation options](#) page.

NOTE: Only one. The system expects only one **Module Configuration** feature configuration to be defined.

Global Context Overview

The framework web application provides each product the ability to nominate certain fields to act as a "global context" within the web application. For example, in Oracle Utilities Customer Care and Billing, the global context fields include Account ID, Person ID and Premise ID. The values of these fields may be populated as a result of searching or displaying objects that use these fields in their keys. If you navigate to the Bill page and display a bill, the global context is refreshed with the Account ID associated with that bill. The global context for Person ID and Premise ID are refreshed with data associated with that account.

The fields designated as global context for the product are defined using the lookup **F1_UI_CTXT_FLDS_FLG**.

Changing the values of the global context typically cause data displayed in zones on the dashboard to be refreshed to show information relevant to the current values of these global context fields.

When the value of one of the global context fields changes, an algorithm plugged into the [installation record](#) is responsible for populating the remaining global context values accordingly. Refer to your specific product for more information about the base algorithm that is provided for that product.

System Data Naming Convention

There are several maintenance objects in the system that include owner flag in one or more of its tables. We refer to the data in these tables as "system data". Some examples of system data tables include Algorithm Type, Batch Control, Business Object and Script. Implementations may introduce records to the same tables. The owner flag for records created by an implementation is set to **CM** (for customer modification), however the owner flag is not part of the primary key for any of the system data tables. As a result, the base product provides the following guidelines for defining the primary key in system data tables to avoid any naming conflict.

Base Product System Data

For any table that includes the owner flag, the base product will follow a naming convention for any new data that is owned by the base product. The primary key for records introduced by the product is prefixed with **xn-** where **xn** is the value of the owner flag. For example, if a new background process is introduced to the framework product, the batch code name is prefixed with **F1-**.

NOTE: There are some cases where the hyphen is not included. For example, portal codes omit the hyphen.

For most system data, the remainder of the primary key is all in capital case. An exception is schema oriented records. For business objects, business services, scripts, data areas and UI maps, the product follows the general rule of using CapitalCase after the product owner prefix. For example, **F1-AddToDoEntry** is the name of a base product business service.

NOTE: Data Explorer Business Services. For business services used to invoke a data explorer zone, it is recommended to name the Business Service the same name as the related zone rather than defining a different CapitalCase name for the business service.

Please note that this standard is followed for all new records introduced by the base product. However, there are base product entries in many of these system data tables that were introduced before the naming convention was adopted. That data does not follow the naming convention described above.

NOTE: Schema naming conventions. A context sensitive "Schema Tips" zone is associated with any page where a schema may be defined. The zone provides recommended naming conventions for elements within a schema along with a complete list of the XML nodes and attributes available to you when you construct a schema.

Implementation System Data

When new system data is introduced for your implementation you must consider the naming convention for the primary key. The product recommends prefixing records with **CM**, which is the value of the owner flag in your environment. This is consistent with the base product naming convention. This convention allows your implementation to use the CM packaging tool in the Software Development Kit as delivered. The extract file provided with the tool selects system data records with an owner flag of **CM AND** with a **CM** prefix.

NOTE: If you choose not to follow the CM naming convention for your records and you want to use the CM packaging tool, your implementation must customize the extract file to define the appropriate selection criteria for the records to be included in the package. Refer to the Software Development Kit documentation for more information.

Also note that owner flag may be introduced to an existing table in a new release. When this happens, the CM packaging tool is also updated to include these new system data tables. Your implementation will have existing records in those tables that probably do not follow any naming convention. After an upgrade to such a release, if you want to include this data in the CM packaging tool, you must customize the extract file for the tables in question.

Caching Overview

A great deal of information in the system changes infrequently. In order to avoid accessing the database every time this type of information is required by an end-user or a batch process, the system maintains a cache of static information on the web server. In addition to the web server's cache, information is also cached on each user's browser.

Server Cache

The cache is populated the first time any user accesses a page that contains cached information. For example, consider a control table whose contents appear in a dropdown on various pages. When a user opens one of these pages, the system verifies that the list of records exists in the cache. If so, it uses the values in the cache. If not, it accesses the database to retrieve the records and saves them in the cache. In other words, the records for this control table are put into the cache the first time they are used by any user. The next user who opens one of these pages will have the records for this control table retrieved from the cache (thus obviating the database access).

The following points describe the type of data that is cached on the web server:

- **Field labels.** This portion of the cache contains the labels that prefix fields on the various pages in the system.
- **System information.** This portion of the cache contains installation and basic information about the various application services (e.g., the URL's that are associated with the various pages).
- **Menu items.** This portion of the cache contains the menu items.
- **Dropdown contents.** This portion of the cache contains the contents of the various dropdowns that appear throughout the system.
- **XSL documents.** This portion of the cache contains each page's static HTML.
- **Portal information.** This portion of the cache contains information about which zones are shown on the various pages.

The contents of the cache are cleared whenever the web server is "bounced". This means that fresh values are retrieved from the database after the application server software is restarted.

If you change the database after the cache is built and the information you changed is kept in the cache, users may continue to see the old values. If you don't want to bounce your web server, you can issue one of the following commands in your browser's URL to immediately clear the contents of the cache (a separate command exists for each type of data that is held in the cache):

- **flushAll.jsp?language=<language code>.** This command flushes every cache described below.
- **flushDropDownCache.jsp?language=<language code>.** This command flushes all objects in the dropdown contents associated with a given language (note, the **language code** is defined on the [Language](#) control table). Use this whenever you change, add or delete values to/from control tables that appear in dropdowns. Unlike the above caches, the objects in the dropdown cache are flushed automatically every 30 minutes from the time they are first built.
- **flushDropDnField.jsp?language=<language code>&key=<field>** If you want to flush the values in a specific dropdown field in the cache, specify the *<field>* using this command.
- **flushMenu.jsp.** This command flushes the menu cache. Use this command if you change menu data.
- **flushMessageCatalog.jsp.** This command flushes the field labels. Use this whenever you add or change any labels (this is typically only done by implementers who have rights to introduce new pages).
- **flushMessaging.jsp.** This command flushes messages.
- **flushNavigationInfo.jsp.** This command flushes the navigation information in the cache.

- **flushPortalMetaInfo.jsp**. This command flushes the portal information cache. Use this command whenever you change any portal zone meta-data.
- **flushSystemLoginInfo.jsp**. This command flushes the system information cache. Use this whenever you change navigation options, cached information from the installation record, or if you change an application service's URL.
- **flushUI_XSLs.jsp**. This command flushes the XSL document cache. Use this command if you change user interface meta-data. Also use this command whenever you change or introduce new zones.

For example, assume the following:

- the web server and port on which you work is called **OU-Production:7500**
- you add a new record to a control table and you want it to be available on the appropriate transactions immediately (i.e., you cannot wait for 30 minutes)

You would issue the following command in your browser's address bar: **http://OU-Production:7500/flushDropdownCache.jsp?language=ENG**. Notice that the command replaces the typical `cis.jsp` that appears after the port number (this is because these commands are simply different JSP pages that are being executed on the web server).

Client Cache

In addition to the web server's cache, information is also cached on each user's browser. After clearing the cache that's maintained on the web server, you must also clear the cache that's maintained on your client's browser. To do this, follow the following steps:

- Select **Tools** on your browser's menu bar
- Select **Internet Options...** on the menu that appears.
- Click the **Delete Files** button on the pop-up that appears.
- Turn on **Delete all offline content** on the subsequent pop-up that appears and then click **OK**.
- And then enter the standard URL to re-invoke the system.

NOTE: Automatic refresh of the browser's cache. Each user's cache is automatically refreshed based on the **maxAge** parameter defined in the `web.xml` document on your web server. We recommend that you set this parameter to **1** second on development / test environments and **28800** seconds (8 hours) on production environments. Please speak to system support if you need to change this value.

Batch Cache

When submitting a batch job, the batch component uses a caching mechanism using a Hibernate data cache. The tables whose records are included in this cache are configured using the Caching Regime value of **Cached for Batch**. Refer to [Table - Main](#) for more information.

When starting a thread pool worker, data in tables marked as cached is loaded and cached for as long as that thread pool is running. If there is a change in cached data that should be available for the next batch job, either the thread pool worker must be restarted or alternatively, the **F1-FLUSH** (Flush all Caches) background process can be submitted. This background process receives the thread pool name as input. It will flush the data cached for batch for that thread pool.

Debug Mode

Your implementation team can execute the system using a special mode when they are configuring the application. To enable this mode, enter **?debug=true** at the end of the URL that you use to access the application. For example, if the standard URL was **http://CD-Production:7500/cis.jsp**, you'd enter **http://CD-Production:7500/cis.jsp?debug=true** to enable configuration mode.

When in this mode certain debugging oriented tools become available right below the main toolbar.

- **Start Debug** starts a logging session. During this session the processing steps that you perform are logged. For example, the log will show the data areas that are passed in at each step and the data areas returned after the step is processed.
- **Stop Debug** stops the logging session.
- **Show Trace** opens a window that contains the logging session. All of the steps are initially collapsed.
- **Clear Trace** clears your log file.
- **Show User Log** allows you to view your own log entries. The number of "tail" entries to view may be specified in the adjacent **Log Entries** field before clicking the button. Limiting the number of entries to view allows the user to quickly and easily see only the latest log entries without having to manually scroll to the end of the log.
- Checking the **Global Debug** indication starts various tracing options.

Other parts of the system may show additional configuration oriented icons when in this mode. For example, explorer zones may provide additional tools to assist in debugging zone configuration. These icons are described in the context of where they appear.

Also, in debug mode drop down lists in data explorer and UI map zones will contain the code for each item in addition to the item's display string.

NOTE: Show User Log button is secured. An application service **FIUSERLOG** has been provided for this functionality to allow implementations to restrict user access to this button. Such restriction may be called for in production environments.

System Override Date

The system provides a way to override the system date used for online operations. This feature is available if the server administrator has enabled it in the environment properties. For instructions on configuring environment properties see the *Server Administration Guide*. The system date override feature is not recommended for production environments.

Under the **General System Configuration**[Feature Configuration](#), the **System Override Date Option Type** holds the date the application will use as the global system date instead of retrieving the same from the database. This feature can be especially useful in running tests that require the system date to be progressed over a period of time.

The system override date feature is also available at the user level. This is useful when a user wants override the system date to run tests without affecting the system date for other users in the environment. In order to override the system date for the user, open the **Admin > User > Search**, page. On the **Characteristics** tab, add the **System Override Date** characteristic type with a characteristic value set to the desired date in the YYYY-MM-DD format.

If system override dates are defined at both the feature configuration level and the user level, the date set at the user level will take precedence.

Advanced Search Options

The product supports fuzzy searching in explorer zone types using the Oracle Text CONTAINS operator.

Refer to the DBA guide for details on setting up the database to support fuzzy searching. Note that there are some implementations where fuzzy searching will not be possible. For example, it's only available for implementations using the Oracle database. Additionally, not all languages are supported. Refer to the Oracle Database documentation for more information about fuzzy searching.

For information about the particular syntax to use in the explorer zones, refer to the Zone Tips context zone available in the Dashboard on the Zone portal.

Defining Security & User Options

The contents of this section describe how to maintain a user's access rights.

The Big Picture of Application Security

The contents of this section provide background information about application security.

Application Security

The system restricts access to its transactions as follows:

- An [application service](#) may be associated with every securable function in the system.
 - All maintenance objects define an application service that includes the basic actions available, typically **Add**, **Change**, **Delete**, and **Inquire**. The base product supplies an application service for every maintenance object.
 - For maintenance objects whose user interface page is not portal-based, the application service also controls whether the menu entry appears. If a user doesn't have access to the maintenance object's application service, the menu item that corresponds with the application service will not be visible.
 - For portal based user interfaces, each main portal defines an explicit application service with the access mode **Inquire**, allowing the user interface to be secured independently of the underlying object security. If a user doesn't have access to the portal's application service, the menu item that corresponds with the application service will not be visible. The base product supplies an application service for every portal that is accessible from the menu.
 - Menu items may define an application service. Use this technique for the following scenarios:
 - Suppress a menu item if the underlying application security for the transaction does not provide enough fine grained control. For example, imagine your implementation creates a special BPA script to add a To Do Entry and would like users to use the special BPA rather than the base supplied Add dialogue for To Do Entry. The underlying security settings for To Do Entry should grant Add access to these users given that the special BPA will still add a record. To suppress the base Add dialogue, link a special application service and access mode for the base supplied menu item for To Do Entry Add. Then define a menu entry for the new special BPA for adding.
 - Suppress the add option if a user does not have add security for the object. By default the product does not suppress the add function if a user does not have add access to the object. Rather, the user is prevented from adding the record at the back-end. If your implementation would like to suppress the icon, link the object's application service and the Add access mode to the Add menu item.
 - Zones define an application service
 - For zones linked to a portal, if a user doesn't have access to the zone's application service, the zone will not be visible on the portal. In most cases the zone defines the same application service as its portal. In special cases, such as the zones on the Dashboard, the product supplies separate application services for each zone allowing implementations to determine at a more granular level which users should have access to which zones.
 - For query zones that are configured on a multi-query zone, if a user doesn't have access to the zone's application service, the zone will not be visible in the dropdown on the multi-query zone. In most cases all zones in a multi-query zone define the same application service as the multi-query zone. The product may supply a special application service for one or more zones in a multi-query zone if the functionality is special to certain markets or jurisdictions and not applicable to all implementations.
 - For zones that are used by business services to perform SQL queries, the product supplies a default application service. Security for these zones is not checked by the product as they are used for internal purposes.

- Business objects define an application service. If the business object defines a lifecycle, the application service must include access modes that correspond to each state. In addition, the standard maintenance object access modes of **Add, Change, Delete** and **Inquire** are included. The base product business objects are supplied with appropriate application services.
- Other configuration tool objects are securable but the base product typically does not supply special application services for each object. An implementation may supply custom application services and link them to the appropriate record:
 - BPA scripts may define an application service with the access mode **Execute**. The base BPA scripts are typically not configured with any application service. An implementation may define one. Note that as mentioned above, a menu item may also be configured with an application service and access mode. This allows for a BPA that is invoked via a menu entry to be secured in more than one way.
 - Business Services and Service Scripts define an application service with the access mode **Execute**. This is needed for services that may be executed from an external system, for example via an inbound web service. Base business services and service scripts are configured with a default application service, which may be overridden by an implementation.
- Users are granted access to application services via [user groups](#). For example, you may create a user group called Senior Management and give it access to senior manager-oriented pages and portals.
 - When you grant a user group access to an application service with multiple access modes, you must also define the access modes that are allowed. Often the access modes correspond to an action on a user interface. For example, you may indicate a given user group has **inquire**-only access to an application service, whereas another user group has **add, change, cancel** and **complete** access to the same service. Refer to [action level security](#) for more information.
 - If the application service has [field level security](#) enabled, you must also define the user group's security level for each secured field on the transaction.
 - And finally, you link individual [users](#) to the user groups to which they belong. When you link a user to a user group, this user inherits all of the user group's access rights.

CAUTION: Menus may be suppressed! If all menu items on a menu are suppressed, the menu is suppressed.

Action Level Security

When you grant a user group access to an [application service](#), you must indicate the actions to which they have access.

- For application services that only query the database, there is a single action to which you must provide access - this is called **Inquire**.
- For application services that can modify the database, you must define the actions that the user may perform. At a minimum, most maintenance transactions support **Add, Change, and Inquire** actions. Additional actions are available depending on the application service's functions.

CAUTION: Important! If an application service supports actions that modify the database other than **Add, Change, and Delete**; you must provide the user with **Change** access in addition to the other access rights. Consider a transaction that supports special actions in addition to **Add, Change, and Inquire** (e.g., **Freeze, Complete, Cancel**). If you want to give a user access to any of these special actions, you must also give the user access to the **Inquire and Change** actions.

Field Level Security

Sometimes transaction and action security is not sufficient. There are situations where you may need to restrict access based on the values of data. For example, in Oracle Utilities Customer Care and Billing you might want to prevent certain users from completing a bill for more than \$10,000. This is referred to as "field level security".

Field level security can be complex and idiosyncratic. Implementing field level security always requires some programming by your implementation group. This programming involves the introduction of the specific field-level logic into the respective application service(s).

NOTE: Field level security logic is added to user exits. Refer to the Public API chapter of the Software Development Kit Developer Guide for more information on how to introduce field-level security logic into an application service's user exits.

Even though the validation of a user's field-level security rights requires programming, the definition of a user's access rights is performed using the same transactions used to define transaction / action level security. This is achieved as follows:

- Create a [security type](#) for each type of field-level security.
- Define the various access levels for each security type. For example, assume you have some users who can complete bills for less than \$300, and other users who can complete bills for less than \$1,000, and still other users who can complete bills for any value. In this scenario, you'd need 3 access levels on this security type:
 - Level 1 (lowest): May authorize bills <= \$300
 - Level 2 (medium): May authorize bills <= \$1,000
 - Level 3 (highest): May authorize all bills
- Link this security type to each [application service](#) where this type of field level security is implemented. This linkage is performed on the [security type](#) transaction.
- Defining each [user group's](#) access level for each security type (this is done for each application service on which the security type is applicable).

NOTE:

Highest value grants highest security. The system expects the highest authorization level value to represent highest security level. Moreover, authorization level is an alphanumeric field so care should be taken to ensure that it's set up correctly.

Encryption and Masking

"Encryption" refers to encrypting data stored in a database using an encryption key.

"Masking" refers to overwriting all or part of a decrypted field value with a masking character before it is presented to a user (or an external system) without the appropriate security access. For example, an implementation can mask the first 12 digits of a credit card number with an asterisk for users who do not have security rights to see credit card numbers. This section is describing defining masking rules that may be applied to data for display on the user interface.

Multiple masking rules. The system allows different masking rules to be applied to different fields. For example, a credit card number can be masked differently than a social security number. In addition, some user groups may be allowed to see certain fields unmasked.

Masking happens after decryption. It is obvious, but worth emphasizing, that only decrypted data can be masked. This means that if a user does not have authority to retrieve decrypted data then masking is not relevant because the data to be masked would be encrypted.

The topics in this section describe how to mask field values.

Identify the Fields to Be Masked

Your implementation should list every field on every page and inbound web service request that requires masking

Classify each field into one of the following categories:

- An element that is retrieved by invoking a business object, a business service, or a service script

- A field that is retrieved by invoking a page service
- A field that is retrieved by invoking a search service
- An ad hoc characteristic type's value

Primary keys cannot be masked. A field defined as a unique identifier of a row cannot be configured for masking. Masking a field that is part of the primary key causes a problem when attempting to update the record. This restriction also applies to elements that are part of a "list" in an XML column on a maintenance object. One or more elements in the list must be defined as a primary identifier of the list. Be sure that primary key elements in the list are not ones that require masking.

Masking applies to strings. Only fields with a data type of String can be masked.

List members that contain different "types". Consider a page with a list that contains a person's phone numbers. You can set up the system so that a person's home phone has different masking rules than their work number. If your implementation has this type of requirement, the list of masked fields should contain an entry for each masking rule.

Create a Security Type For Each Logical Field

Examine the list of masked fields and look for "logical fields".

For example, assume your list contains the following masked fields:

- The Person - Main page retrieves information using the person page service. This page contains a grid that contains the various forms of ID associated with the person. Entries in the grid with an ID Type of "Social Security Number" are subject to masking.
- The Control Central - Main page retrieves data by invoking a search service. This service shows a person's primary form of ID in the search results. If a person's primary ID is their social security number then it is subject to masking.
- The Control Central - Account Information page contains a map zone that retrieves data by invoking a service script. One of the elements in this script's schema holds the person's social security number and it is subject to masking.

In the above example, there is a single "logical field" associated with the three secured elements: the social security number.

Examine your list and define the distinct logical fields. For each one, create a security type with two authorization levels:

- **1** - Can only see the element masked
- **2** - Can only see the element unmasked

You should link all of the security types to an application service of your choosing. We recommend linking every masking-oriented security type to a single application service (e.g., **CM_MASK**) as it makes granting access easier.

Create An Algorithm For Each Security Type

A masking algorithm must be created for each security type.

These algorithms determine if a user has the rights to view a given field unmasked, and, if not, how the field should be masked.

The base package provides the algorithm type **F1-MASK** whose parameters are designed to handle most masking needs. This algorithm type's parameters are described below to provide the complete picture of how to control how a field is masked and who can see it unmasked:

- **Masking Character:** Enter the character used to overwrite a field's value for users who can only see masked values.
- **Number of Unmasked Characters:** If some of the field value should remain unmasked at the end of the string, enter the number of unmasked characters.
- **Unmasked Characters:** If some characters in a field value should never be masked, enter them. For example, if a phone number can contain dashes and parenthesis and you don't want these characters masked, you would enter - () . .

- **Application Service:** Enter the application service that you created above.
- **Security Type:** Enter the security type that you created above.
- **Authorization Level:** Enter the authorization level that a user must have to see this field unmasked. If you followed the recommendations above, this will be **2**.

Create A Feature Configuration For Each Secured Element

Create a feature configuration with a Feature Type of **Data Masking**.

Add an option for every field that you defined in [Identify The Fields To Be Masked](#).

Each field's option value will have an **Option Type** of **Field Masking** and a **Value** that references the respective algorithm defined above. In addition, the Value will contain mnemonics that differ depending on how the field is retrieved.

NOTE: Only fields defined as strings are supported.

Schema Based Object Field Masking

For data that is accessed via a schema-based object call and displayed in a UI map, the field to be masked must reference a meta-data field name in its schema definition: **field="fld_name", alg="algorithm name"**

If the element references an mdField in the schema, that is the field used to identify the masking rule. If there is no mdField reference but only a mapField reference, that is the field used to identify the masking rule. For example, if you want to mask a credit card number, let's assume that field is defined in the schema is the following:

```
<creditCard mdField="CCNBR" mapField="EXT_ACCT_ID"/>
```

In this case, the option value should be **field="CCNBR", alg="algorithm name"**. An option value of **field="EXT_ACCT_ID", alg="algorithm name"** would not result in masking.

A "where" clause may also be specified. This is useful for data that resides in a list where only data of a certain type needs to be masked: **field="fld_name", alg="algorithm name", where="fld_name='value'"**

For example, person can have a collection of IDs and only IDs of type 'SSN' (social security number) should be masked. If the person data including its collection of person IDs is displayed on a UI map via a business object call, let's assume the collection is defined in the following way:

```
<personIds type="list" mapChild=CI_PER_ID">
  <isPrimaryId mapField="PRIM_SW"/>
  <idType mapField="ID_TYPE_CD"/>
  <personIdNumber mapField="PER_ID_NBR"/>
</personIds>
```

The option value may look like this: **field="PER_ID_NBR", alg="algorithm name", where="ID_TYPE_CD='SSN'"**

Please note the following important points for schema based masking:

- **Limitation of 'where' field** Although the main use of a 'where' clause for schema oriented elements is to mask certain elements in a list based on a 'type', it is also possible to mask a single field in the schema based on the value of another field. For example, imagine that a customer submits a registration form that defines an ID type and ID value. Although this data is not in a list, the implementation may still want to only mask the ID value if the ID type is "SSN". The framework is only able to mask an element in the schema based on a 'where' clause if the element in the 'where' clause is a "sibling" in the schema.
 - If the element to be masked is in a list, the element in the 'where' clause must be in the same list.
 - If an element to be masked maps to a real column in a table, the element in the 'where' clause must also map to a real column in the table.
 - If an element to be masked maps to the CLOB of a table as a single element, the element in the 'where' clause must map to the same CLOB as a single element.

- **Multiple feature option entries for the same field.** It's possible that different schemas in the system have a similar type of data that may be masked based on different conditions. For example, imagine that an implementation has different schemas that captured or referenced person identifiers in different ways:

- One schema captures a single person ID without any corresponding "type" record and it should always be masked using Algorithm CM_SSN_MASK:

```
<personSSN mapXML=BO_DATA_AREA mdField=PER_ID_NBR/>
```

- One schema captures a person ID and a corresponding ID Type and it should be masked with Algorithm CM_SSN_MASK if the type is "SSN" and masked with algorithm CM_FEIN_MASK if the type is "FEIN".

```
<personIdType mapXML=BO_DATA_AREA mdField=ID_TYPE_CD/>
<personId mapXML=BO_DATA_AREA mdField=PER_ID_NBR/>
```

- One schema captures a person ID and a corresponding ID Type and it has the same masking rules as the previous schema, but a different field name is used for the ID Type code. (This scenario could happen if for example a different label is desired for ID Type on the user interface for this schema.)

```
<personIdType mapXML=BO_DATA_AREA mdField=CM_ID_TYPE/>
<personId mapXML=BO_DATA_AREA mdField=PER_ID_NBR/>
```

For this scenario, the feature options may look like this:

1. **field="PER_ID_NBR", alg="CM_SSN_MASK"**
2. **field="PER_ID_NBR", alg="CM_SSN_MASK", where="ID_TYPE_CD='SSN'"**
3. **field="PER_ID_NBR", alg="CM_FEIN_MASK", where="ID_TYPE_CD='FEIN'"**
4. **field="PER_ID_NBR", alg="CM_SSN_MASK", where="CM_ID_TYPE='SSN'"**
5. **field="PER_ID_NBR", alg="CM_FEIN_MASK", where="CM_ID_TYPE='FEIN'"**

For each schema, the system will first find whether the element applies to any masking option. It will find 5 masking options for the field PER_ID_NBR. Then it will determine if any sibling elements match the 'where' clause.

- If more than one sibling element matches a 'where' clause, a runtime error is issued. For example if a schema has an element that references "mdField=ID_TYPE_CD" and an element that references "mdField=CM_ID_TYPE", this is an error. Additionally, if multiple elements reference mdField=ID_TYPE_CD", this is an error.
- If one and only one sibling element matches a 'where' clause, the value of the element is compared to the values defined in the 'where' clause. If it finds a match on the value, the appropriate masking algorithm is applied. If no match is found (for example, the Person ID Type is "LICENSE") the element is displayed as is.
- If no sibling element matches a 'where' clause and a feature option exists with no 'where' clause (option 1 above), then the masking algorithm of the option with no 'where' clause is applied.
- **Changing the value in the 'where' clause.** If your implementation has some users that are allowed to change records where some data is masked based on a condition, it is recommended to design the user interface to reset the masked value when the value in the 'where' clause changes. For example, if a user is prevented from viewing a person's social security number, but the user is allowed to make updates to the person's record, changing the value of the Person ID Type should reset the Person ID Number. This would ensure that the user does not 'unmask' the social security number by simply changing the ID Type.

Records Maintained Using Page Maintenance

For data that is accessed via a page maintenance service call, indicate the table name and the field name where the data resides: **table="table_name", field="fld_name", alg="algorithm name"**

For example if the Person record and its collection of identifiers are displayed and maintained using page maintenance, the option value should be **table="CI_PER_ID", field="PER_ID_NBR", alg="algorithm name"**

A "where" clause may also be specified: **table="table_name", field="fld_name", where="fld_name='value'", alg="algorithm name"**

This is useful for data that resides in a child table where only data of a certain type needs to be masked. For the person ID example, `table="CI_PER_ID", field="PER_ID_NBR", alg="algorithm name", where="ID_TYPE_CD='SSN'"`

Characteristic Data

For data that is stored as a characteristic, simply indicate the characteristic type: `CHAR_TYPE_CD='char type', alg="algorithm name"`

This needs to be defined only once regardless of which characteristic entity the char type may reside on. Note that only ad-hoc characteristics are supported.

Masking Fields in Explorer Zones or Info Strings

In explorer zones data is often retrieved using SQL directly from the database. No masking is applied automatically in this case. If there is data in the explorer zone results that should be masked, the masking must be applied by calling a business service.

Similarly, an MO Info algorithm may not use BO interaction to get data. It may access data using SQL for efficiency purposes. No masking is applied when retrieving data via SQL. To apply masking to a string prior to including it in an info string, the masking must be applied by calling a business service.

The system supplies two business services to be called to determine if masking rules apply for a specific field.

- **F1-TableFieldMask** - Mask a Table field. This business service receives a table name, field name and one or more field values. If masking applies it returns the masked value.
- **F1-SchemaFieldMask** - Mask a Schema field. This business service receives a schema name and type, XPath and field value. If masking applies it returns the masked value.

Search Service Results

For data that is displayed via a search service call, indicate the search name and the appropriate field to mask along with the masking algorithm. For example: `search="SearchServiceName", field="PER_ID_NBR", where="ID_TYPE_CD='SSN'", alg="algorithm name"`

To find the name of the search service, launch the search in question, right click in the filter area and choose View Source. Search for ServiceName. The service name is listed there. To find the field name to mask, go back to the search window and right click on the results area and choose View Source. Look for the Widget Info section and find the field name in the SEARCH RESULTS (do not include the \$). Note, the "where" statement can only apply to fields that are also part of the search results.

Grant Access Rights To The User Groups

For each security type, identify which users can see its data unmasked and which users can only see its data masked.

If the masked and unmasked users fit into existing user groups, no additional user groups are necessary. Otherwise, create new user groups for the masked and unmasked users.

After the user groups for each security type are defined, link each user group to the application service defined above. When a user group is linked to the application service, you will define the authorization level for each security type linked to the application service. If a user group's users should see the security type's field values unmasked, set the authorization level to 2; otherwise set it to 1. Refer to [User Group - Application Services](#) for the transaction used to link a user group to an application service.

NOTE: Flush the cache. Remember that any time you change access rights you should flush the security cache (by entering flushAll.jsp on the URL of the application) if you want the change to take effect immediately.

Additional Masking Information

The following points provide additional information to assist in your masking configuration:

- If the demonstration database includes a **Data Masking** feature configuration, review the settings because it will probably contain masking rules that will match your own.
- On data input pages, a user might be able to enter or change masked data, such as a bank account number, but not be able to subsequently see what they added or changed.
- External systems can request information by performing a service call via a web service. Please keep in mind that some web service requests require data to be masked and some do not. For example, a request from an external system to synchronize person information needs the person's social security number unmasked; whereas a request from a web self service application to retrieve the same person information for display purposes needs the person's social security number masked. To implement this type of requirement, different users must be associated with each of the requests and these users must belong to separate user groups with different access rights.
- If a maintenance object (MO) contains a field that holds an XML document and a service call invokes the MO's service program directly, the system will mask individual XML elements in the field if a **Determine BO** algorithm has been plugged into the [maintenance object](#) and the element(s) in the respective BO schema have been secured as described above.

The Base Package Controls One User, One User Group, And Many Application Services

When the system is initially installed, the following information is delivered:

- Application services for all secured transactions, maintenance objects, business objects, business services, scripts and zones in the base package.
- A user identified by the user id **SYSUSER**.
- A user group identified by the user group code **ALL_SERVICES**. This user group is associated with all supported application services delivered with the base product. This user group is given access to all access modes for all application services (i.e., all actions on all transactions).
- The user **SYSUSER** is linked to the **ALL_SERVICES** user group. This means that this user has access to all transactions and all actions.

NOTE: An implementation may use the User Enable setting on the user record to disable SYSUSER so that it is not available as a valid user in the application.

When you receive an upgrade:

- New application services are delivered for the new transactions, business objects, zones introduced in the release.
- Existing application services are updated with changes in their access modes (e.g., if a new action is added to a transaction, its application service is updated accordingly).
- The **ALL_SERVICES** user group is updated so it can access the new / changed application services.

CAUTION: Important! You cannot change or remove the information delivered for **ALL_SERVICES**. This information is owned by the base package. It is provided so that an "initial user" has access to the entire system and can setup user groups and users as per your organization's business requirements. It is not recommended to provide your own users with access to the **ALL_SERVICES** user group. Rather, create user groups that are appropriate for the organization's business requirements and define user access to these user groups. If you introduce new transactions, configure them for the appropriate custom user groups.

Importing Security Configuration from an External Source

The product provides support for importing security information from an external source:

- If your organization uses Lightweight Directory Access Protocol (LDAP), you can import your existing LDAP users and groups into the system. Once imported, all user and group functions are available. You can import a user group, or a single user. You can resynchronize your LDAP users and groups at any time.

FASTPATH: For more information refer to [LDAP Integration](#).

- The system provides an integration with Oracle Identity Manager. When a user is created in the identity manager product, its information can automatically be interfaced to the product. Once the user is successfully created in the system, all functions are available.

FASTPATH: For more information refer to [Oracle Identity Manager Integration](#).

The Big Picture of Row Security

Some products allow you to limit a user's access to specific rows. For example, in Oracle Utilities Customer Care and Billing, row level security prevents users without appropriate rights from accessing specific accounts.

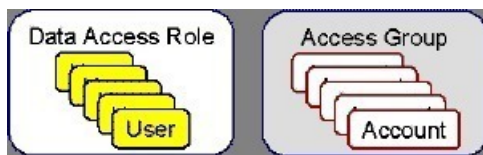
By granting a user access rights to an account, you are actually granting the user access rights to the account's bills, payment, adjustments, orders, etc.

The topics in this section describe basic row level security objects.

FASTPATH: Refer to your product's documentation for more information on row level security, if applicable.

Access Groups, Data Access Roles and Users

We'll use an example from Oracle Utilities Customer Care and Billing to describe how access groups and roles are used to restrict access to accounts. The following diagram illustrates the objects involved with account security:



An **Access Group** defines a group of **Accounts** that have the same type of security restrictions. A **Data Access Role** defines a group of **Users** that have the same access rights (in respect of access to accounts). When you grant a data access role rights to an access group, you are giving all users in the data access role rights to all accounts in the access group.

The following points summarize the data relationships involved with account security:

- An account references a single **access group**. An **access group** may be linked to an unlimited number of **accounts**.
- A **data access role** has one or more **users** associated with it. A **user** may belong to many **data access roles**.
- A **data access role** may be linked to one or more **access group**. An **access group** may be linked to one or more **data access roles**.

If you use row level security, setting up your access roles and groups can be easy or challenging - it all depends on your organization's requirements. Refer to the product's **Administration Guide - Implementing Account Security** for several case studies. These case studies provide examples of how different requirements can be implemented using these concepts.

Defining Application Services

An application service exists for every transaction in the system. Please refer to [Application Security](#) for a description of how application services are used when you grant user groups access rights transactions.

CAUTION: Important! If you introduce a new application service, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

Application Service - Main

Select **Admin > Security > Application Service** to define an application service.

Description of Page

Enter a unique **Application Service** code and **Description** for the application service.

Indicate the application service's various **Access Modes** (i.e., actions). Refer to [Action Level Security](#) for more information about the significance of these fields.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [SC_APP_SERVICE](#).

Application Service - Application Security

Use the Application Security portal to set up security for an application service.

Open this page using **Admin > Security > Application Service**, and then navigate to the **Application Security** tab.

This section describes the available zones on this page.

Application Service Details zone. The Application Service Details zone contains display-only information about the selected application service, including the Access Modes for the application service and its security type.

User Groups with Access zone. The User Groups with Access zone lists the user groups that have access to the application service. The following actions are available:

- Click the **Description** link to navigate to the User Group - Users page for the adjacent user group.
- Click **Deny Access** to remove the user group's access rights from the selected Application Service.
- Use the search filters to display the user groups that contain a specific user.

User Groups without Access zone. The User Group without Access zone lists the user groups that do not have access to the application service. The following actions are available:

- Click the **Description** link to navigate to the User Group - Users page for the user group.
- Click **Grant Access** to navigate to the User Group - Application Services page for the user group. The page is automatically positioned at the selected application service.
- Use the search filters to display the user groups that contain a specific user.

Defining Security Types

Security types are used to define the types of [field level security](#).

NOTE: Programming is required. You cannot have field level security without introducing logic to user exits. Refer to [Field Level Security](#) for more information on how security types are used to define field level security.

Security Type - Main

Select **Admin > Security > Security Type** to define your security types.

Description of Page

Enter a unique **Security Type** and **Description**.

Use the **Authorization Level** grid to define the different authorization levels recognized for this security type. Enter an **Authorization Level Number** and its **Description**.

NOTE: Programming is required. Note that the values that you enter are not interpreted by the system itself, but by the user exit code used to implement the special security. Check with the developer of the user exit logic for the correct values. Refer to [Field Level Security](#) for more information on how security types are used to define field level security.

Use the **Application Services** grid to define the application service(s) to which this security type is applicable. If this application service is already associated with user groups, you must update each user group to define their respective security level. This is performed using [User Group - Application Service](#).

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_SC_TYPE](#).

Defining User Groups

A user group is a group of users who have the same degree of security access. Think of a user group as a "role"; associated with a role are:

- The users who play this role
- The application services to which the role's users have access (along with the actions they can execute for each service and their field level security authorization levels).

User Group - Main

Select **Admin > Security > User Group** to view the application services to which a user has access.

CAUTION: Application services may not be changed or removed from the **ALL_SERVICES** user group. Refer to [The Base Package Controls One User, One User Group, And Many Application Services](#) for an explanation.

Description of Page

Enter a unique **User Group** code and **Description** for the user group.

Owner indicates if this user group is owned by the base package or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add a user group. This information is display-only.

The **Application Services** grid displays the various application services to which users in this group have access. Please note the following in respect of this grid:

- Use the **Application Service** search to restrict the application services displayed in the grid. For example, if you only want to see application services that start with the word "field", you can enter this word and press enter.
- To add additional application services to this user group, navigate to the [User Group - Application Services](#) page and click the add icon.

- To remove or change this user group's access to an application service, click the go to button adjacent to the respective application service. This will cause you to be transferred to the [User Group - Application Services](#) tab where you should click the - icon to remove the application service from the user group.
- Note, **Owner** indicates if this user group / application service relationship is owned by the base package or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add an application service to the user group. This information is display-only.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [SC_USER_GROUP](#).

User Group - Application Services

Select **Admin > Security > User Group** and navigate to the **Application Services** tab to maintain a user group's access rights to an application service.

NOTE: Important! When you grant a user group access rights to an application service, you are actually granting all users in the user group access rights to the application service.

Description of Page

The **Application Service** scroll contains the application services to which the **User Group** has access.

NOTE: You can also use Main page to select the application service for which you wish to change the access privileges. To do this, simply click the go to button adjacent to the respective application service.

To add additional application services to this user group, click the + icon and specify the following:

- Enter the **Application Service ID** to which the group has access.
- Define the **Expiration Date** when the group's access to the application service expires.

Define the **Access Modes** that users in this group have to the **Application Service**. When a new application service is added, the system will default all potential **Access Modes** associate with the **Application Service**. You need only remove those modes that are not relevant for the **User Group**. Refer to [Action Level Security](#) for more information about access modes.

CAUTION: Important! If an application service supports actions that modify the database other than **Add, Change, and Delete**; you must provide the user with **Change** access in addition to the other access rights. Consider a transaction that supports actions in addition to **Add, Change, and Inquire** (e.g., **Freeze, Complete, Cancel**). If you want to give a user access to any of these additional actions, you must also give the user access to the **Inquire** and **Change** actions.

If you require additional security options, often referred to as "field level" security, then you use **Security Type Code** and assign an **Authorization Level** to each. When a new application service is added, the system will display a message indicating how many security types are associated with this application service. Use the search to define each Security Type Code and indicate the appropriate Authorization Level for this user group. Refer to [Field Level Security](#) for more information about security types.

User Group - Users

Select **Admin > Security > User Group** and navigate to the **Users** tab to maintain the users in a user group.

Description of Page

The scroll area contains the users who are part of this user group.

NOTE: Keep in mind that when you add a **User** to a **User Group**, you are granting this user access to all of the application services defined on the **Application Services** tab.

The following fields are included for each user:

- Enter the **User ID** of the user.
- Use **Expiration Date** to define when the user's membership in the group expires.
- **Owner** will be **Customer Modification**.

NOTE: You can also add a user to a user group using [User - Main](#).

Defining Access Groups

FASTPATH: Refer to [The Big Picture of Row Security](#) for a description of how access groups are use to restrict access to specific objects.

Access groups control which groups of users (referred to as Data Access Roles) have rights to accounts (or other objects) associated with the access group. Select **Admin > Security > Access Group** to define your access groups.

Description of Page

Enter a unique **Access Group** code and **Description** for the data access group.

Use the **Data Access Role** collection to define the data access roles whose users have access to the access group's accounts (or other objects). Keep in mind that when you add a **Data Access Role** to an **Access Group**, you are granting all users who belong to this role access to all of the accounts (or other objects) linked to the access groups. Refer to [Access Groups, Data Access Roles and Users](#) for more information.

NOTE: You can also use [Data Access Role - Access Group](#) to maintain a data access role's access groups.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_ACC_GRP](#).

Defining Data Access Roles

FASTPATH: Refer to [The Big Picture of Row Security](#) for a description of how access groups are use to restrict access to specific objects.

The data access role transaction is used to define two things:

- The users who belong to the data access role.
- The access groups whose accounts (or other objects) may be accessed by these users.

Data Access Role - Main

Select **Admin > Security > Data Access Role** to define the users who belong to a data access role.

Description of Page

Enter a unique **Data Access Role** code and **Description** for the data access role.

The scroll area contains the **Users** who belong to this role. A user's data access roles play a part in determining the accounts (or other objects) whose data they can access. Refer to [Access Groups, Data Access Roles and Users](#) for more information.

To add additional users to this data access role, press the add button, **+**, and specify the following:

- Enter the **User ID**. Keep in mind that when you add a **User** to a **Data Access Role**, you are granting this user access to all of the accounts (or other objects) linked to the data access role's access groups.
- Use **Expiration Date** to define when the user's membership in this data access role expires.

NOTE: Also maintained on the user page. You can also use [User - Access Security](#) to maintain a user's membership in data access roles.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_DAR](#).

Data Access Role - Access Group

Select **Admin > Security > Data Access Role** and navigate to the **Access Groups** tab to define the access groups whose accounts (or other objects) may be accessed by the users in this data access role.

Description of Page

Use the **Access Group** collection to define the access groups whose objects can be accessed by this role's users. Keep in mind that when you add an **Access Group** to a **Data Access Role**, you are granting all users who belong to this role access to all of the accounts (or other objects) linked to the access groups. Refer to [Access Groups, Data Access Roles and Users](#) for more information.

NOTE: You can also use [Access Group - Main](#) to maintain an access group's data access roles.

Defining Users

The user maintenance transaction is used to define a user's user groups, data access roles, portal preferences, default values, and To Do roles. To access the user maintenance transaction, select **Admin > Security > User**.

The user maintenance transaction is the same transaction invoked when the user launches [Preferences](#).

User Interface Tools

This section describes tools that impact many aspects of the user interface.

Defining Menu Options

The contents of this section describe how you can add and change menus.

CAUTION: Updating menus requires technical knowledge of the system. This is an implementation and delivery issue and should not be attempted if you do not have previous experience with menus.

NOTE: Security and menus. Refer to [Application Security](#) for a discussion of how application security can prevent menu items (or an entire menu) from appearing.

NOTE: Module configuration and menus. Your [module configuration](#) can prevent menu items (or an entire menu) from appearing.

Menu - Main

This transaction is used to define / change any menu in the system. Navigate to this page using **Admin > System > Menu**.

Description of Page

Enter a meaningful, unique **Menu Name**.

Owner indicates if this menu line is owned by the base product or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add a menu line. This information is display-only.

The **Flush Menu** button is used to flush the cached menu items so you can see any modified or newly created menus. Refer to [Caching Overview](#) for more information.

Menu Type defines how the menu is used. You have the following options:

- **Admin** is one of the menus that appears in the Application Toolbar. It is a special type of menu because [admin menu](#) items can be grouped alphabetically or by functional group. Refer to the description of Admin Menu Order on [Installation Options - Framework](#) for more information about admin menu options.
- **Context** refers to a [context menu](#).
- **Main** is another menu that appears in the Application Toolbar that is simply titled [Menu](#).
- **Page Action Menu** defines buttons that appear in the [Page Title Area](#).
- **Submenu** defines a menu group that appears when an Application Toolbar menu is selected. for the Admin menu, this is only visible when it's organized functionally.
- Enter **User Menu** refers to the menu items that appear on the [user menu](#); for example, User Preferences.

Description provides a description of the menu. Note that this is not the text used when displaying a menu option.

Sequence is only enabled for the **Main** and **Admin** menu types.

The grid contains a summary of the menu's lines. Besides the standard add and delete icons available in a grid, the following information is displayed:

- **Menu Line ID** is the unique identifier of the line on the menu. This information is display-only. Before the menu line id is a Go To icon that allows a user to drill into the Menu Items for the displayed menu line.
- **Sequence** is the relative position of the line on the menu. Note, if two lines have the same **Sequence**, the system organizes the lines alphabetically (based on the **Long Label**, which is defined on the next tab).

NOTE: An implementation may override the sequence of a base product owned menu line. Also note that the sequence is defined on the menu line language table, allowing for different orders to be used for different languages (or to let the menu be sorted alphabetically in one language and in a specified order in a different one).

- **Navigation Option / Submenu** contains information about the line's items. If the line's item invokes a submenu, the submenu's unique identifier is displayed. If the line's item(s) invoke a transaction, the description of the first item's [navigation option](#) is displayed.
- **Long Label** is the verbiage that appears on the menu line.
- **Item Count** is the number of menu items on the line.
- **Owner** indicates if this menu line is owned by the base product or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add a menu line. This information is display-only.

NOTE: Adding menu lines to base owned menus. An implementation may choose to add custom menu lines along with its menu item (or items) to a base owned menu.

Refer to the description of [Menu Items](#) for how to add items to a menu line.

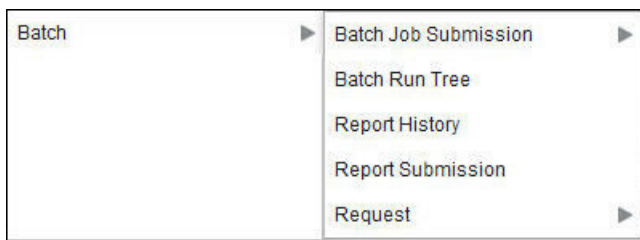
Menu - Menu Items

Once a menu has lines (these are maintained on the main page), you use this page to maintain a menu line's items.

Each menu line can contain one or two menu items. The line's items control what happens when a user selects an option on the menu.

There are two types of menu lines that define a single menu item: one type causes a submenu to appear; the other type causes a transaction or script to be invoked when it's selected.

- The following is an example of a menu line with a single item that opens a submenu:



- The following is an example of a menu line with a single menu item that launches a transaction or script:



A menu line that defines two menu items is used to provide an Add option and a Search option for the same type of object. In this case each menu item defines a transaction or script to be launched. The menu is rendered with the Add and Search options displayed. The following is an example of a menu line with two menu items.



Navigate to this tab by clicking the Go To button adjacent to a menu line from the Main tab.

Description of Page

Menu Name is the name of the menu on which the line appears. **Menu Line ID** is the unique identifier of the line on the menu. **Owner** indicates if this menu is owned by the base product or by your implementation (**Customer Modification**). This information is display-only.

The **Menu Line Items** scroll contains the line's menu items. The following points describe how to maintain a line's items:

- **Menu Item ID** is the system assigned unique identifier of the item.
- **Owner** indicates if this item is owned by the base product or by your implementation (**Customer Modification**).
- If the menu item should invoke a submenu:
 - Use **Sub-menu Name** to identify the menu that should appear when the line is selected
 - Use **Long Label** to define the verbiage that should appear on the menu line
 - Populate the **Override Label** to override the long label of a base product owned sub-menu.
- If the item should invoke a transaction or BPA script:

- Use **Sequence** to define the order the item should appear in the menu line (we recommend this be set to **1** or **2** as a menu line can have a maximum of 2 menu items). The “search” menu item should be defined as sequence 1 and the “add” menu item as sequence 2 given that the label of the “search” menu item is used for the menu line’s label.
- Use **Navigation Option** to define the transaction or script to open. Refer to [Defining Navigation Options](#) for more information.
- For a menu line that includes two items — one for Add and one for Search, if one of the items includes configuration for **Image GIF Location and Name** , the system assumes that this represents the Add. This functionality is a carry over from earlier releases where the Add function rendered in the menu with a “+” image, which also identified the item that represents the Add. If neither item includes Image configuration (because it is no longer needed for rendering the menu), the system relies on the order of the items as mentioned above. The first item is the “search” and the second item is the “add”.
- **Image Height, Image Width and Balloon Description** are not applicable at this time.
- Use the **Long Label** to define the text to appear on the menu entry. Note that when a menu line defines two menu items, the long label on the search entry is used to build the menu entry text. The label long on the menu line that defines the Add option is information only.
- The **Override Label** is provided in case you want to override the base-package's label.
- Use **Application Service** and **Access Mode** to easily suppress a menu item for one or more users. Refer to [Application Security](#) for more information.

The Big Picture of System Messages

All error, warning and informational messages that are displayed in the system are maintained on the message table. Every message is identified by a combination of two fields:

- **Message category number.** Think of a message category as a library of messages related to a given functional area. For example, there is a message category for billing messages and another one for payment messages.
- **Message number.** A unique number identifies each message within a category.

Every message has two components: a brief text message and a long description. On the **Main** tab, you can only maintain the brief message. If you need to update a message's long description, you must display the message on the **Details** tab.

NOTE: You cannot change the product’s text. If the message is "owned" by the product, you cannot change the product’s message or detailed description. If you want your users to see a different message or detailed description other than that supplied by the product, display the message on the **Details** tab and enter your desired verbiage in the "customer specific" fields (and flush the [cache](#)).

Defining System Messages

The contents of this section describe how to maintain messages that appear throughout the system. An implementation may introduce messages used in custom processes or may choose to override the text for messages delivered by the product.

Message - Main

Select **Admin > System > Message** to maintain a message category and its messages.

Description of Page

To add a new message category, enter a **Message Category** number and **Description**.

CAUTION: Message category 80000 or greater must be used to define new messages introduced for a specific implementation of the system. Changes to other Message Text will be overwritten when you next upgrade. If you want

to make a change to a Message, drill down on the message and specify Customer Specific Message Text. Note that even for message categories 80000 and higher, message numbers lower than 1000 are reserved for common base product messages.

NOTE: Owner indicates if this message category is owned by the base product or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add a category. This information is display-only.

To update a message, you must first display its **Message Category**. You can optionally start the message grid at a **Starting Message Number**.

To override the message text or detailed description of messages owned by the base product, click on the message's go to button. When clicked, the system takes you to the **Details** tab on which you can enter your implementation's override text.

The following points describe how to maintain messages owned by your implementation:

- Click the - button to delete a message.
- Click the + button to add a new message. After clicking this button, enter the following fields:
- Use **Message Number** to define the unique identifier of the message within the category.
- Use **Message Text** to define a basic message. You can use the %n notation within the message text to cause field values to be substituted into a message. For example, the message text **The %1 non-cash deposit for %2 expires on %3** will have the values of 3 fields merged into it before it is displayed to the user (%1 is the type of non-cash deposit, %2 is the name of the customer, and %3 is the expiration date of the non-cash deposit).

NOTE: The system merges whatever values are supplied to it. Therefore, if a programmer supplies a premise address as the second merge parameter in the above message, this address is merged into the message (rather than the customer's name).

- **Owner** indicates if this message number is owned by the base product or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add a message. This information is display-only.
- Click the go to button to enter a detailed description of the message. Clicking this button transfers you to the **Details** tab.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_MSG](#). In addition, messages are used throughout the system for error messages and other system messages.

Message - Details

Select **Admin > System > Message** and navigate to the **Details** tab to define detailed information about a message.

NOTE: Drilling in from the Main tab. Rather than scrolling through the messages, you can display a message by clicking the respective go to button in the grid on the main tab.

Description of Page

The **Message Collection** scroll contains an entry for every message in the grid on the Main tab. It's helpful to categorize messages into two categories when describing the fields on this page:

- Product messages
- Implementation-specific messages (i.e., a message added to **Message Category** 80000 or greater)

For product messages, you can use this page as follows:

- If you want to override a message, specify **Customer Specific Message Text**.

- You are limited to the same substitution values used in the original **Message Text**. For example, if the original **Message Text** is **The %1 non-cash deposit for %2 expires on %3** and %1 is the type of non-cash deposit, %2 is the name of the customer, and %3 is the expiration date of the non-cash deposit; your **Customer Specific Message Text** is limited to the same three substitution variables. However, you don't have to use any substitution variable in your message and you can use the substitution variables in whatever order you please (e.g., %3 can be referenced before %1, and %2 can be left out altogether).
- If you want to override the detailed description of an error message, specify **Customer Specific Description**. Note that the system does not present detailed descriptions when warnings are shown to users. Therefore, it doesn't make sense to enter this information for a warning message.

For implementation-specific messages, you can use this page as follows:

- **Message Text** is the same Message Text displayed on the main tab.

CAUTION: If both **Message Text** and **Customer Specific Message Text** are specified, the system will only display the **Customer Specific Message Text** in the dialog presented to the user.

- Use **Detailed Description** to define additional information about an error message. Note that the system does not present detailed descriptions when warnings are shown to users. Therefore, it doesn't make sense to enter this information for a warning message.

CAUTION: If both **Detailed Description** and **Customer Specific Description** are specified, the system will only display the **Customer Specific Description** in the dialog presented to the user.

The Big Picture of Portals and Zones

A portal is a page that is comprised of one or more information zones. The base product pages are built using either a fixed page metaphor or using portals and zones. The contents of this section describe general information about portals and zones.

There Are Three Types of Portals

There are three broad classes of portals:

- **Standalone Portal.** Standalone portals are separate pages where the main tab of the page is built using a portal. These pages are opened using any of the standard methods (e.g., by selecting a menu item, by selecting a favorite link, etc.). Additional tabs for a stand-alone portal may be included using tab page portals.
- **Tab Page Portals.** These types of portals cannot be attached to a menu. They simply define the zones for a tab on either a standalone portal or on a “fixed” page. Please contact customer support if you need to add portals to existing transactions.
- **Dashboard Portal.** The dashboard portal is a portal that appears in the [Dashboard Area](#) on the user’s desktop. Its zones contain tools and information that exist on the user's desktop regardless of the transaction.

There is only one dashboard portal. This portal and several zones are delivered as part of the base-package. Your implementation can add additional zones to this portal. Please contact customer support if you need to add zones to the dashboard portal.

Common Characteristics of All Portals

The topics that follow describe characteristics common to all types of portals.

Portals Are Made Up of Zones

A portal is a page that contains one or more zones, and each zone contains data of some sort.

All zones reference a **Zone Type**. The zone type controls the behavior of the zone and the parameters available to configure the zone.

Configuring Zones for a Portal

The portal includes configuration of how the zones should appear on the portal by default. This includes the following options, all of which may be overridden by an implementation.

- The order in which the zone should appear. An implementation may configure an override sequence to change the order zones on a base delivered portal.
- Whether the zone is visible on the portal. Zones delivered in the base product should be configured to be visible. But an implementation may override this if desired.
- Whether the zone should display initially collapsed or not. A zone's data is only retrieved when it is expanded. As such, a zone may be configured to be initially collapsed when the data is not needed very often. A user can expand the zone when the information is required. Implementations may change the collapsed setting of a base product portal / zone.

FASTPATH: Refer to [Zones May Appear Initially Collapsed When a Page Opens](#) for more information.

FASTPATH: Refer to [Defining Portals](#) for more information about this configuration.

In addition, the portal includes configuration to indicate whether or not the portal should appear on a user's portal preferences. This is typically enabled for a portal that provides disparate information where not all zones are applicable to all users or where users may wish to adjust the order of the zones. An example of a portal enabled for portal preferences is the Dashboard portal. The user can override zone oriented configuration for the portal:

- Which zones appear on that portal
- The order in which the zones appear
- Whether the zones should be initially collapsed when the portal opens.
- The refresh seconds. This is applicable to zones displaying data that changes often.

An implementation can optionally configure the system to define portal preferences on one or more "template" users. When a template user is linked to a "real" user, the real user's preferences are inherited from the "template" user and the "real" user cannot change their preferences. Some implementations opt to work this way to enforce a standard look and feel for users in the same business area.

FASTPATH: Refer to [User — Portal Preferences](#) for more information about how users configure their zones.

Granting Access to Zones

An [application service](#) is associated with each zone. A user must be granted access rights to the respective application service in order to see a zone on a portal.

FASTPATH: Refer to [The Big Picture Of Application Security](#) for information about granting users access rights to an application service.

Please note the following with respect to zone application security:

- For most base product portals, all the zones for all the portals reference the same application service that is used to grant access to the main (stand-alone) portal for the page. In other words, if the user has access to the page, then he has access to all the zones on all portals for the page. There may be exceptions to this rule for certain portals.
- For a base product multi-query zones, typically the individual query zones and the multi-query zone reference the same application service that is used to grant access to the main (stand-alone) portal for the page. However, there may be individual query zones provided with a unique application service. This may occur when the query option is unusual and not applicable to all users or even to all implementations. If a user does not have security access to an individual query zone, that option will not be available in the dropdown.
- For base product portals that are configured to show on portal preferences, it is common that the portal contains different types of zones that may be applicable to different types of users. Typically these types of portals will deliver a unique application service for each zone so that an implementation may configure which user groups are allowed to view each zone. For these types of portals, please note the following:
 - A user's [Portal Preferences](#) page contains a row for a zone regardless of whether the user has access rights to the zone. Because of this, the system displays an indication of the user's access rights to each zone.
 - If a user's access rights to a zone are revoked, the zone will be suppressed when the user navigates to the respective portal.
 - Revoking a user's access rights does not change the user's [portal preferences](#) (i.e., a user can indicate they want to see a zone even if they don't have access to the zone - such a zone just won't appear when the respective portal appears).

NOTE: If you don't need to use zone security. When defining a zone, an application service is required. For zones that don't require special security, the product provides a "default" application service (F1-DFLT) that may be used. The expectation is that all user groups are granted access to this application service.

Common Characteristics of Stand-Alone Portals

The topics that follow describe additional characteristics specific to [stand-alone portals](#).

Putting Portals on Menus

A stand-alone portal should appear as a menu item on one of your menus. The following points provide how to do this:

- Every stand-alone portal has an associated navigation option. You can see a portal's navigation option on the [Portal - Main](#) page.
- To add a portal to a menu, you must add a [menu item](#) to the desired menu. This menu item must reference the portal's navigation option. There are two ways to add a menu item:
- If the portal's navigation option doesn't currently exist on a menu, you can press the **Add To Menu** button on the [Portal - Main](#) page. When you press this button, you will be prompted for the menu. The system will then create a menu item on this menu that references the portal's navigation option.
- You can always use the [Menu](#) page to add, change and delete menu items.

NOTE: No limit. A portal's navigation option can appear on any number of menu items (i.e., you can create several menu items that reference the same portal).

NOTE: Favorite links. Your users can set up their preferences to include the portal's navigation option on their [Favorite Links](#). This way, they can easily navigate to the portal without going through menus.

Granting Access to A Portal

An [application service](#) is associated with each [stand-alone portal](#). A user must be granted access rights to the respective application service in order to see a portal. Tab portals do not have separate security access. If a user has access to the main stand-alone portal, then the user will have security access to all its tabs. However, as mentioned in [Granting Access to Zones](#), in some scenarios, the individual zones on the portal may have different security access rights depending on the functionality.

FASTPATH: Refer to [The Big Picture Of Application Security](#) for information about granting users access rights to an application service.

NOTE: Automatically created. When you add a new stand-alone portal, the system automatically creates an application service behind the scenes. You'll need to know the name of this application service as this is what you use to grant access to the portal. The name of each stand-alone portal's application service is shown on the portal transaction.

Please note the following in respect of how application security impacts a user's portals:

- A user's [Portal Preferences](#) page only shows the portals configured to show on user preferences and where they have security access.
- The system's menus only show portals to which a user has security access.
- Users can set up favorite links to all portals, but they must have security rights to the portal's application service in order to invoke the favorite link.

Custom Look and Feel Options

The default look and feel of the application can be customized via feature configuration and cascading style sheets. The base product is provided with a [Custom Look And Feel Feature Configuration](#) type. You may want to set up a feature configuration of this type to define style sheet and UI Map help options.

User Interface

The base product allows for the conditional inclusion of customer styles into the system style set. The custom style may override any style provided by the base product. The style sheet may also include new styles for use in customer zone definitions. Use the [Style Sheet](#) option on the [Custom Look And Feel Feature Configuration](#) to define your custom style sheet.

NOTE: Some styles cannot change if they are part of the HTML code.

CAUTION: Implementers must ensure that the customized user interface is stable and scalable. Changing font, alignment padding, border size, and other user interface parameters may cause presentation problems, like scrollbars appearing or disappearing, cursors not working as expected, and unanticipated look and feel alterations of some layouts.

UI Map Help

A [tool tip](#) can be used to display additional help information to the user. This applies to section elements as well as individual elements on a map zone or UI Map. Refer to the tips context sensitive zone associated with the UI Map page for more information. The [Custom Look And Feel Feature Configuration](#) provides options to control the following:

- Whether UI Map Help functionality is turned on or off. By default it is turned on.

- Override the default help image with a custom image
- The location of the help image, either before or after the element.

FASTPATH: Refer to the feature configuration for a detailed description of each option.

Setting Up Portals and Zones

The topics in this section describe how to set up portals and zones. Please refer to [The Big Picture of Portals and Zones](#) for background information.

Defining Zone Types

A Zone Type represents a particular type of zone with a specific behavior. For example, a data explorer zone type is used to select data using a specific SQL statement and display the data based on parameter configuration. The zone type defines the Java Class that controls the behavior of the zone and defines the parameters that the Java Class supports. The base product supports many zone types used to build the portal / zone user interface. Implementations may introduce their own zone types.

NOTE: It is not very common for an implementation to introduce their own zone types.

Select **Admin > System > Zone Type** to maintain zone types.

Description of Page

Specify an easily recognizable **Zone Type** code and **Description**. Use the **Detailed Description** to describe in detail what the zone type does.

CAUTION: When adding new zone types, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

Owner indicates if this zone type is owned by the base product or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add a zone type. This information is display-only.

Java Class Name is the Java class responsible for building the zone using the parameters defined below.

Two types of parameters are specified when defining a zone type:

- Parameter values that have a **Usage** of **Zone** are defined on the zones and control the functionality of each zone governed by the zone type. A **Usage** value of **Zone - Override Allowed** indicates that an implementation may override the parameter value for a base zone.
- Parameter values that have a **Usage** of **Zone Type** are defined directly on the zone type and control how the zone type operates (e.g., the name of the XSL template, the name of the application service). A **Usage** value of **Zone Type - Override Allowed** indicates that an implementation may override the parameter value for a base zone type.

The following points describe the fields that are defined for each parameter:

- **Sequence** defines the relative position of the parameter.
- **Parameter Name** is the name of the parameter.
- **Description** is a short description that allows you to easily identify the purpose of the parameter.
- **Comments** contain information that you must know about the parameter or its implementation. For parameters with a usage of **Zone** or **Zone — Override Allowed**, this information is visible to the user when viewing or defining this parameter for a zone of this type.

- **Usage** indicates whether the parameter value is defined in a **Zone** of this type or in the **Zone Type**. **Zone - Override Allowed** and **Zone Type - Override Allowed** indicate that override values for the parameters defined in a base zone or base zone type can be entered.
- **Required** is checked to indicate that a zone must define a value for the parameter. It is not checked if a value for the parameter is optional. This field is protected if the **Usage** is **Zone Type** or **Zone Type - Override Allowed**.
- **Parameter Value** is used to define the value of zone type parameters. This field is protected if the **Usage** is **Zone** or **Zone - Override Allowed**.
- **Owner** indicates if this parameter is owned by the base product or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add a parameter. This information is display-only.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_ZONE_HDL](#).

Zone Type Parameter Comments

For the product owned zone type parameters, the parameter's detailed description provides the detail needed for properly configuring the parameter. For the Action parameters (IMPLEMENTOR_ACTION_n), the parameter description is abbreviated. Additional detail about configuring this parameter may be found in the [Zone Action Parameter](#) detailed information. The same details apply.

Defining Zones

The contents of this section describe how to maintain zones.

Zone - Main

Implementations may use the zone page to define custom zones. In addition, an implementation may override descriptions or some parameter values for base product zones.

Select **Admin > System > Zone** to create or maintain a zone.

Description of Page

Specify an easily recognizable **Zone** identifier and **Description**. Note that if this zone appears on a portal, this description acts as the zone title.

Override Description is provided if your implementation wishes to override the description of the value provided by the product.

CAUTION: Important! When introducing a new zone, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

Owner indicates if this zone is owned by the base product or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add a zone. This information is display-only.

Zone Type identifies the zone type that defines how the zone functions.

Application Service is the application service that is used to provide security for the zone. Refer to [Granting Access To Zones](#) for more information.

The **Width** defines if the zone occupies the **Full** width of the portal or only **Half**.

NOTE: Zones on the [dashboard portal](#) are always the width of the dashboard.

If the zone type supports help text, you can use **Zone Help Text** to describe the zone to the end-users. Note that for multi-query zones, if the multi-query zone has help text, that is displayed for any zone selected. If the multi-query zone does not

have help text, but the selected zone has help text, the selected zone's help text is displayed. Please refer to the section on [zone help text](#) for more information on how you can use HTML and cascading style sheets to format the help text.

Use **Override Zone Help Text** to override the product provided help text for this zone.

NOTE: Viewing Your Text. You can press the **Test** button to see how the help text will look when it's displayed in the zone.

The grid contains the zone's parameter values. The Zone Type controls the list of parameters. The grid contains the following fields:

- **Description** describes the parameter. This is display-only. Note that if there is a detailed description on the zone type parameter, a question mark icon appears next to the parameter's description. Click the icon to see details related to the parameter, including tips on how to populate the parameter value.

NOTE: Additional Details for Some Parameters. There are several parameter types that have a lot of detail related to the possible configuration that cannot easily fit into the detailed description. Refer to [Common Zone Parameters for Additional Information](#) about these parameters.

- **Parameter Value** is the value for the parameter.
- Use **Override Parameter Value** to override the existing value for this parameter. This field is enabled when the related zone type parameter value is **Zone - Override Allowed**, and the zone is owned by the base product.
- **Owner** indicates if this parameter is owned by the base product or by your implementation (**Customer Modification**). This information is display-only.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_ZONE](#).

Zone - Portal

Select **Admin > System > Zone** and navigate to the **Portal** tab to define the portals on which a zone appears.

Description of Page

The scroll area contains the portals on which the zone appears.

To add a zone to a portal, press the + button and specify the **Portal**.

NOTE: Owner indicates if this portal / zone relationship is owned by the base product or by your implementation (**Customer Modification**). This information is display-only.

NOTE: You can also add a zone to a portal using [Portal - Main](#). Additional configuration about how the zone appears on the portal is available only on the Portal.

Additional Zone Topics

The topics in this section provide additional information related to setting up your zones.

Zone Help Text

Most zone types support a button that allows a user to see zone-specific help text, which is defined on the zone page.

You can use HTML tags in the zone help text. The following is an example of help text that contains a variety of HTML tags:

This zone summarizes **revenue** in 4 periods:

The above would cause the word **revenue** to be bold and blue:

- `` and `` are the HTML tags used to indicate that the surrounded text should be bold
- `` and `` are the HTML tags used to indicate that the surrounded text should be blue.

The following are other useful HTML tags:

- `
` causes a line break in a text string. If you use `

` a blank line will appear.
- `<i>` causes the surrounded text to be italicized

Please refer to an HTML reference manual or website for more examples.

You can also use "spans" to customize the look of the contents of a text string. For example, your text string could be `revenue`. This would make the word "revenue" appear as large, bold, Courier text. Please refer to a Cascading Style Sheets (CSS) reference manual or website for more examples.

The following is an example of help text using a variety of HTML tags:

```
<font FACE="arial" size=2>
```

This zone summarizes **revenue** in 4 periods:

- The **1st period** is under your control. You simply select the desired **Period**, above *(you may need to click the down arrow to expose the filter section)*

- The **2nd period** is the period before the 1st period

- The **3rd period** is the same as the 1st period, but in the previous year

- The **4th period** is the period before the 3rd period

```
<br>
```

The traffic light's color is determined as follows:

- The ratio of the 1st and 3rd period is calculated

- If this value is between 80 and 100, **yellow** is shown

- If this value is < 80, **red** is shown

- If this value is > 100, **green** is shown

- If the value of the 3rd period is 0, no color is shown

```
</font>
```

NOTE: It is possible to associate tool tip help with individual HTML and UI map elements. For more information, see [UI Map Help](#).

Common Zone Parameters

For most zone parameters, the inline help for the parameter provides the detailed information needed for configuring the parameter values. For some parameters with very detailed descriptions, the inline help is abbreviated and more detail is provided here.

Column Parameters

Data explorer zones are used to select data to display using one or more SQL statements. For each SQL statement, the zone may configure up to 20 Columns that contain the formatting definition for displaying the output data. The parameters follow the description pattern of **Column x for SQL y**.

These parameters are applicable to the zone types

- Info Data Explorer - Multiple SQLs (**F1-DE**)

- Query Data Explorer - Multiple SQLs (**F1–DE-QUERY**)
- Info Data Explorer - Single SQL (**F1–DE-SINGLE**)

The following sections describe the various types of mnemonics.

Source Mnemonics

This table describe the mnemonics that control how the data in a column is derived.

Mnemonic	Description	Valid Values	Comments
source=	Defines how the column's value is derived.	SQLCOL	Indicates that the source of the column's value comes from a column in the SQL statement. This type of column must also reference the sqlcol= mnemonic.
		BO	Indicates that the source of the column's value comes from a business object . This type of column must also reference the bo= , input= and output= mnemonics to define how to interact with the business object.
		BS	Indicates that the source of the column's value comes from a business service . This type of column must also reference the bs= , input= and output= mnemonics to define how to interact with the business service.
		SS	Indicates that the source of the column's value comes from a service script . This type of column must also reference the ss= , input= and output= mnemonics to define how to interact with the service script.
		FORMULA	Indicates that the source of this column's value is calculated using a formula. This type of column must also reference the formula= mnemonic.
		SETFUNC	Indicates that the source of this column's value is calculated using a superset of values from the rows in the SQL statement. This type of column must also reference the setfunc= mnemonic.
		ICON	Indicates that the source of this column's value is a display icon reference (meaning that an icon will be displayed in the column). This type of column must also reference the icon= mnemonic to define the icon reference. NOTE: When using this source mnemonic, the formatting mnemonic type= is not applicable.
		FKREF	Indicates that the source of this column's value is an FK reference (meaning that the FK reference's context menu and information string will be displayed in the column). This type of column must also reference the fkref= and input= mnemonics to define how the FK reference is called. NOTE: When using this source mnemonic, the formatting mnemonic type= is not applicable.

Mnemonic	Description	Valid Values	Comments
		SPECIFIED	Indicates that the source of this column's value is specified by concatenating literals and other column values. This type of column must also reference the spec= mnemonic.
		MSG	Indicates that the source of this column is a message from the message table (along with any substitution variables). This type of column must also reference the msg= mnemonic.
sqlcol=	Defines the column in the SQL statement when source=SQLCOL .	COLUMN_NAME	Enter the name of a column that is retrieved in the SELECT statement. Note that if the select statement uses an alias for a column, then the alias should be referenced here.
		x	Where x is an integer value that references a column by its relative position in the SELECT statement. For example, sqlcol=3 would display the 3rd column in the SELECT statement).
bo=	Defines the business object to invoke when source=BO . This mnemonic must be used in conjunction with the input= and output= mnemonics to define how information is sent to / received from the business object.	Cx	This means business object code is defined in an earlier column. For example, define C1 if column 1 defines the business object.
		COLUMN_NAME	This means the business object was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause.
		'Business Object Code'	This means the business object is defined directly. For example 'F1-BundleImport'.
bs=	Defines the business service to invoke when source=BS . This mnemonic must be used in conjunction with the input= and output= mnemonics to define how information is sent to/received from the business service.	Cx	This means business service code is defined in an earlier column. For example, define C1 if column 1 defines the business service.
		COLUMN_NAME	This means the business service was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause.
		'Business Service Code'	> This means the business service is defined directly. For example 'F1-GetCountryStates'.
ss=	Defines the service script to invoke when source=SS . This mnemonic must be used in conjunction with the input= and output= mnemonics to define how information is sent to / received from the service script.	Cx	This means service script code is defined in an earlier column. For example, define C1 if column 1 defines the service script.
		COLUMN_NAME	This means the service script was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause.
		'Service Script Code'	This means the service script is defined directly. For example 'F1-ReturnBtn'.
formula=	Defines the formula to use when source=FORMULA . Examples: <ul style="list-style-type: none">• formula=C1*.90/C2• formula=(C1/C2)*100	The formula can contain numeric constants, operators and column references.	For column references, use the format Cx where x represents the column number. The following operators are supported: *, /, -, +, ABS, NEGATE, ROUND, FLOOR (round down), CEILING (round up), SIN, ASIN, COS, ACOS,

Mnemonic	Description	Valid Values	Comments
			TAN, ATAN, LOG, LOG10, EXP, EXP10, SQRT, **.
setfunc=	Defines the function to apply to the rows of a given column when source=SETFUNC .	function(Cx)	Where Cx represents a column whose rows should have the function applied and the function is one of the following <ul style="list-style-type: none"> • MAX. This derives the maximum value of all rows in the column. • MIN. This derives the minimum value of all rows in the column. • TOT. This derives the sum (total value) of all rows in the column. • ACC. This derives the cumulative total of all rows up to an including the current row.
input=	<p>This is used to define one or more input fields and values passed to business objects, business services, service scripts, and FK references.</p> <p>The syntax is as follows: [ELEMENT_NAME=ELEMENT_REF ELEMENT_NAME=ELEMENT_REF ...]</p> <p>In other words, the list of input values is surrounded by square brackets separated by a space. Each passed value first defines the ELEMENT_NAME, which is the name of the element / field in the target. ELEMENT_REF is the value passed in. The next column indicates the possible values for ELEMENT_REF.</p>	<p>Cx</p> <hr/> <p>COLUMN_NAME</p> <hr/> <p>'literal value'</p> <hr/> <p>userTimeZone</p>	<p>Where Cx represents the value of a previous column. If the value to pass is in the first column, reference C1.</p> <hr/> <p>This means the value to pass in was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause.</p> <hr/> <p>This means a literal value within the single quotes should be passed in.</p> <hr/> <p>This means the current user's time zone should be passed in. This is typically used with the business service F1-ShiftDateTime to convert data in the storage time zone to the user's time zone for display.</p>
		Examples:	
		<ul style="list-style-type: none"> • input=[USER_ID=C1] • input=[USER_ID=USER_ID] • input=[input/targetTimeZone=userTimeZone] 	
output=	This is used to define the name of the element retrieved from the business object, business service or service script used to populate this column.	elementName	Example: output=personInfo

Formatting Mnemonics

This table describe the mnemonics that control how a column is formatted.

Mnemonic	Description	Valid Values	Comments
type=	<p>Defines how the column's value is formatted.</p> <p>NOTE: Icon and Foreign Key columns. The source=source mnemonic may be used to indicate a column should be derived from an icon reference or a foreign key (FK) reference. If you use either of these sources, the type= mnemonic is not relevant as either an icon or a context menu / info string will appear in the column.</p>	<p>STRING</p> <hr/> <p>DATE</p> <hr/> <p>TIME</p> <hr/> <p>DATE/TIME</p> <hr/> <p>MONEY</p>	<p>Columns of this type capture a string. This is the default value.</p> <hr/> <p>Columns of this type capture a date.</p> <hr/> <p>Columns of this type capture a time (in database format) and will be displayed using the user's display profile.</p> <hr/> <p>Columns of this type capture a date and time (in database format) and will be displayed using the user's display profile.</p> <hr/> <p>Columns of this type capture a monetary field. This type of column may also reference the cur= mnemonic. If the cur mnemonic is not specified, the currency code on the installation record is used.</p>

Mnemonic	Description	Valid Values	Comments
		NUMBER	Columns of this type capture a numeric field. This type of column may also reference the dec= mnemonic.
label=	<p>Defines the column's override label. The label appears in the column's heading and in the zone's drag and drop area.</p> <p>If this mnemonic is not defined, the system uses the column's default label. The source of a column's default label differs depending on the column's source. Note that some sources don't have a default value and omitting this mnemonic will result in a blank label.</p>	<p>FIELD_NAME</p> <hr/> <p>'text'</p>	<p>Enter a valid field name whose label should be used for the column label. This should always be the option used if multiple languages are needed.</p> <hr/> <p>Defines the text directly.</p>
cur=	Defines the currency code applied when type=MONEY if the installation record's currency should not be used.	<p>Cx</p> <hr/> <p>COLUMN_NAME</p> <hr/> <p>'Currency Code'</p>	<p>This means currency code value is defined in an earlier column. For example, define C1 if column 1 defines the currency code.</p> <hr/> <p>This means the currency code was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause.</p> <hr/> <p>This means the currency code is defined directly. For example 'USD'.</p>
dec=	<p>Defines the number of decimal places when type=NUMBER.</p> <p>It is optional. If provided it should be an integer. If not provided, the number of decimals will default to the number of decimal places defined on the currency code specified on the installation record.</p>	nR	<p>Where n is the number of decimal places to show. Suffixing the number of decimal places with R means that the system should round up / down.</p> <p>Simply specifying n (without an R) means that decimal places should be truncated. For example, entering dec=4 will display 4 decimal places and truncate the remainder.</p> <p>NOTE: Formatting only. This mnemonic is only used for formatting, it does not impact the precision used for subsequent calculations. For example, if a column retrieved from the database contains 6 significant digits and dec=0, the column will be shown with no decimal places (truncated), however any references to the column in subsequent will use 6 decimal places. For example, if the column is referenced in a formula or set function, all 6 decimal places will be used.</p>
char=	This mnemonic applies special character(s) to the column's value.	'x[]x'	<p>Where x references the literal value to display and [] defines the relative position of the characters (before or after the value).</p> <p>You need only include the [] if you want to position characters in front of the value. For example, char='%' will place a percent sign after the value. If you want to position the word 'minutes' before a value, enter char='minutes []'. If you want to output a value like BUDGET \$123.12 (YTD), enter char='BUDGET [] (YTD)'.</p>
suppress=	This is used to indicate a column should not be displayed.	true	

Mnemonic	Description	Valid Values	Comments
	A column would be suppressed if it's only defined for use by subsequent columns, for example, if there is a formula that derives a column using two other columns. In this scenario, the columns referenced in the formula can be suppressed.	false	This is the default, meaning that you don't need to indicate suppress=false at all to indicate that the field should be shown.
suppressSearch=	This is used to indicate a column should not be displayed when the zone is invoked in search mode only.	true false	This is the default, meaning that you don't need to indicate suppressSearch=false at all to indicate that the field should be shown.
suppressExport=	This is used to indicate a column should not be downloaded to Excel.	true false	This is the default, meaning that you don't need to indicate suppressExport=false at all to indicate that the field should be included in a download.
width=	This is used to override the width of a column (number of pixels). The default value is the maximum width of any cell in the column.	n	Where n is a number between 0 and 999. NOTE: If there is no available breaking point in the data, the column will be longer than the specified number of pixels. The length of the column's label (which appears in the column's heading) may also make the width wider than specified.
color=	This is used to override the column's text color.	A valid HTML "named" color A valid RGB color model combination	For example color=red or color=yellow . For example color=#FF0000 or color=#CCCCCC . Note that the # is required.
bgcolor=	This is used to override the column's background color.	A valid HTML "named" color A valid RGB color model combination	Similar to the color= mnemonic. Similar to the color= mnemonic.
order=	Defines the column's default sort order.	ASC DESC	Indicates that the order is ascending. This is the default meaning that it is not necessary to indicate order=ASC . Indicates that the order is descending.

Click Mnemonics

This table describe the mnemonics that define whether a column value may be clicked and if so, what should happen.

Mnemonic	Description	Valid Values	Comments
navopt=	Defines the navigation option that references the target transaction or script when the user clicks a column. Note, this mnemonic should be used in conjunction with the context= mnemonic to define what information is sent to the navigation option's target transaction. This mnemonic is ignored if type=fkref because the FK reference code defines the hyperlink's destination.	Cx COLUMN_NAME 'Navigation Option Code'	This means navigation option code is defined in an earlier column. For example, define C1 if column 1 defines the navigation option. This means the navigation option was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause. Example: navopt=MAIN_PORTAL This means the navigation option code is defined directly. For example navopt='userMaint' .

Mnemonic	Description	Valid Values	Comments
context=	<p>This is used to define one or more context fields and values passed to the target navigation option to go along with the navopt= mnemonic.</p> <p>The syntax is as follows: [FIELD_NAME=FIELD_REF FIELD_NAME=FIELD_REF ...]</p> <p>In other words, the list of input values is surrounded by square brackets separated by a space. Each passed value first defines the FIELD_NAME, which is the name of the context field in the navigation option. FIELD_REF is the value passed in. The next column indicates the possible values for FIELD_REF.</p>	Cx	Where Cx represents the value of a previous column. For example, if the value to pass is in the first column, reference C1 .
		COLUMN_NAME	This means the value to pass in was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause.
		'literal value'	This means a literal value within the single quotes should be passed in.
bpa=	<p>Indicates that a BPA script should be executed with the user clicks the column and indicates the BPA to execute.</p> <p>Note, this mnemonic should be used in conjunction with the tempstorage= mnemonic to define the temporary storage values that will be initiated when the script is executed.</p> <p>This mnemonic is ignored if type=fkref because the FK reference code defines the hyperlink's destination.</p>	Cx	Indicates that the BPA script is defined in a previous column.
		COLUMN_NAME	This means the BPA script to execute was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause.
		'BPA Script Code'	This means that the BPA script to execute is defined directly.
tempstorage=	<p>This is used to define how temporary storage variables are initiated when the bpa= mnemonic is used.</p> <p>The syntax is as follows: [FIELD_NAME=FIELD_REF FIELD_NAME=FIELD_REF ...]</p> <p>In other words, the list of input values is surrounded by square brackets separated by a space. Each passed value first defines the FIELD_NAME, which is the name of the field in temporary storage. FIELD_REF is the value passed in. The next column indicates the possible values for FIELD_REF.</p>	Cx	Where Cx represents the value of a previous column. For example, if the value to pass is in the first column, reference C1 .
		COLUMN_NAME	This means the value to pass in was retrieved by the SELECT statement. The value should match the name defined in the SELECT clause.
		'literal value'	This means a literal value within the single quotes should be passed in.
list=	<p>This is used to enable work list capability for this column.</p> <p>You may optionally populate the listdesc= mnemonic to override the text that will be placed in the worklist zone.</p>	true	<p>Setting list=true will cause the work list icon to appear in the column's header. If a user clicks the column, it will populate all the rows in the output into the work list zone.</p> <p>NOTE: In the case of the zone type Info Data Explorer - Multiple SQLs (F1-DE), the output may be showing a union of the results of multiple SQL statements. In this case, if some of the SQL statements configure a given column with list=true, but not all, only the data in the cells for the statements that configure this mnemonic are put into the work list when the user clicks the icon.</p>
listdesc=	This is an optional mnemonic when using the list= mnemonic. It can	Cx	Where Cx represents the value of a previous column. For example, if

Mnemonic	Description	Valid Values	Comments
	be used to override the text that is placed in the work list zone.		the text to use is in the first column, reference C1 .
listbroadcast=	Indicates that the broadcast information for the column is also to be made available in the work list zone. This means that the work list can be used to broadcast information to a portal in the same manner as a data explorer.	true	Use this setting to turn on the feature.

Zone Action

Most zone types provided by the product allow for one or more Zone Actions to be defined to appear in the zone header. An action can appear as a hyperlink, icon or button. The action can also be provided as an HTML string.

NOTE: Zone types also include parameters for actions defined at the zone type level using **IMPLEMENTOR_ACTION_n** (Action n) parameters. These are rarely used by the product zone types. The actions defined here override any actions defined on the zone type (if present). The details below apply to the zone type level actions as well.

A zone action is defined using the following mnemonics:

Mnemonic	Description	Valid Values	Comments
type=	This mnemonic defines the appearance of the action in the zone header.	LINK	Indicates that the action is shown as a textual hyperlink.
		ICON	Indicates that the action is shown as a graphical icon.
		BUTTON	Indicates that the action is shown as an HTML button.
		ASIS	Indicates that the parameter will provide the HTML to be used for the action.
action=	This mnemonic defines the action to take when the link/icon/button is clicked. This is ignored when the type=ASIS .	NAVIGATION	Indicates that the action is navigation to a page.
		SCRIPT	Indicates that the action is to run a BPA script.
navopt=	Defines the navigation option to use when the action=NAVIGATION .	'NAV_OPT_CD'	Enter a reference to a valid navigation option in single quotes.
bpa=	Defines the script to run when the action=SCRIPT .	'SCRIPT_CD'	Enter a reference to a valid BPA script in single quotes.
icon=	Indicates the icon to use when type=ICON .	DISP_ICON_CD	Enter a reference to a valid display icon.
		'path'	Enter an explicit path to the icon, for example 'images/gotoZone.gif'.
asis=	This is required when the type=ASIS . This provides the ability to precisely define the HTML you wish to have included in the header. All valid HTML is permitted including the use of "ora" css classes and JavaScript functions.	['HTML']	
label=	By default, the label or tooltip will come from the navigation option or BPA script description. Use this mnemonic to override that label.	FIELD_NAME	Enter a valid field name whose label should be used. This should always be the option used if multiple languages are needed.
		'text'	Enter the text directly in single quotes.

Mnemonic	Description	Valid Values	Comments
context =[target1=source1 target2=source2]	<p>This is used to pass context data when navigating to a page or executing a BPA script. The mnemonic supports passing multiple values.</p> <p>In each case the target context field or BPA script variable is defined first followed by an equal sign, followed by source data defined using one of the valid values defined in the next column.</p> <p>One or more values may be defined. Each context value is defined separated by spaces. The whole set of context values should be surrounded by square brackets.</p>	<p>FIELD_NAME</p> <hr/> <p>xpath</p> <hr/> <p>'constant'</p>	<p>Indicates that the value should be taken from the field with this name from portal context, global context or the page data model. The mnemonic sourceLoc is used for defining the source.</p> <hr/> <p>Indicates that the value should be taken from a schema field, represented by the Xpath, displayed in this zone. This is valid when the zone is displaying a UI Map.</p> <hr/> <p>Indicates that the value defined in single quotes should be passed.</p>
sourceLoc =	<p>This mnemonic defines the source of the FIELD_NAME's value in the context mnemonic.</p> <p>If this mnemonic is left blank, the default behavior is as follows:</p> <ul style="list-style-type: none"> - The portal context is checked. - If no portal context value is found, the global context is checked. - If neither value is available, the field is ignored. 	<p>G</p> <hr/> <p>P</p> <hr/> <p>D</p>	<p>Indicates that the field's value is retrieved from the global context.</p> <hr/> <p>Indicates that the field's value is retrieved from the portal context.</p> <hr/> <p>Indicates that the field's value is retrieved from the page data model.</p>
class =	Use this mnemonic to override the look and feel of the link / icon / button using a different CSS style.	'className1' 'className2'	Enter one or more classes in single quotes. Multiple class names may be provided.
style =	Use this mnemonic to override the look and feel of the action element using the indicated css style.	Standard style= format.	All allowed css style definitions may be used.

Examples:

- **type=BUTTON action=SCRIPT bpa='F1-SET-USER' context=[USER_ID=USER_ID] label=UPDATE_LBL**
- **type=LINK action=NAVIGATION navopt='gotoUser' context=[USER_ID=path(schema/userId)]**
- **type=ASIS asis=['Search']**

NOTE: If the zone type has actions defined and there is a desire to simply remove the zone type actions, the Zone Action can be set with the following configuration: **type=ASIS asis=[]**

User Filters

Data explorer zones include the ability to define User filters to allow a user to enter data to restrict the zone's rows and / or columns. The filters may be defined individually using User Filter parameters 1–25. Alternatively, a UI map may be defined for capturing filters. In this case, the map's input fields must be associated with the zone's filters by specifying the **xpath**= mnemonic on the respective User Filter parameters.

These parameters are applicable to the zone types

- Info Data Explorer - Multiple SQLs (**F1-DE**)

- Query Data Explorer - Multiple SQLs (**F1-DE-QUERY**)
- Info Data Explorer - Single SQL (**F1-DE-SINGLE**)

A user filter is defined using the following mnemonics:

Mnemonic	Description	Valid Values	Comments
name=	This mnemonic is used if the zone's filter should be pre-populated with a value from global context, portal context or broadcast from another zone.	FIELD_NAME	
datasource=	This mnemonic defines the source of the filter's pre-populated value defined in the name mnemonic. If this mnemonic is left blank, the default behavior is as follows: - If the field has been broadcast from another zone, the broadcast value is used. - If no value is broadcast, the portal context is checked to determine if this field exists (if so, its value is taken). - If still no value, the global context is checked. - If still no value, no default value is shown.	G P D	Indicates that the zone should look for the filter value in global context. Indicates that the zone should look for the filter value in portal context. Indicates that the zone should look for the filter value in the page data model.
type=	Defines the visual metaphor used to capture the filter values.	DATE DATE/TIME STRING MONEY NUMBER LOOKUP TABLE CHARTYPE ASIS	Filters of this type capture a date. Filters of this type capture a date and time. Filters of this type capture a string Filters of this type capture a monetary field. This type of filter must also reference the cur mnemonic. Filters of this type capture a numeric field. This type of filter may also reference the decimals mnemonic. Filters of this type capture a lookup value. This type of filter must also reference the lookup mnemonic. Filters of this type capture an administrative table's value (code and description). This type of filter must also reference the table mnemonic. Filters of this type capture predefined characteristic values for a characteristic type (code and description). This type of filter must also reference the chartype mnemonic. Filters of this type capture a list of values to be referenced within an 'IN' clause within the SQL statement.
label=	Defines the filter's label that appears in the zone's description bar and in the input area.	FIELD_NAME 'text'	Enter a valid field name whose label should be used for the filter label. This should always be the option used if multiple languages are needed. Defines the text directly.
cur=	Defines the currency code applied when type=MONEY .	CURRENCY_CD	Enter a reference to a valid currency code.
dec=	Defines the number of decimal places when type=NUMBER .	N	It is optional. If provided it should be an integer. If not provided, the

Mnemonic	Description	Valid Values	Comments
			number of decimals will default to the number of decimal places defined on the currency code specified on the installation record.
lookup=	Defines the lookup flag whose values appear when type=LOOKUP .	LOOKUP_FIELD_NAME	Enter a reference to a valid lookup field name.
table=	Defines the admin table whose values appear when type=TABLE .	TABLE_NAME	Enter a reference to a valid admin table name.
chartype=	Defines the characteristic type code whose values appear when type=CHARTYPE .	CHAR_TYPE_CD	Enter a reference to a valid characteristic type code.
xpath=	This mnemonic is used in conjunction with a Filter Area UI Map. For each filter, you must specify the xpath to the corresponding UI map schema element.	nameFromUIMap	The type= mnemonic must also be appropriate for the map's input field, otherwise the query's SQL could fail.
likeable=	This mnemonic defines if a likeable search is performed on the entered value when type=STRING .	S	The query will add % to the suffix of the filter value.
		P	The query will add % to the prefix of the filter value.
		PS	The query will add % to the prefix and suffix of the filter value.
divide=	The mnemonic controls if a divider line appears above and/or below the filter. Note, you can specify this parameter twice if you want divider lines placed above and below a filter, e.g., divide=above divide=below .	above	This results in a divider line placed above the filter.
		below	This results in a divider line placed below the filter.
searchField=	This mnemonic controls the initial population of the filter when the zone is launched as a search from a UI map.	FIELD_NAME	Enter the field name that exactly matches the searchField name specified in the oraSearchField html element in the UI map.
encrypt=	This mnemonic defines if the user filter is encrypted and needs to be searched by hashed value.	[TBL_NAME,FLD_NAME,WHERE_FLD,WHERE_VALUE]	A valid table name and field name are required. The WHERE_FLD and WHERE_VALUE are optional, but if entered, both are required. Use this to only encrypt the field in FIELD_NAME if another field has a certain value. For example encrypt=[CI_PERSON,PER_ID_NBR,ID_TYPE_NBR,SSN].

Examples:

- **label=F1_NBR_DAYS type=NUMBER**
- **label=F1_SHOW_ALL_REQ_FLG type=LOOKUP lookup=F1_SHOW_ALL_REQ_FLG**
- Filter value where a Filter UI Map is defined and Description is one of the filters. **type=STRING xpath=description likeable=S**
 - **type=STRING label=DESCR likeable=S divide=below**
 - **label=REQ_TYPE_CD type=TABLE table=F1_REQ_TYPE**

Hidden Filters

Data explorer zones include the ability to define Hidden filters to restrict the rows and / or columns that appear in the zone. The following are the potential sources of a hidden filter's value:

- The global area contains the fields whose values are maintained in [global context](#).
- The portal area contains the fields describing the object currently displayed in a portal.

- Other zones on a portal can broadcast information to the portal area, which can then in turn be used by the zone as a hidden filter.

These parameters are applicable to the zone types

- Info Data Explorer - Multiple SQLs (**F1-DE**)
- Query Data Explorer - Multiple SQLs (**F1-DE-QUERY**)
- Info Data Explorer - Single SQL (**F1-DE-SINGLE**)

A hidden filter is defined using the following mnemonics:

Mnemonic	Description	Valid Values	Comments
name=	This mnemonic defines the name of the field that needs to be broadcast from other zones or populated in the portal context	FIELD_NAME	
datasource=	This mnemonic defines the source of the hidden filter's value. If this mnemonic is left blank, the default behavior is as follows: - If the field has been broadcast from another zone, the broadcast value is used. - If no value is broadcast, the portal context is checked to determine if this field exists (if so, its value is taken). - If still no value, the global context is checked. - If still no value, the zone appears as per the poprule mnemonic.	G P D	Indicates that the zone should look for the filter value in global context. Indicates that the zone should look for the filter value in portal context. Indicates that the zone should look for the filter value in the page data model.
poprule=	This mnemonic controls what happens if the hidden filter is not present.	R O	Indicates that a value for the filter is required. The zone will be set to the "empty state" and the "please broadcast" message will appear in the zone. This is the default value. Indicates that the value is optional. If no value is required, the zone is still built without that value.
type=	Defines the visual metaphor used to capture the filter values.	DATE DATE/TIME STRING MONEY NUMBER LOOKUP TABLE CHARTYPE	Filters of this type capture a date. Filters of this type capture a date and time. Filters of this type capture a string Filters of this type capture a monetary field. This type of filter must also reference the cur mnemonic. Filters of this type capture a numeric field. This type of filter may also reference the decimals mnemonic. Filters of this type capture a lookup value. This type of filter must also reference the lookup mnemonic. Filters of this type capture an administrative table's value (code and description). This type of filter must also reference the table mnemonic. Filters of this type capture predefined characteristic values for a characteristic type (code

Mnemonic	Description	Valid Values	Comments
			and description). This type of filter must also reference the chartype mnemonic.
		ASIS	Filters of this type capture a list of values to be referenced within an 'IN' clause within the SQL statement.
label=	Defines the filter's label that appears in the zone's description bar.	FIELD_NAME	Enter a valid field name whose label should be used. This should always be the option used if multiple languages are needed.
		'text'	Defines the text directly.
cur=	Defines the currency code applied when type=MONEY .	CURRENCY_CD	Enter a reference to a valid currency code.
dec=	Defines the number of decimal places when type=NUMBER .	N	It is optional. If provided it should be an integer. If not provided, the number of decimals will default to the number of decimal places defined on the currency code specified on the installation record.
lookup=	Defines the lookup flag whose values appear when type=LOOKUP .	LOOKUP_FIELD_NAME	Enter a reference to a valid lookup field name.
table=	Defines the admin table whose values appear when type=TABLE .	TABLE_NAME	Enter a reference to a valid admin table name.
chartype=	Defines the characteristic type code whose values appear when type=CHARTYPE .	CHAR_TYPE_CD	Enter a reference to a valid characteristic type code.
searchField=	This mnemonic controls the initial population of the filter when the zone is launched as a search from a UI map.	FIELD_NAME	Enter the field name that exactly matches the searchField name specified in the oraSearchField html element in the UI map.

Multi-Select Action

This parameter defines an action to be included in the action area for multi-selection processing. Note that a multi-selection action can only be used if the Multi Select parameter has been set to YES, which causes a checkbox to appear on each row displayed. The action defined here will trigger against all rows selected by the user via the checkbox.

These parameters are applicable to the zone types

- Info Data Explorer - Multiple SQLs (**F1-DE**)
- Query Data Explorer - Multiple SQLs (**F1-DE-QUERY**)
- Info Data Explorer - Single SQL (**F1-DE-SINGLE**)

A multi select action has the following mnemonics:

Mnemonic	Description	Valid Values	Comments
script=	This mnemonic defines the script to be invoked when the action is clicked. This is required.	SCR_CD	Enter a reference to a valid BPA script or Service Script.
type=	This mnemonic defines how the action should be rendered.	BUTTON	The action is rendered as a button. This is the default.
		LINK	The action is rendered as hypertext.
		ICON	The action is rendered as a graphic icon. For this option, the icon mnemonic is required.

Mnemonic	Description	Valid Values	Comments
icon=	This mnemonic defines the icon to display when type=ICON .	DISPLAY_ICON_CD	Enter a reference to a valid display icon code.
refresh=	This mnemonic indicates how and if a refresh should occur after the script completes.	NO	Indicates that no refresh is performed. This is the default.
		ZONE	Indicates that a refresh of the zone is performed.
		PORTAL	Indicates that a refresh of the entire portal is performed.
label=	By default, the button label, link text or icon tooltip will come from the script description. Use this mnemonic to override that label.	FIELD_NAME	Enter a valid field name whose label should be used. This should always be the option used if multiple languages are needed.
		'text'	Enter the text directly in single quotes.
list=	When executing the script, the framework builds an XML list containing information from each row selected. This list must be defined in the script's schema and referenced in this mnemonic.	listElementName	Enter a valid list element name from the script schema.
context=[elementName1=rowData1 elementName2=rowData2]	<p>This mnemonic is used to populate the list with the appropriate information from each selected row. The mnemonic supports passing multiple values.</p> <p>In each case the element in the schema list is defined first followed by an equal sign, followed by information about the data used to populate the element defined using one of the valid values defined in the next column.</p> <p>One or more values may be defined. Each context value is defined separated by spaces. The whole set of context values should be surrounded by square brackets.</p> <p>Example of a schema:</p> <pre><schema> <accountInfo type="list"> <accountId/ > <name /> <amount /> <process /></pre>	Cx	Indicates that the element should be populated with a value in the referenced column parameter.
		Px	Indicates that the element should be populated with a value in the referenced post processing parameter.
		COLUMN_NAME	Indicates that the element should be populated with a value from a column in the SQL statement.
		'constant'	Indicates that the value defined in single quotes should be passed.

Mnemonic	Description	Valid Values	Comments
	<pre></ accountInfo> </schema></pre> <p>Example of list and context mnemonics.</p> <p>list=accountInfo</p> <p>context=[accountId=ACCT_ ID name=C2 amount=P3 process='O']</p>		
class=	Use this mnemonic to override the look and feel of the action using a different CSS style.	'className1' 'className2'	Enter one or more classes in single quotes to be appended to the standard class(es). Multiple class names may be provided.
style=	Use this mnemonic to override the look and feel of the action element using the indicated css style.	Standard style= format.	All allowed css style definitions may be used.

Defining Context-Sensitive Zones

A context-sensitive zone allows you to associate a zone with a specific user-interface transaction. A context-sensitive zone appears at the top of the Dashboard when a user accesses a page for which the zone is specified as the context. For example, when viewing a business object, additional zones are visible that are specific to the business object page.

CAUTION: Make sure that the zone is appropriate for the transaction on which you are specifying it. For example, if your zone requires a business object ID as one of its keys, it should not be displayed on the To Do entry transaction.

Select **Admin > Context Sensitive Zone** to maintain context-sensitive zones.

Description of Page

The **Navigation Key** is a unique identifier of a tab page within the system. **Owner** indicates if this navigation key is owned by the base product or by your implementation (**Customer Modification**).

CAUTION: Important! When introducing a new context sensitive zone, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

The grid contains the list of context-sensitive zones and the sequence in which they appear in the dashboard for the selected navigation key. The grid contains the following fields:

- **Zone** is the name of the zone to display in the Dashboard.
- **Sequence** is the sequence in which the zone is displayed (if multiple context-sensitive zones are defined).
- **Owner** indicates if this context sensitive zone is owned by the base product or by your implementation (**Customer Modification**).

Where Used

A context-sensitive zone displays at the top of the Dashboard whenever a user accesses the transaction for with the zone is specified.

Defining Portals

This transaction is used to define / change portals. An implementation may define their own portals. In addition, an implementation may override some of the settings for base product provided portals.

Portal - Main

Navigate to this page using **Admin > System > Portal**.

Description of Page

Enter a meaningful and unique **Portal** code and **Description**. Please be aware that for [stand-alone portals](#), the Description is the portal's title (i.e., the end-users will see this title whenever they open the portal).

CAUTION: Important! When introducing a new portal, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

Owner indicates if this portal is owned by the base product or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add a portal. This information is display-only.

Type flag indicates whether the portal is a **Standalone Portal**, a **Tab Page Portal** or the **Dashboard**. Refer to [There Are Three Types of Portals](#) for more information.

The following fields are only enabled for **Standalone Portals**:

- **Navigation Option** defines the navigation option that is used to navigate to this portal from menus, scripts and your favorite links. The navigation option is automatically created when a **Standalone Portal** is added.
- You'll find an **Add To Menu** button adjacent. This field is only enabled if the navigation option is not referenced on a menu. When you click this button, a pop-up appears where you define a menu. If you subsequently press **OK**, a menu item is added to the selected menu. This menu item references the portal's navigation option. You can reposition the menu item on the menu by navigating to the [Menu page](#). Refer to [Putting Portals on Menus](#) for more information.
- **Application Services** defines the service used to secure this portal. The application service is automatically created when a **Standalone Portal** is added. Please note that only users with access to this application service will be able to view this portal and its zones. Refer to [Granting Access to A Portal](#) for more information.
- **Show on Portal Preferences** indicates if a user is allowed to have individual control of the zones on this portal. The portal will not appear in the accordion on the user's Portal Preferences page if this value is set to No. Note that an implementation may change this value for a product delivered portal.

The grid contains a list of zones that are available in the portal. Click + to add a new zone to the portal. Click - to remove a zone from the portal. The grid displays the following fields:

- **Zone** is the name of the zone as defined on the Zone page.
- **Description** is a description of the zone as defined on the Zone page.
- **Display** controls whether or not the zone is visible in the portal. For portals that are configured to Show on Portal Preferences, users may override this value for their view of the portal.
- **Initially Collapsed** controls whether or not the zone is initially collapsed in the portal. For portals that are configured to Show on Portal Preferences, users may override this value for their view of the portal.
- **Default Sequence** is the default sequence number for the zone within the portal. It does not need to be unique within the portal. Note that a sequence of zero will appear last, not first, in the portal. For portals that are configured to Show on Portal Preferences, users may override this value for their view of the portal.

- **Override Sequence** can be used by an implementation team to override the Default Sequence value that is set in the base product.
- **Refresh Seconds** defines in seconds how often the zone is automatically refreshed. The minimum valid value is 15. The maximum valid value is 3600 (1 hour). A value of 0 indicates no automatic refresh. Implementers can change this value as needed.
- **Owner** indicates if this portal / zone relationship is owned by the base product or by your implementation (**Customer Modification**). This information is display-only.

NOTE: Removing zones from a portal. You cannot remove a base product zone from a base product portal. An implementation may override the Display setting to prevent a zone from displaying on the portal. In addition, you cannot remove a zone if a user has enabled it on their Portal Preferences. To remove a zone from the portal list, first make sure that no user has it enabled in their portal preferences.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_PORTAL](#).

Portal - Options

Use this page to maintain a portal's options. Open this page using **Admin > System > Portal** and then navigate to the **Options** page.

Description of Page

The options grid allows you to configure the options that provide additional information related to the portal.

Select the **Option Type** dropdown to define its **Value**. **Detailed Description** may display additional information on the option type.

Set the **Sequence** to **1** unless the option can have more than one value.

Owner indicates if this is owned by the base product or by your implementation (**Customer Modification**).

NOTE: You can add new option types. Your implementation may want to add additional portal option types. To do that, add your new values to the customizable lookup field **PORTAL_OPT_FLG**.

Defining Display Icons

Icons are used to assist users in identifying different types of objects or instructions. A limited number of control tables allow administrative users to select an icon when they are configuring the system. Select **Admin > System > Display Icon Reference** to maintain the population of icons available for selection.

Description of Page

Each icon requires the following information:

- **Display Icon** is a code that uniquely identifies the icon.
- **Icon Type** defines how big the icon is (in pixels). The permissible values are: **30 x 21**, **21 x 21**, and **20 x 14**. Note that only icons that are **20 x 14** can be used on base product instructions.
- **Description** contains a brief description of the icon.
- **URL** describes where the icon is located. Your icons can be located on the product's web server or on an external web server.
- To add a new icon to the product web server, place it under the **/cm/images** directory under the **DefaultWebApp**. Then, in the **URL** field, specify the relative address of the icon. For example, if the icon's file name is **myIcon.gif**, the **URL** would be **/cm/images/myIcon.gif**.

- If the icon resides on an external web server, the **URL** must be fully qualified (for example, **http://myWebServer/images/myIcon.gif**).
- **Owner** indicates if this icon is owned by the base product or by your implementation (**Customer Modification**). This information is display-only.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_DISP_ICON](#).

Defining Navigation Keys

Each location to which a user can navigate (e.g., transactions, tab pages, tab menus, online help links, etc.) is identified by a navigation key. A navigation key is a logical identifier for a URL.

Navigation Key Types

There are two types of navigation keys:

- **System navigation keys** define locations where the URL is derived from other entities within the system. The items that are derived include the program location (i.e., physical location on the server), file name (i.e. program component name), and the file extension (e.g. COB). System navigation keys refer to program components, such as tab menus or tab pages.
- **External navigation keys** define locations simply as a URL. External URLs can be specified as relative to the product web server or fully qualified. External navigation keys always launch in a new instance of a browser window. Examples of external navigation keys include online help links and URLs to external systems.

Navigation Key vs. Navigation Option

The system has two entities that work in conjunction with each other to specify how navigation works:

- **Navigation Key** defines a unique location to which a user can navigate. For example, each page in the system has a unique navigation key. Navigation keys can also define locations that are "outside" of the system. For example, you can create a navigation key that references an external URL. Think of a navigation key as defining "where to go".
- **Navigation Option** defines how a page is opened when a user wants to navigate someplace. For example, you might have a navigation key that identifies a specific page. This navigation key can then be referenced on two navigation options; the first navigation option may allow users to navigate to the page in "add mode", while the second navigates to the page in "update mode".
- Please note that a wide variety of options can be defined on a navigation option. In addition to defining if a page is opened in add or update mode, you can define which tab is opened, which fields should be passed to the page, which search program is used, etc.

The Flexibility of Navigation Keys

Navigation keys provide a great deal of functionality to your users. Use navigation keys to:

- Allow users to navigate to new pages or search programs
- Allow users to transfer to an external system or web page. After setting up this data, your users may be able to access this external URL from a menu, a context menu, their favorite links, etc. Refer to [Linking to External Locations](#) for more information.

Refer to the Tool Suite Guide for more information on developing program components.

NOTE: Replacing Base-Package Pages or Searches. If your new page or search has been designed to replace a module in the base-package, the navigation key must indicate that it is [overriding an existing navigation key](#).

Linking to External Locations

If you want to include links to external systems or locations from within the system, you need to:

- Define a [navigation key](#) that specifies the URL of the location. For example, define an external navigation key that as a URL of <http://www.oracle.com/>.
- Define a [navigation option](#) that specifies from where in the system a user can go to your external location. For example, define a navigation option with a usage of **Favorites** or with a usage of **Menu**. Your navigation option points to the navigation key you defined above.
- Add your navigation option to the appropriate location within the system. For example, have users add the navigation option to their [Favorite Links](#) or add the navigation option as an item on a [menu](#).

Overriding Navigation Keys

Your implementation may choose to design a program component (e.g., a maintenance transaction or search page) to replace a component provided by the system. When doing this, the new navigation key must indicate that it is overriding the system navigation key. As a result, any menu entry or navigation options that reference this overridden navigation key automatically navigates to the custom component.

For example, if you have a custom On-line Batch Submission page and would like users to use this page rather than the one provided by the system, setting up an override navigation key ensures that if a user chooses to navigate to the On-line Batch Submission from Menu or a context menu, the user is brought to the custom On-line Batch Submission page.

To create an override navigation key, you need to:

- Define a [navigation key](#) using an appropriate [naming convention](#).
- If the URL Location of the navigation key being overridden is **External**, specify a URL Location of **Override (External)** and define the appropriate URL Override Location.
- If the URL Location of navigation key being overridden is **System**, specify a **URL Location of Override (System)** and populate the Program Component ID with your custom program component ID.
- Specify the navigation key that you are overriding in the **Overridden Navigation Key** field.

Refer to the Tool Suite Guide for more information about developing your own program components.

Maintaining Navigation Keys

Select **Admin > System > Navigation Key** to maintain navigation keys.

CAUTION: Important! When introducing a new navigation key, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

Description of Page

The **Navigation Key** is a unique name of the navigation key for internal use. Try to use a name that is easily recognizable.

Owner indicates if this navigation key is owned by the base product or by your implementation (**Customer Modification**). This information is display-only.

For **URL Location**, you can select from the following options:

- Use **External** to create a navigation key where the location is specified in the **URL Override** field.

- Use **Override (External)** to create a navigation key that overrides another external navigation key. If you use this option, you specify the name of the navigation key you are overriding in the **Overridden Navigation Key** field.
- Use **Override (System)** to point a system navigation key to a different location. If you use this option, you specify the name of the navigation key you are overriding in the **Overridden Navigation Key** field.
- Use **System** to create a navigation key for a custom program component developed for your implementation.

FASTPATH: Refer to [Navigation Key Types](#) for more information about system and external navigation keys.

FASTPATH: Refer to [Overriding Navigation Keys](#) for more information about settings required to override a system navigation key.

Program Component ID is the name of the program component identified by the key (for system navigation keys). The program component ID can also be used to specify the transaction with which an online help link is associated.

Overridden Navigation Key is the name of the navigation key that the current navigation key is overriding (if **Override (External)** or **Override (System)** is selected for the **URL Location**). Refer to [Overriding Navigation Keys](#) for more information.

URL Override is the specific URL for the navigation key (external navigation keys only). The URL can be relative to the product web server or fully qualified.

Open Window Options allows you to specify options (e.g., width and height) for opening a browser window for an external navigation key. (External navigation keys always launch in a new browser window.) You can use any valid features available in the `Window.open()` JavaScript method. The string should be formatted the same way that it would be for the features argument (e.g., **height=600,width=800,resizable=yes,scrollbars=yes,toolbar=no**). Refer to a JavaScript reference book for a complete list of available features.

Application Service is the application service that is used to secure access to transactions associated with **External** navigation keys. If a user has access to the specified application service, the user can navigate to the URL defined on the navigation key. Refer to [The Big Picture of Application Security](#) for more information.

The grid displays menu items that reference the navigation key (actually, it shows menu items that reference navigations options that, in turn, reference the navigation key).

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_MD_NAV](#).

Defining Navigation Options

Every time a user navigates to a transaction, the system retrieves a navigation option to determine which transaction should open. For example,

- A navigation option is associated with every menu item. When a user selects a menu item, the system retrieves the related navigation option to determine which transaction to open.
- A navigation option is associated with every [favorite link](#). When a user selects a favorite link, the system retrieves the related navigation option to determine which transaction to open.
- A navigation option is associated with every node in the various trees. When a user clicks a node in a tree, the system retrieves the related navigation option to determine which transaction to open.
- Etc.

Many navigation options are shipped with the base product and cannot be modified as these options support core functionality. As part of your implementation, you may add additional navigation options to support your specific business processes.

Navigation options may also be used to launch a BPA script.

The topics in this section describe how to maintain navigation options.

CAUTION: In order to improve response times, navigation options are cached the first time they are used after a web server is started. If you change a navigation option and you don't want to wait for the cache to rebuild, you must clear the cached information so it will be immediately rebuilt using current information. A special button has been provided on the Main tab of the navigation option transaction that performs this function. Please refer to [Caching Overview](#) for information on the various caches.

Navigation Option - Main

Select **Admin > System > Navigation Option** to maintain a navigation option.

Description of Page

Enter a unique **Navigation Option** code and **Description**.

CAUTION: When introducing a new navigation option, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

The **Flush System Login Info** button is used to flush the cached navigation options so you can use any modified navigation options. Refer to [Caching Overview](#) for more information.

Owner indicates if this navigation option is owned by the base product or by your implementation (**Customer Modification**). This field is display-only. The system sets the owner to **Customer Modification** when you add a navigation option.

NOTE: You may not change navigation options that are owned by the base product.

Use **Navigation Option Type** to define if the navigation option navigates to a **Transaction**, launches a **BPA Script** or opens an **Attachment**.

NOTE: The **Attachment** option type is only applicable to navigation options that are used to launch a file attached to a record in the Attachment maintenance object.

For navigation option types of **script**, indicate the **Script** to launch. You can use the **Context Fields** at the bottom of the page if you want to transfer the contents of specific fields to temporary storage variables available to the script. The script engine creates temporary storage variables with names that match the Context Field names.

For navigation option types of **transaction**, define the **Target Transaction** (navigation key) and optionally a specific **Tab Page** (also a navigation key) if a specific tab on the transaction (other than the Main tab) should be opened when navigating to that transaction.

NOTE: Finding transaction navigation keys. When populating the **Target Transaction** and **Tab Page** you are populating an appropriate navigation key. Because the system has a large number of transactions, we recommend using the "%" metaphor when you search for the transaction identifier. For example, if you want to find the currency maintenance transaction, enter "%currency" in the search criteria.

The additional information depends on whether the target transaction is a fixed page or a portal-based page:

- For portal-based pages:
 - **Navigation Mode** is not applicable and should just be set to **Add Mode**.
 - If navigating to a query portal, by default the query portal will open with the default search option defined. If the navigation should open a different search option, define the **Multi-Query Zone** for that query portal and indicate the **Sub-Query Zone** to open by default. Note that for this configuration, it is common to define **Context Fields** to pre-

populate search criteria in the target query zone. When using this configuration, be sure that the target query zone's user filters are defined to populate data from context.

- For fixed pages:
 - **Navigation Mode** indicates if the **Target Transaction** should be opened in **Add Mode** or **Change Mode**.
 - **Add Mode** should be used if the option is used to navigate to a transaction ready to add a new object. You can use the **Context Fields** at the bottom of the page if you want to transfer the contents of specific fields to the transaction when it opens.
 - **Change Mode** is only applicable for fixed pages and should be used if the option is used to navigate to a transaction ready to update an object. You have two ways to define the object to be changed:
 - Define the name of the fields that make up the unique identifier of the object in the **Context Fields** (and make sure to turn on **Key Field** for each such field).
 - Define the **Search Transaction** (navigation key) if you want to open a search window to retrieve an object before the target transaction opens. Select the appropriate **Search Type** to define which search method should be used. The options in the drop down correspond with the sections in the search (where **Main** is the first section, **Alternate** is the 2nd section, **Alternate 2** is the 3rd section, etc.). You should execute the search window in order to determine what each section does.

When you select a **Search Type**, define appropriate **Context Fields** so that the system will try to pre-populate the search transaction with these field values when the search first opens. Keep in mind that if a search is populated with field values the search is automatically triggered and, if only one object is found that matches the search criteria, it is selected and the search window closes.
 - **Search Group** is only visible if the **Development Tools** module is **not turned off**. It is used to define the correlation between fields on the search page and the tab page. You can view a tab page's **Search Groups** by viewing the HTML source and scanning for **allFieldPairs**.

The **Go To Tooltip** is used to specify the label associated with the tool tip that appears when hovering over a **Go To** object. Refer to the **Usage** grid below.

The **Usage** grid defines the objects on which this navigation option is used:

- Choose **Favorites** if the navigation option can be used as a **favorite link**.
- Choose **Menus** if the navigation option can be used as a user's **home page** or as a menu or context menu item.
- Choose **Script** if the navigation option can be used in a **script**.
- Choose **Foreign Key** if the navigation option can be used as a **foreign key reference**.
- Choose **Go To** if the navigation option can be used as a "go to" destination ("go to" destinations are used on Go To buttons, tree nodes, and hyperlinks).
- If your product supports marketing campaigns, you can choose **Campaign** if the navigation option can be used as a "post completion" transaction on a campaign. For more information refer to that product's documentation for campaigns.

The **Context Fields** grid contains the names of the fields whose contents will be passed to the **Target Transaction** or **Script** or used to launch an **Attachment**. The system retrieves the values of these fields from the "current" page and transfers them to the target transaction or to the script's temporary storage. Turn on **Key Field** for each context field that makes up the unique identifier when navigating to a transaction in **Change Mode**.

NOTE: For an **Attachment**, the grid should contain the Attachment ID.

NOTE: Navigating from a Menu. The standard followed for many base main **menu** navigation options for fixed transactions is that navigation options launched from the Menu dropdown are configured with no context.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CL_NAV_OPT](#).

Navigation Option - Tree

This page contains a tree that shows how a navigation option is used. Select **Admin > System > Navigation Option** and navigate to the **Tree** tab to view this page.

Description of Page

The tree shows every menu item, favorite link, and tree node that references the navigation option. This information is provided to make you aware of the ramifications of changing a navigation option.

Database Tools

This section describes a variety of database tools that are supplied with the your product.

Defining Table Options

The topics in this section describe the transaction that allows you to define metadata for the application's tables.

Table - Main

Navigate using **Admin > Database > Table**.

Description of Page

Table Name is the identifier of this table.

Description contains a brief description of the table.

System Table defines if the table includes rows that are owned by the base product.

Enable Referential Integrity defines if the system performs referential integrity validation when rows in this table are deleted.

Data Group ID is used for internal purposes.

Table Usage is not in use at this time.

Table Type defines if the table is an **External Table**, a **View** or a physical **Table**.

Date / Time Data Type defines if the system shows times on this table in **Local Legal Time** or in **Local Standard Time**. Local Legal Time is the time as adjusted for daylight savings / summer time.

Table Classification Type specifies the category of data the table will hold. This is for information purposes only and is not used by any system processing. Valid values are **Admin System Table**, **Admin Non System Table**, **Master Table**, **Transaction Table**, and **Unclassified**.

Table Volume Type specifies the expected amount of data the table will hold. This is for information purposes only and is not used by any system processing. Valid values are **High Volume**, **Low Volume**, **Medium Volume**, and **Unclassified**. The volume of data in a particular table in the system may differ greatly from one implementation to another based on unique business requirements. The values populated for base product tables are set to volumes that are typical but may not be true for a given implementation. The value may be updated to reflect the situation for a given implementation.

Audit Table is the name of the table on which this table's audit logs are stored. Refer to [The Audit Trail File](#) for more information.

Use **Audit Program Type** to define if the audit program is written in **Java** or **Java (Converted)**, meaning it was converted into Java.

NOTE: Java (Converted) program types are not applicable to all products.

Audit Program is the name of the program that is executed to store an audit log. Refer to [Turn On Auditing For a Table](#) for more information.

NOTE: View the source. If the program is shipped with the base package, you can use the adjacent button to display the source code of this program in the [Java docs viewer](#).

Upgrade controls what happens to the rows in this table when the system is upgraded to a new release:

- **Keep** means that the rows on this table are not touched during an upgrade
- **Merge** means that the rows on this table are merged with rows owned by the base product
- **Refresh** means that the rows on this table are deleted and refreshed with rows owned by the base product.

Data Conversion Role controls if / how the table is used by the conversion tool:

- **Convert (Retain PK)** means that the table's rows are populated from the conversion schema and the prime key in the conversion schema is used when the rows are converted. A new key is not assigned by the system.
- **Convert (New PK)** means that the table's rows are populated from the conversion schema and the prime key is reassigned by the system during conversion.
- **Not Converted** means that the table's rows are not managed by the conversion tool.
- **View of Production** means that the conversion tool uses a view of the table in production when accessing the rows in the table. This is commonly used for administrative tables.

A **Language Table** is specified when fields containing descriptions are kept in a child table. The child table keeps a separate record for each language for which a description is translated.

Enable Data Dictionary defines if the table is to be included in the [Data Dictionary](#) application viewer.

A **Key Table** is specified when the prime-key is assigned by the system. This table holds the identity of the prime keys allocated to both live and archived rows.

Type of Key specifies how prime key values are generated when records are added to the table:

- **Other** means a foreign-system allocates the table's prime-key.
- **Sequential** means a sequence number is incremented whenever a record is added to the table. The next number in the sequence determines the key value.
- **System-generated** means a program generates a random key for the record when it is added. If the record's table is the child of another table, it may inherit a portion of the random number from its parent's key.
- **User-defined** means the user specifies the key when a record is added.

Inherited Key Prefix Length defines the number of most significant digits used from a parent record's primary key value to be used as the prefix for a child record's key value. This is only specified when the Type of Key is **System-generated** and the high-order values of the table's key is inherited from the parent table.

Java Table Name. This field is used to identify the entity/Java class name of the class that represents the table in the Java code. It should contain a short "camelCased" name to be used as the name of the entity within the system. It must also be a valid Java name, and must be unique across the system. This name is used as follows:

- As the short class name on all classes in the Java hierarchy for the class: the Impl class, the Gen, and the interface.
- In HQL queries, it is used to identify the hibernate entity being selected.

Caching Regime determines if the table's values should be cached when they are accessed by a batch process. The default value is **Not Cached**. You should select **Cached for Batch** if you know the values in the table will not change during the course of a batch job. For example, currency codes will not change during a batch process. Caching a table's values will reduce unnecessary SQL calls and improve performance.

Key Validation determines if and when keys are checked for uniqueness. The default value is **Always Check Uniqueness**. Select **Check Uniqueness Online Only** when the database constructs the keys in the table, such as in log tables. Select **Never Perform Uniqueness Checking** when you know that the database constructs the keys in the table and that users cannot add rows directly to the table, such as in log parameter tables. This will reduce unnecessary SQL calls and improve performance.

Help URL is the link to the user documentation that describes this table.

Help Text contains additional information about the table.

Extract for Translation is only visible in a development environment. It indicates whether or not the table includes strings that are eligible for product translation.

Translation Context is only visible in a development environment. It is used to provide information to a translator about the nature and purpose of rows in the table.

NOTE: Changes to base owned tables. Only the following attributes of tables that are owned by the product are modifiable: Audit Table, Audit Program Type, Audit Program, Caching Regime, Key Validation and Table Volume Type.

The grid contains an entry for every field on the table. Drilling down on the field takes you to the [Table Field](#) tab where you may modify certain attributes. The following fields may also be modified from the grid: **Description**, **Override Label**, **Audit Delete**, **Audit Insert** and **Audit Update**. Refer to the Table Field tab for descriptions of these fields.

Table - Table Field

Navigate using **Admin > Database > Table** and click the **Table Field** tab.

Description of Page

Field Name is the name of the field. It is followed by its **Java Field Name**.

Field Help Text displays the help text listed for this field on the Field page, if populated.

Nullable, **Required** and **Validate** are for internal use.

Turn on **Audit Delete** if an audit record should be stored for this field when a row is deleted. Refer to [How To Enable Auditing](#) for more information.

Turn on **Audit Insert** if an audit record should be stored for this field when a row is added. Refer to [How To Enable Auditing](#) for more information.

Turn on **Audit Update** if an audit record should be stored for this field when it is changed. Refer to [How To Enable Auditing](#) for more information.

The **Allow Customization** is only applicable if the table is an Admin System Table. It indicates fields that an implementation is allowed to change for a base owned record. Changes to the field value of one of these types of fields by an implementation are maintained when upgrading to a new version of the product.

Standard Time Type is only enabled if the Table indicates that the Date/Time Data Type is **Local Standard Time**. Each field that represent date/time should define if it uses **Logical Standard Time**, **Physical Standard Time** or uses a **Referenced Time Zone**.

Sequence is a unique sequence of this field with respect to other fields on the table.

The **Label** column is used to define a special label for this field when it relates to this table if it should be different from the field's label. Note that this only impacts the label on a fixed page user interface. Labels on portal based user interfaces do not use this information.

The **Override Label** is provided if an implementation wants to override the base-product label.

NOTE: If you want the Override Label to be shown in the [data dictionary](#), you must regenerate the data dictionary.

Help Text contains any additional information about the field with respect to its use on this table.

Field Usage is not used at this time.

Extract for Translation is only visible in a development environment. For tables marked to extract for translation, each translatable field on the table should indicate **Yes**.

Translation Context is only visible in a development environment. It is used to provide information to a translator about the nature and purpose of the data in this field on this table.

NOTE: Changes to base owned table / fields. Only the following attributes of table / fields that are owned by the product are modifiable: Audit Delete, Audit Insert, Audit Update, Override Label.

Table - Constraints

Select **Admin > Database > Table** and navigate to the **Constraints** tab to view the constraints defined on the table.

Description of Page

The fields on this page are protected as only the product development group may change them.

This page represents a collection of constraints defined for the table. A constraint is a field (or set of fields) that represents the unique identifier of a given record stored in the table or a field (or set of fields) that represents a given record's relationship to another record in the system.

Constraint ID is a unique identifier of the constraint.

Owner indicates if this is owned by the base package or by your implementation (**Customer Modification**)

Constraint Type Flag defines how the constraint is used in the system:

- **Primary Key** represents the field or set of fields that represent the unique identifier of a record stored in a table.
- **Logical Key** represents an alternate unique identifier of a record based on a different set of fields than the Primary key.
- **Foreign Key** represents a field or set of fields that specifies identifying and non-identifying relationships to other tables in the application. A foreign key constraint references the primary key constraint of another table.
- **Conditional Foreign Key** represents rare relationships between tables where a single field (or set of fields) may reference multiple primary key constraints of other tables within the application as a foreign key.

When **Enable Referential Integrity** is checked, the system validates the integrity of the constraint when a row in the table is modified.

Referring Constraint Owner indicates if this is owned by the base package or by your implementation (**Customer Modification**).

Referring Constraint ID is the **Primary Key** constraint of another table whose records are referenced by records stored in this table.

Referring Constraint Table displays the table on which the Referring Constraint ID is defined. You can use the adjacent go-to button to open the table.

Additional Conditional SQL Text is only specified when the constraint is a **Conditional Foreign Key**. The SQL represents the condition under which the foreign key represents a relationship to the referring constraint table.

NOTE: Additional Conditional SQL Syntax. When specifying additional conditional SQL text, all table names are prefixed with a pound (#) sign.

The Constraint Field grid at the bottom of the page is for maintaining the field or set of fields that make up this constraint.

Field is the name of the table's field that is a component of the constraint.

Sequence The rank of the field as a component of the constraint.

The Referring Constraint Field grid at the bottom of the page displays the field or set of fields that make up the **Primary key** constraint of the referring constraint.

Field is the name of the table's field that is a component of the referring constraint.

Sequence is the rank of the field as a component of the referring constraint.

Table - Referred by Constraints

Select **Admin > Database > Table** and navigate to the **Referred By Constraints** tab to view the constraints defined on other tables that reference the **Primary Key** constraint of this table.

Description of Page

This page is used to display the collection of constraints defined on other tables that reference the table.

Referred By Constraint Id is the unique identifier of the constraint defined on another table.

Referred By Constraint Owner indicates if this constraint is owned by the base package or by your implementation (**Customer Modification**).

Prime Key Constraint Id is the **Primary Key** constraint of the current table.

Prime Key Owner indicates if this prime key is owned by the base package or by your implementation (**Customer Modification**).

Referred By Constraint Table is the table on which Referred By Constraint Ids are defined.

When **Enable Referential Integrity** is checked, the system validates the integrity of the constraint when a row in the table is modified.

The grid at the bottom of the page displays the **Field** and **Sequence** for the fields that make up the constraint defined on the other table.

Defining Field Options

The topics in this section describe the transaction that can be used to view information about a field and to change the name of a field on the various pages in the system.

Field - Main

Open this page using **Admin > Database > Field**.

Description of Page

Field Name uniquely identifies this field.

CAUTION: As described in [System Data Naming Convention](#) for most system data tables, the base product follows a specific naming convention. However, this is not true for the Field table. If you introduce new fields, you must prefix the field with **CM**. If you do not do this, there is a possibility that a future release of the application could introduce a new field with the name you allocated.

Owner indicates if this field is owned by the base package or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add a field.

Base Field indicates that the field inherits some of its definitions from another field.

Data Type indicates if the type of data the field will hold. Valid values are **Character**, **Character Large Object**, **Date**, **DateTime**, **Number**, **Time**, **Varchar2** and **XML Type**. This field is protected if the field refers to a Base Field.

Ext Data Type or Extended Data Type is used to further define the type of data for certain data types. Valid values are **Currency Source, Day of Month, Duration, Money, Month of Year, Flag and Switch**. This field is protected if the field refers to a Base Field.

Precision defines the length of the field. In the case of variable length fields, it is the maximum length possible. For number fields that include decimal values, the precision includes the decimal values. This field is protected if the field refers to a Base Field.

Scale is only applicable for number fields. It indicates the number of decimal places supported by the field. This field is protected if the field refers to a Base Field.

Sign is only applicable for numbers. It indicates if the data may contain positive or negative numbers.

Value List is only visible for products that had at one point included COBOL. It defines the copybook that includes the list of valid values for the field.

Description contains the label of the field. This is the label of the field that appears on the various pages on which the field is displayed. Note, the field's label can be overridden for a specific table by specifying an Override Label on the [table / field](#) information. However, this override is not used in portal based user interfaces. It is only applicable if the field is displayed on fixed page user interfaces.

Java Field Name is the reference to this field used in Java code.

Override Label is used if an implementation would like to override the label that appear on user interfaces in the system.

CAUTION: For fixed pages, if the field's label is overridden for a specific table, that override takes precedence. In this is the case the override on the [table / field](#) page should be used.

Work Field indicates that the field does not represent a database table column.

Help Text is used to provide field level embedded help to this field. If the field is displayed on a user interface that supports display of embedded help, this text may be displayed.

Use **Override Help Text** to override the existing embedded help text for this field.

Extract for Translation is only visible in a development environment. It indicates whether or not the field and its description should be included in an extract of translatable strings when doing a product translation. This flag may be set to "No" for work fields whose label is not visible to a user on any user interface.

Translation Context is only visible in a development environment. It is used to provide information to a translator about the use of the field's label so that an appropriate translation can be provided.

Field - Tables Using Field

Select **Admin > Database > Field** and navigate to the **Tables Using Field** tab to view the tables that contain a field.

Description of Page

The grid on this page contains the **Tables** that reference the **Field**. You can use the adjacent go to button to open the [Table Maintenance](#) transaction.

Defining Maintenance Object Options

A maintenance object defines the configuration of a given "entity" in the system. It includes the definition of the tables that together capture the physical data for the entity. In addition, the maintenance object includes options that define important information related to the maintenance object that may be accessed for logic throughout the system. Several algorithm plug-in spots are also defined on the maintenance object, allowing for business rules that govern all records for this maintenance object.

Many maintenance objects in the system support the use of business objects to further define configuration and business rules for a given record. Refer to [The Big Picture of Business Objects](#) for more information.

Maintenance Object - Main

Select **Admin > Database > Maintenance Object** to view information about a maintenance object.

Description of Page

Most maintenance objects are provided with the base package. An implementation can introduce custom maintenance objects when needed. Most fields may not be changed if owned by the base package.

Enter a unique **Maintenance Object** name and **Description**. **Owner** indicates if this business object is owned by the base package or by your implementation (**Customer Modification**).

IMPORTANT: If you introduce a new maintenance object, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

Service Name is the name of the internal service associated with the maintenance object.

Click the **View XML** hyperlink to view the XML document associated with the maintenance object service in the [Service XML Viewer](#).

Click the **View MO** hyperlink to view the definition of the maintenance object in the [Maintenance Object Viewer](#).

The grid displays the following for each table defined under the maintenance object:

- **Table** is the name of a given table maintained as part of the maintenance object.
- **Table Role** defines the table's place in the maintenance object hierarchy. Only one **Primary** table may be specified within a maintenance object, but the maintenance object may contain many **Child** tables.
- **Parent Constraint ID** specifies the [constraint](#) used to link the table to its parent table within the maintenance object table hierarchy.
- **Owner** indicates if this is owned by the base package or by your implementation (**Customer Modification**).

Maintenance Object - Options

Use this page to maintain a maintenance object's options. Open this page using **Admin > Database > Maintenance Object** and then navigate to the **Options** tab.

Description of Page

The options grid allows you to configure the maintenance object to support extensible options.

Select the **Option Type** drop-down to define its **Value**. **Detailed Description** may display additional information on the option type.

Set the **Sequence** to **1** unless the option can have more than one value.

Owner indicates if this is owned by the base package or by your implementation (**Customer Modification**).

NOTE: You can add new option types. Your implementation may want to add additional maintenance option types. For example, your implementation may have plug-in driven logic that would benefit from a new type of option. To do that, add your new values to the customizable lookup field **MAINT_OBJ_OPT_FLG**.

Maintenance Object - Algorithms

Use this page to maintain a maintenance object's algorithms. Open this page using **Admin > Database > Maintenance Object** and then navigate to the **Algorithms** tab.

Description of Page

The **Algorithms** grid contains algorithms that control important functions for instances of this maintenance object. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).
- Specify the **Sequence Number** and **Algorithm** for each system event. You can set the **Sequence Number** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.
- If the algorithm is implemented as a script, a link to the **Script** is provided. Refer to [Plug-in Scripts](#) for more information.
- **Owner** indicates if this is owned by the base package or by your implementation (**Customer Modification**).

The following table describes each **System Event**.

System Event	Optional / Required	Description
Audit	Optional	<p>Algorithms of this type are called to notify of any changes to the maintenance object's set of tables.</p> <p>Click here to see the algorithm types available for this system event.</p>
Determine BO	Optional	<p>Algorithm of this type is used to determine the Business Object associated with an instance of the maintenance object. It is necessary to plug in such an algorithm on a Maintenance Object to enable the business object rules functionality.</p> <p>The system invokes a single algorithm of this type. If more than one algorithm is plugged-in the system invokes the one with the greatest sequence number.</p> <p>Click here to see the algorithm types available for this system event.</p>
ILM Eligibility	Optional	<p>Algorithms of this type are used for maintenance objects that are enabled for Information Lifecycle Management. They are used to review records that have reached the maximum retention days and evaluate if they are ready to be archived.</p> <p>The system invokes a single algorithm of this type. If more than one algorithm is plugged-in the system invokes the one with the greatest sequence number.</p> <p>Click here to see the algorithm types available for this system event.</p>
Information	Optional	<p>We use the term "Maintenance Object Information" to describe the basic information that appears throughout the system to describe an instance of the maintenance object. The data that appears in this information description is constructed using this algorithm.</p> <p>The system invokes a single algorithm of this type. If more than one algorithm is plugged-in the system invokes the one with the greatest sequence number.</p> <p>Click here to see the algorithm types available for this system event.</p>
Revision Control	Optional	<p>An algorithm of this type is used to enforce revision control rules when an object is added, changed or deleted. The maintenance object service calls the plug-in once before the object is processed and once more after</p>

System Event	Optional / Required	Description
		<p>applying all business object rules. This allows revision rules to take place in proper revision timings.</p> <p>Click here to see the algorithm types available for this system event.</p>
Transition	Optional	<p>The system calls algorithms of this type upon each successful state transition of a business object as well as when it is first created. These are typically used to record the transition on the maintenance object's log.</p> <p>Note that most base maintenance objects are already shipped with an automatic logging of state transitions. In this case you may use these algorithms to override the base logging functionality with your own. Refer to State Transitions are Audited for more information.</p> <p>Click here to see the algorithm types available for this system event.</p>
Transition Error	Optional	<p>The system calls this type of algorithm when a state transition fails and the business object should be saved in its latest successful state. The algorithm is responsible for logging the transition error somewhere, typically on the maintenance object's log.</p> <p>Notice that in this case, the caller does NOT get an error back but rather the call ends successfully and the exception is recorded somewhere, as per the plug-in logic.</p> <p>The system invokes a single algorithm of this type. If more than one algorithm is plugged-in the system invokes the one with the greatest sequence number.</p> <p>Click here to see the algorithm types available for this system event.</p>

NOTE: You can inactivate algorithms on Maintenance Objects. Your implementation may want to inactivate one or more algorithms plugged into the base maintenance object. To do that, go to the options grid on Maintenance Object - Options and add a new option, setting the option type to **Inactive Algorithm** and setting the option value to the algorithm code.

Maintenance Object - Maintenance Object Tree

You can navigate to the **Maintenance Object Tree** to see an overview of the tables and table relationships associated with the maintenance objects.

Description of Page

This page is dedicated to a [tree](#) that shows the maintenance object's tables as well as [business objects](#), if you have defined any. You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

Defining Valid Values

The product provides several options for defining valid values for a column on a table:

- Lookup
- Extendable Lookup
- Control Table

The following provides more information about the functionality of each of the options available for defining valid values for a column.

Lookup

The simplest mechanism for defining valid values for a column on a table is via the Lookup table. This is sometimes referred to as a “simple” lookup to distinguish it from an extendable lookup (described below). Using the lookup table, you can define valid values and their descriptions. When choosing a valid value that is defined by a lookup, a dropdown UI metaphor is used.

The following highlights functionality related to lookups:

- Lookups are associated with a [Field](#). The field is defined as a character data type with an extended data type of **Flag**. The field’s label serves as the description for the prompt to select the valid value.
- The lookup code is limited to four characters and must be all uppercase. If there is any functionality where a valid value in the application must match valid values in an external system, the lookup table may not be the appropriate choice.
- The lookup table does not support additional attributes to be defined for each value. This option is only appropriate when a simple code and description pair is needed.
- The product may also use Lookups to define valid values for functionality unrelated to a column on a table. For example, an algorithm plug-in spot may define an input parameter that supports one or more valid values. The plug-in spot may define the valid values using a lookup, allowing for a simple way to validate the value supplied when invoking the algorithm and to document the valid values.

FASTPATH: For more information, refer to [Defining Lookup Options](#).

Extendable Lookup

The extendable lookup provides a way of defining valid values for a column with additional capabilities that are not supported using the Lookup table. When choosing a valid value that is defined by an extendable lookup, a dropdown UI metaphor is used.

The following highlights functionality related to extendable lookups:

- Each Extendable Lookups is defined using a business object.
- A field should be defined for the extendable lookup code. The field defines the label for the lookup code and defines the size of the lookup code. The size is determined based on the business use case. In addition, there are standard fields included in all extendable lookups, including a description, detailed description and an override description (so that implementations can override the description of base delivered values).
- The extendable lookup may define additional information for each value if warranted by the business requirement. See [Additional Attributes](#) for technical information about additional attributes.

FASTPATH: For more information, refer to [Defining Extendable Lookups](#).

Control Table

There may be scenarios where a list of valid values warrants a standalone maintenance object, which is considered an administrative or control table object. When choosing a valid value that is defined by a control, either a dropdown UI metaphor or a search metaphor is used, depending on how it has been designed.

The following points highlight some reasons why this option may be chosen:

- The records require a lifecycle such that BO status is warranted.
- The additional attributes are sophisticated enough that they warrant their own column definition rather than relying on using CLOB or flattened characteristic. For example, if a list of information needs to be captured with several attributes in the list and the information in the list needs to be searchable.

In this situation, if a product has provided a control table for this type of functionality, it will be documented fully in the appropriate functional area. If an implementation determines that a custom control table is warranted, all the standard functionality for a maintenance object is required: database tables, maintenance object metadata, appropriate Java maintenance classes, portals, zones, etc. Refer to the Software Development Kit for more information. No further information is provided in this section for this option.

Defining Lookup Options

Lookup fields may be used to define valid values for a column in a table or for other types of values like parameters to an algorithm.

FASTPATH: Refer to [Defining Valid Values](#) for some background information.

The base product provides many different lookup fields and their values as part of the product. The following points highlight some functionality related to base-package lookups.

- Fields that are owned by the product will typically provide base lookup values. Implementations are not permitted to remove base delivered lookup values. Implementations may be able to add custom values to base owned lookups. This is controlled with the Custom switch on lookup.
- When the custom switch is unchecked, it means that there is functionality controlled by the base values and an implementation may not extend or customize this functionality. An example of this type of lookup is the Data Type field on the [Field](#) table. The system supports a distinct list of data types and an implementation may not add additional values.
- When the custom switch is checked, it means that there is base functionality supplied for the base values but that an implementation can extend the functionality by supplying their own values. An example of this type of lookup is the Access Mode on [Application Service](#). The product provides many values for the access mode lookup, representing various actions a user may perform. Implementations may add their own values to this lookup. Documentation should indicate when functionality may be extended and should highlight the lookup value that can be extended.

CAUTION: Important! If you introduce new lookup values, you must prefix the lookup value code with **X** or **Y**. If you do not do this, there is a possibility that a future release of the application could introduce a new lookup value with the name you allocated.

- There may be some scenarios where the product supplies a base field and base lookup field with no base lookup values supplied. This occurs when the product doesn't have any base functionality driven by the lookup values. Typically this type of lookup is for information or categorization purposes. The configuration guide for the functional area associated with the lookup should include a configuration step regarding defining values for this type of lookup.
- The description of base delivered values may be overridden by an implementation.

An implementation may also identify the need for defining a new lookup field with its values.

Lookup - Main

Select **Admin > Database > Lookup** to maintain lookup values.

Description of Page

Field Name is the name of the field whose lookup values are maintained in the grid. If you need to add a new lookup field, you must first add the lookup field here, then navigate to the [Field](#) page to create a field with a data type of **Character** and an extended data type of **Flag**.

Owner indicates if this lookup field is owned by the base package or by your implementation (**Customer Modification**). This information is display-only.

Custom switch is used to indicate whether you are allowed to add valid values for a lookup field whose owner is not **Customer Modification**.

- If this switch is turned on, you may add new values to the grid for system owned lookup fields.
- If this switch is turned off, you may not add, remove or change any of the values for system owned lookup fields, with the exception of the override description.

This field is always protected for system owned lookup fields because you may not change a field from customizable to non-customizable (or vice versa).

Java Field Name indicates the name of the field as it is referenced in Java code.

The grid contains the lookup values for a specific field. The following fields are visible:

Field Value is the unique identifier of the lookup value. If you add a new value, it must begin with an **X** or **Y** (in order to allow future upgrades to differentiate between your implementation-specific values and base-package values).

Description is the name of the lookup value that appears on the various transactions in the system

Java Value Name indicates the unique identifier of the lookup value as it is referenced in Java code.

Status indicates if the value is **Active** or **Inactive**. The system does not allow **Inactive** values to be used (the reason we allow **Inactive** values is to support historical data that references a value that is no longer valid).

Detailed Description is the detailed description for a lookup value, which is provided in certain cases.

Override Description is provided if your implementation wishes to override the description of the value provided by the product.

NOTE: If you wish the override descriptions of your lookup values to appear in the application viewer, you must [regenerate](#) the data dictionary application viewer background process.

Owner indicates if this lookup value is owned by the base package or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add lookup values to a field. This information is display-only.

Defining Extendable Lookups

Extendable lookups are a way of defining valid values that are more sophisticated than simple lookups.

FASTPATH: Refer to [Defining Valid Values](#) for some background information.

The base product provides extendable lookups as part of the product. The following points highlight some functionality related to base-package extendable lookups.

- The base product may supply base extendable lookup values. Implementations are not permitted to remove base delivered extendable lookup values. It is also possible that implementations may be able to add custom values to base owned lookups. If an implementation is not permitted to add lookup values to the base extendable lookup, the extendable lookup's business object will include validation to prevent this. There is no equivalent of the Custom switch that is on the [lookup](#) field.
- There may be some scenarios where the product supplies a base extendable lookup with no base lookup values supplied. This occurs when the product doesn't have any base functionality driven by the extendable lookup values. The configuration guide for the functional area associated with the extendable lookup should include a configuration step regarding defining values for this type of extendable lookup.
- The description of base delivered values may be overridden by an implementation.

Open this page using **Admin > Database > Extendable Lookup**.

You are brought to the **Extendable Lookup Query** where you need to search for the extendable lookup object (i.e., its business object).

Once you have found the appropriate extendable lookup, select the value and you are brought to a standard [All-in-One](#) portal that lists the existing lookup values for the extendable lookup. The standard actions for an All-in-One portal are available [here](#).

Defining Additional Attributes

The product provides a few different ways to define additional values for an extendable lookup. Some of the methods are only relevant for base delivered lookup values as they may impact whether or not an implementation can update the values.

The following table highlights the options available and some summary information about what the option provides.

Option	Brief Description	Extendable Lookup Value Searchable by this Attribute?	Base Delivered Value Modifiable?
Element mapped to BO_DATA_AREA	The element is mapped to a CLOB field that allows for base delivered values to be modified.	No	Yes
Element mapped to BASE_BO_DATA_AREA	The element is mapped to a CLOB field that does not allow for base delivered values to be modified.	No	No
Flattened characteristic	The element is defined using the flattened characteristic mechanism.	Yes	No

The following points highlight information from the table above:

- The decision of defining an additional attribute using a CLOB mapping or a flattened characteristic will depend on whether the functionality expects that the lookup will be known when the attribute is needed (in which case a CLOB mapping is appropriate) or if the functionality expects to determine the extendable lookup based on the attribute (in which case, a flattened characteristic is appropriate).
- When the base product defines an extendable lookup with additional attributes and intends to provide base extendable lookup values, it needs to determine whether or not implementations may update the additional attribute or not.
 - If no and the value is mapped to a CLOB, it will map the value to the BASE_BO_DATA_AREA column. This means that implementations will receive an owner mismatch error when attempting to change the value. In addition, upgrading to a new release will replace the value with the base value.
 - If yes and the value is mapped to a CLOB, it will map the value to the BO_DATA_AREA column. This means that implementations will be able to change the value for a base owned record. In addition, upgrading to a new release will not make any changes to the value.
 - For values mapped to a characteristic, the product does not support an implementation changing the value of a base delivered record. If the product would like to support an implementation overriding this type of value, the business object will need to be designed with a corresponding “override” element (also a flattened characteristic), similar to how the product supplies an Override Description field to support an implementation overriding the base product delivered description for a base value. This element will not be delivered with any value and will allow an implementation to populate that value.

NOTE: Note that in this situation, the product functionality that uses this value must cater for the override value.

- All of this detail is only relevant for base provided extendable lookup values. If an implementation adds custom values for a base supplied extendable lookup, all the additional attributes may be populated as appropriate.
- If an implementation defines a custom extendable lookup business object and wants to define an additional attribute using a CLOB, it doesn’t matter which CLOB column is used. Both BO_DATA_AREA and BASE_BO_DATA_AREA provide the same functionality for custom business objects.

The Big Picture Of Audit Trails

The topics in this section describe auditing, enabling auditing for fields, and auditing queries that you can use to view audit records.

Captured Information

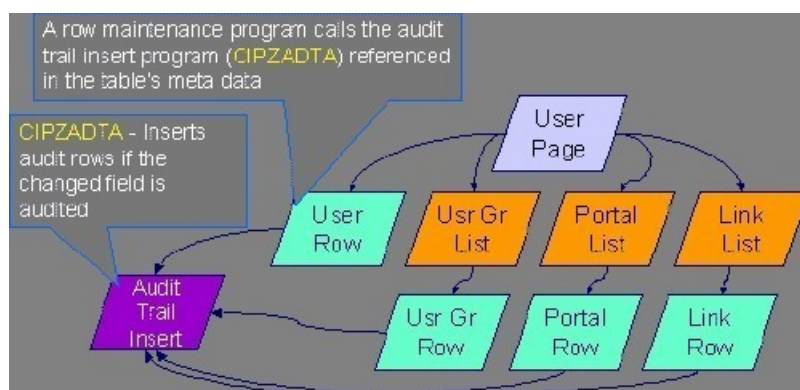
When auditing is enabled for a field, the following information is recorded when the field is changed, added and/or deleted (depending on the actions that you are auditing for that field):

- User ID
- Date and time
- Table name
- Row's prime key value
- Field name
- Before image (blank when a row is added)
- After image (blank when a row is deleted)
- Row action (add, change, delete)

How Auditing Works

You enable auditing on a table in the table's meta-data by specifying the name of the table in which to insert the audit information (the audit table) and the name of the program responsible for inserting the data (the audit trail insert program). Then you define the fields you want to audit by turning on each field's audit switch in the table's field meta-data. You can audit fields for delete, insert and update actions.

Once auditing is enabled for fields in a table, the respective row maintenance program for the table assembles the list of changed fields and calls the audit trail insert program (**CIPZADTA** is supplied with the base package). If any of the changed fields are marked for audit, **CIPZADTA** inserts audit rows into the audit table (**CI_AUDIT** is the audit table supplied with the base package).



NOTE: Customizing Audit Information. You may want to maintain audit information other than what is described in [Captured Information](#) or you may want to maintain it in a different format. For example, you may want to maintain audit information for an entire row instead of a field. If so, your implementation team can use **CIPZADTA** and **CI_AUDIT** as examples when creating your own audit trail insert program and audit table structures.

The Audit Trail File

Audit log records are inserted in the audit tables you define. The base product contains a single such table (called [CI_AUDIT](#)). However, the audit insert program ([CIPZADTA](#)) is designed to allow you to use multiple audit tables.

If you want to segregate audit information into multiple tables, you must create these tables. Use the following guidelines when creating new audit tables (that use the [CIPZADTA](#) audit insert program):

- The new audit tables must look identical to the base table ([CI_AUDIT](#)).
- The new tables must be prefixed with **CM** (e.g., [CM_AUDIT_A](#), [CM_AUDIT_B](#), etc.).
- The name of the new table must be referenced on the various tables whose changes should be logged in the new table.

NOTE: Interesting fact. It's important to note if you use your own tables (as opposed to using the base package table called [CI_AUDIT](#)), the SQL used to insert and access audit trail records (in [CIPZADTA](#)) is dynamic. Otherwise, if the base package's table is used, the SQL is static.

How To Enable Auditing

Enabling audits is a two-step process:

- First, you must turn on auditing for a table by specifying an audit table and an audit trail insert program.
- Second, you must specify the fields and actions to be audited for the table.

The following topics describe this process.

Turn On Auditing For a Table

In order to tell the system which fields to audit, you must know the name of the table on which the field is located. You must specify the audit table and the audit trail insert program for a table in the table's meta-data.

NOTE: Most of the system's table names are fairly intuitive. For example, the user table is called [SC_USER](#), the navigation option table is called [CI_NAV_OPT](#), etc. If you cannot find the table using the search facility on the [Table Maintenance](#) page, try using the [Data Dictionary](#). If you still cannot find the name of the table, please contact customer support.

To enable auditing for a table:

- Navigate to the [Table maintenance](#) page and find the table associated with the field(s) for which you want to capture audit information.
- Specify the name of the **Audit Table**.

NOTE: Specifying the Audit Table. You can use the audit table that comes supplied with the base package ([CI_AUDIT](#)) to audit multiple tables and fields. All the audit logs are combined in a single table ([CI_AUDIT](#)). However, you can also have a separate audit table for each audited table. Refer to [The Audit Trail File](#) for more information.

- Specify the name of the **Audit Program** ([CIPZADTA](#) is the default audit program supplied with the base package).

CAUTION: By default, none of a table's fields are marked for audit. Even though you have enabled auditing for a table, you must still specify the fields and actions on those fields to be audited (see below).

Specify The Fields and Actions To Be Audited

The system only audits actions (insert, update and delete) made to fields that you want audited.

To specify the fields and actions to be audited:

- Navigate to the [Table - Table Field maintenance](#) page for a table on which you have enabled auditing.
- For each field you want to audit, specify the actions you want to audit by turning on the **Audit Delete**, **Audit Insert** and **Audit Update** switches as appropriate.

NOTE: Note. You can also turn on the audit switches using the Field grid at the bottom of the [Table maintenance page](#).

CAUTION: Audit Program Caching! The audit program from the table meta-data is read into a program cache on the application server whenever the date changes or when the server starts. If you implement new auditing on a table, your audit trail does not become effective until this program cache is reloaded. In other words, new audits on tables where the audit program was not previously specified do not become effective until the next day (or the next restart of the application server). However, if you change the fields to be audited for a table where the audit program is already in the cache, your changes are effective immediately.

Audit Queries

There are two queries that can be used to access the audit information.

Audit Query by User

This transaction is used to view changes made by a user that are stored on a given [Audit Trail File](#).

CAUTION: The system only audits changes that you've told it to audit. Refer to [The Big Picture Of Audit Trails](#) for more information.

Navigate to this page by selecting **Admin > Audit Query > Audit Query By User**.

Description of Page

To use this transaction:

- Enter the **User ID** of the user whose changes you wish to view.
- Enter the name of the table on which the audit trail information is stored in **Audit Table**. Refer to [The Audit Trail File](#) for more information about this field.

NOTE: Default Note. If only one audit table is used to store audit trail information, that table is defaulted.

- Specify a date and time range in **Created between** to restrict the records that result from the query.

NOTE: Default Note. The current date is defaulted.

- Click the search button to display all changes recorded on a specific audit table associated with a given user.

Information on this query is initially displayed in reverse chronological order.

The following information is displayed in the grid:

- **Row Creation Date** is the date and time that the change was made.
- **Audited Table Name** contains the name of the table whose contents were changed.

- **Primary Key** is the prime key of the row in the **Audited Table** whose contents were changed.
- **Audited Field Name** is the name of the field that was changed.
- **Audit Action** indicates whether the row action was **Add**, **Change** or **Delete**.
- **Field Value Before Update** contains the content of the field before the change. This column is blank if information was **Added**.
- **Field Value After Update** contains the content of the field after the change. This column is blank if information was **Deleted**.

Audit Query by Table / Field / Key

This transaction is used to view audited changes made to a given table.

CAUTION: The system only audits changes that you've told it to audit. Refer to [The Big Picture Of Audit Trails](#) for more information.

NOTE: Most of the system's table names are fairly intuitive. For example, the user table is called **SC_USER**, the navigation option table is called **CI_NAV_OPT**, etc. If you cannot find the table using the search facility on the [Table Maintenance](#) page, try using the [Data Dictionary](#). If you still cannot find the name of the table, please contact customer support.

This transaction can be used in several different ways:

- You can view all audited changes to a table. To do this, enter the **Audited Table Name** and leave the other input fields blank.
- You can view all audited changes to a given row in a table (e.g., all changes made to a given user). To do this, enter the **Audited Table Name** and row's prime key (the row's prime key is entered in the field(s) beneath **Audited Field Name**).
- You can view all audited changes to a given field in a table (e.g., all changes made to all customers' rates). To do this, enter the **Audited Table Name** and the **Audited Field Name**.
- You can view all audited changes to a given field on a specific row. To do this, enter the **Audited Table Name**, the **Audited Field Name**, and row's prime key (the row's prime key is entered in the field(s) beneath **Audited Field Name**).

Navigate to this page by selecting **Admin > Audit Query > Audit Query By Table/Field/Key**.

Description of Page

To use this transaction:

- Enter the name of the table whose changes you wish to view in **Audited Table Name**.
- If you wish to restrict the audit trail to changes made to a specific field, enter the **Audited Field Name**.
- If you wish to restrict the audit trail to changes made to a given row, enter the row's prime key (the row's prime key is entered in the field(s) beneath **Audited Field Name**). These fields are dynamic based on the **Audited Table Name**.
- Specify a date and time range in **Created between** to restrict the records that result from the query.

NOTE: The current date is defaulted.

- Click the search button to display all changes made to this data.

Information on this query is initially displayed in reverse chronological order by field.

The following information is displayed in the grid:

- **Create Date/Time** is the date / time that the change was made.
- **User Name** is the name of the person who changed the information.

- **Primary Key** is the prime key of the row in the **Audited Table** whose contents were changed.
- **Audited Field Name** is the name of the field that was changed.
- **Audit Action** indicates whether the row action was **Add**, **Change** or **Delete**.
- **Value Before Update** contains the content of the field before the change. This column is blank if information was **Added**.
- **Value After Update** contains the content of the field after the change. This column is blank if information was **Deleted**.

Bundling

The topics in this section describe the bundling features in the application.

About Bundling

Bundling is the process of grouping entities for export or import from one environment to another.

For example, you might export a set of business objects and service scripts from a development environment and import them into a QA environment for testing. The group of entities is referred to as a bundle. You create export bundles in the source environment; you create import bundles in the target environment.

Working with bundles involves the following tasks:

- Configuring entities for bundling if they are not preconfigured
- Creating an export bundle, which contains a list of entities to be exported from the source environment
- Creating an import bundle to import those entities to the target environment
- Applying the import bundle, which adds or updates the bundled entities to the target environment

Sequencing of Objects in a Bundle

Bundle entities are added or updated to the target environment in the sequence defined in the bundle

Typically, the sequence of entities does not matter. However, sequence is important in the following situations:

- Entities that are referenced as foreign keys should be at the top of the sequence, before the entities that reference them. Specify zones last, as they typically contain numerous foreign key references.
- When importing a business object, specify the business object first, then its plug-in scripts, then the algorithms that reference the scripts, and then the algorithm types that reference the algorithms.
- When importing a portal and its zones, specify the portal first and then its zones.
- When importing a multi-query zone, specify the referenced zones first and then the multi-query zone.
- Always specify algorithm types before algorithms.

You can specify the sequence when you define the export bundle or when you import the bundle to the target environment.

Recursive Key References

Recursive foreign keys result when one object has a foreign key reference to another object that in turn has a foreign key reference to the first object.

For example, a zone has foreign keys to its portals, which have foreign keys to their zones. If the objects you want to bundle have recursive relationships, you must create a 'bundling add' business object that has only the minimal number of elements

needed to add the entity. A bundling add business object for a zone contains only the zone code and description, with no references to its portals. In the same way, a bundling add business object for a portal defines only its code and description.

When you apply the bundle, the system initially adds the maintenance object based on the elements defined in the bundling add business object. Before committing the bundle, the system updates the maintenance object with the complete set of elements based on its physical business object.

Owner Flags on Bundled Entities

The owner flag of the entities in an import bundle must match the owner flag of the target environment.

If you need to import objects that your source environment does not own, you must replace the owner flag in the import bundle with the owner flag of the target environment.

Configuring Maintenance Objects for Bundling

All base package meta-data objects are pre-configured to support bundling. All other objects must be manually configured.

If a base package maintenance object is pre-configured for bundling, the **Eligible For Bundling** option will be set to "Y" on the Options tab for the maintenance object.

To configure other objects for bundling, review the configuration tasks below and complete all those that apply:

Complete this configuration task...	for...
Make maintenance objects eligible for bundling	All objects to be included in the bundle
Add a foreign key reference	All objects to be included in the bundle
Create a physical business object	All objects to be included in the bundle
Create a bundling add business object	Objects with recursive foreign key references
Add the Current Bundle zone	All objects, if you want the Current Bundle zone to appear on the maintenance object's dashboard. This is not required by the bundling process.
Create a custom Entity Search zone and add it to the Bundle Export portal	All objects, if you want them to be searchable in the Bundle Export portal. This is not required by the bundling process.

Making Maintenance Objects Eligible for Bundling

The "Eligible For Bundling" maintenance object option must be set to "Y" for all bundled objects.

To make maintenance objects eligible for bundling:

1. Select **Admin > Maintenance Object > Search** and search for the maintenance object.
2. On the Options tab, add a new option with the type **Eligible For Bundling**.
3. Set the value to "Y" and click **Save**.

Adding a Foreign Key Reference

Each maintenance object in a bundle must have a foreign key reference. Bundling zones use the foreign key reference to display the standard information string for the maintenance object.

To add a foreign key reference to the maintenance object:

1. Select **Admin > FK Reference > Add** and set up a foreign key reference for the primary table of the maintenance object.
2. Select **Admin > Maintenance Object > Search** and search for the maintenance object.

3. On the **Option** tab, add a new option with the type **Foreign Key Reference**. The value is the name of the foreign key reference you just created.

Creating a Physical Business Object

Each maintenance object in a bundle must have a physical business object. The physical business object's schema represents the complete physical structure of the maintenance object, and includes elements for all fields in the maintenance object's tables. The bundling process uses this schema to generate the XML for the import bundle.

To create a physical business object for the maintenance object:

1. Select **Admin > System > Business Object > Add** and specify the maintenance object.
2. Click **Generate** in the **BO Schema** dashboard zone to generate a schema that looks like the physical structure of the maintenance object.
3. Save the physical business object.
4. Select **Admin > Maintenance Object > Search** and search for the maintenance object.
5. On the **Option** tab, add a new option with the type **Physical Business Object**. The value is the name of the physical business you just created.

Creating a Bundling Add Business Object

If the objects to be bundled have recursive foreign key references, you must create a bundling add business object to avoid problems with referential integrity.

To create a bundling add business object:

1. Select **Admin > System > Business Object > Add** and specify the maintenance object.
2. Click **Generate** in the **BO Schema** dashboard zone to generate a schema that looks like the physical structure of the maintenance object.
3. Remove all elements that are not essential. Typically, only a code and description are required.
4. Save the physical business object.
5. Select **Admin > Maintenance Object > Search** and search for the maintenance object you want to bundle.
6. On the **Option** tab, add a new option with the type **Bundling Add BO**. The value is the name of the bundling add business object you just created.

Adding the Current Bundle Zone

If you want the Current Bundle zone to appear on the maintenance object's dashboard, you must add the Current Bundle zone as a context-sensitive zone for the maintenance object.

To add the Current Bundle zone to the maintenance object:

1. Select **Admin > System > Context Sensitive Zone** and search for the navigation key for the maintenance object.
2. Add the Current Bundle zone, F1-BNDLCTXT, to that navigation key.

Adding a Customized Entity Search Query Zone to the Bundle Export Portal

If you want the maintenance object to be searchable in the Bundle Export portal, you must first create an entity-specific query zone to search for the maintenance object. Then you must create a customized entity search zone that references this new query zone. Finally, you must add the customized entity search zone to the Bundle Export portal.

To make the maintenance object searchable:

1. Create an entity-specific query zone to search for the maintenance object:
 - a) Select **Admin > System > Zone > Search** and search for one of the base query zones, such as the Algorithm Search zone, F1-BNALGS.
 - b) Click the **Duplicate** button in the page actions toolbar.
 - c) Enter a name for the new zone.
 - d) Click **Save**.
 - e) Locate the **User Filter** parameter in the parameter list. Add SQL to search for the maintenance object(s) you want to appear in the zone.
 - f) Save the query zone.
2. Create a customized entity search zone:

This step only needs to be done once. If you already have a customized search zone in the Bundle Export portal go to step 3

 - a) Select **Admin > System > Zone > Search** and search for the F1-BNDLENTQ Entity Search zone.
 - b) Duplicate this zone (as described above).
 - c) Remove any references to base query zones.
3. Add the new entity-specific query zone to the customized entity search zone:
 - a) Locate the customized entity search zone for your Bundle Export portal. This is the zone created in Step 2.
 - b) Locate the Query Zone parameter in the parameter list. Add the name of the query zone you created in Step 1.
 - c) Save the entity search zone.
4. Add the customized entity search zone to the Bundle Export portal:

This step needs to be done only once.

 - a) Select **Admin > System > Portal > Search** and search for the Bundle Export portal, F1BNDLEM.
 - b) In the zone list, add the entity search zone you created in Step 2. (Add the new zone after the base entity search zone).
 - c) Save the portal.

Working with Bundles

Use the Bundle Export portal to create an export bundle. The export bundle contains a list of entities to be exported from the source environment. When you are ready to import the objects, use the Bundle Import portal to import the objects to the target environment.

Creating Export Bundles

An export bundle contains a list of entities that can be imported into another environment.

To create an export bundle:

1. Log on to the source environment from which objects will be exported.
2. Select **Admin > System > Bundle Export > Add**.
3. Complete the fields in the Main section to define the bundle's basic properties.

NOTE: You can use the Entities section to add bundle entities now, or save the bundle and then add entities as described in step 5.

4. Click **Save** to exit the Edit dialog. The export bundle status is set to Pending.
5. While an export bundle is in Pending state, use any of the following methods to add entities to the bundle:
 - a) Use the **Entity Search** zone on the Bundle Export portal to search for entities and add them to the bundle. If an entity is already in the bundle, you can remove it.
 - b) To import entities from a .CSV file, click **Edit** on the Bundle Export portal, and then click **CSV File to Upload**. Specify the file name and location of the .CSV file containing the list of entities. Click **Submit** to upload the file, and then click **Save** to save the changes.
 - c) Use the **Current Bundle** zone in the dashboard of the entity you want to add. (All entities that are configured to support bundling display a Current Bundle zone). This zone displays a list of all pending export bundles to which you can add the entity.
 - d) When you check an entity into revision control, specify the export bundle on the **Revision Info** dialog.
6. When you have added all entities, click **Bundle** in the Bundle Actions zone on the Bundle Export portal. The export bundle state is set to Bundled and the Bundle Details zone displays the XML representation of every entity in the bundle.

NOTE: The owner flags of the entities in the bundle must match the owner flag of the bundle itself. If the owner flags do not match, the system displays a warning message. Click **OK** to continue or **Cancel** to abort the bundle. If you click OK, you will need to resolve the owner flag discrepancy before you import the bundle to the target environment.

7. Copy the XML from the **Bundle Detail** zone to the clipboard (or to a text file). You can now create an import bundle and apply it to the target environment.

NOTE: If you need to make additional changes to the bundle, you must change the bundle state by selecting the **Back to Pending** button in the **Bundle Actions** zone.

Creating and Applying Import Bundles

Import bundles define a group of entities to be added or updated in the target environment.

Before you create an import bundle, you must have already created an export bundle, added entities, and set the bundle's state to Bundled.

To create an import bundle and apply it to the target environment:

1. If you have not already copied the XML from the export bundle, do so now:
 - a) Select **Admin > System > Bundle Export** and search for the bundle.
 - b) Copy the XML from the **Bundle Detail** zone to the clipboard (or to a text file).
2. Log on to the target environment.
3. Select **Admin > System > Bundle Import > Add**.
4. In the **Bundle Actions** zone, click **Edit XML**.
5. Paste the contents of the clipboard (or text file if you created one) into the **Bundle Detail** zone.
6. Make any necessary changes to the XML and click **Save**. The status of the import bundle is set to Pending.

NOTE: Use caution when editing the XML to avoid validation errors.

7. To remove entities from the import bundle or change their sequence, click **Edit**. Enter your changes and click **Save** to exit the Edit dialog.
8. When you are ready to apply the bundle, click **Apply**. The import bundle state is set to Applied and the entities are added or updated in the target environment.

Editing Export Bundles

You can add or remove entities from an export bundle when it is in Pending state. You can also change the sequence of entities.

To edit to an export bundle that has already been bundled, you must change the bundle state by selecting the **Back to Pending** button on the Bundle Export portal.

To edit a pending export bundle:

1. Open the bundle in edit mode.
2. Click **Edit** on the Export Bundle portal.
3. Make any necessary changes on the edit dialog and then click **Save**.

Editing Import Bundles

You can remove entities from an import when it is in Pending state. You can also change the sequence of entities and edit the generated XML.

To edit a pending import bundle:

1. Open the bundle in edit mode.
2. To edit the XML snapshot, click **Edit XML**. Edit the XML code as needed, then click **Save**.

NOTE: Use caution when editing the XML to avoid validation errors.

3. To remove entities or change their sequence, click **Edit**. Make any necessary changes and click **Save**.

Revision Control

The topics in this section describe the revision control features in the application.

About Revision Control

Revision control is a tool provided for the development phase of a project to allow a user to check out an object that is being worked on. In addition, it captures the version of the maintenance object when users check in an update, maintaining a history of the changes to the object.

If revision control is enabled for an object you must check out the object to change it. While the object is checked out no one else can work on it. You can revert all changes made since checking out an object, reinstate an older version of an object, recover a deleted object, and force a check in of an object if someone else has it checked out.

NOTE: Revision control does not keep your work separate from the environment. Because the metadata for maintenance objects is in the central database, any changes you make to an object while it is checked out will be visible to others and may impact their work.

Many of the maintenance objects used as configuration tools are already configured for revision control, but it is turned off by default. For example, business objects, algorithms, data areas, UI maps, and scripts are pre-configured for revision control.

Turning On Revision Control

Revision control is turned off by default for maintenance objects that are configured for revision control.

To turn on revision control:

1. Add the base package **Checked Out** zone to the Dashboard portal.

- a) Select **Admin > System > Portal**.
- b) Search for the portal **CI_DASHBOARD**.
- c) In the zone list for the **Dashboard** portal, add the zone **F1-USRCHKOUT**.

2. Set up application security.

For users to have access to revision control, they must belong to a user group that has access to the application service **F1-OBJREVBOAS**.

3. Add the revision control algorithm to the maintenance object that you want to have revision control.

This step must be repeated for each maintenance object that you want to have revision control.

- a) Select **Admin > System > Maintenance Object** and search for the maintenance object that you want to have revision control.
- b) On the **Algorithms** tab of the maintenance object, add the revision control algorithm **F1-REVCTL**.

Configuring Maintenance Objects for Revision Control

Most configuration tool maintenance objects are pre-configured for revision control. You can configure other maintenance objects for revision control, as well.

To configure other objects for revision control:

1. Create a physical business object for the maintenance object.

A physical business object is one that has a schema with elements for all of the fields for the tables in the maintenance object. Follow these steps to create a physical business object:

- a) Select **Admin > System > Business Object > Add** and specify the maintenance object.
- b) Use the **BO Schema** dashboard zone to generate a schema that looks like the physical structure of the maintenance object.
- c) Save the physical business object.
- d) Select **Admin > Database > Maintenance Object > Search** and search for the maintenance object for which you want to enable revision control.
- e) On the **Options** tab of the maintenance object add a new option with the type **Physical Business Object**. The value is the name of the physical business object that you just created.

2. Add a foreign key reference to the maintenance object.

The revision control zones will display the standard information string for the object based on the foreign key reference. Follow these steps to create a foreign key reference:

- a) Select **Admin > Database > FK Reference > Add** and set up a foreign key reference for the primary table of the maintenance object.
 - b) Select **Admin > Database > Maintenance Object > Search** and search for the maintenance object.
 - c) On the **Options** tab of the maintenance object, add a new option with the type **Foreign Key Reference**. The value is the name of the foreign key reference that you just created.
3. Add the **Revision Control** zone to the maintenance object.
- a) Select **Admin > System > Context Sensitive Zone** and search for the navigation key for the maintenance object.
 - b) Add the Revision Control zone, F1-OBJREVCTL, to that navigation key.
4. Add the revision control algorithm to the maintenance object.
- a) Select **Admin > Database > Maintenance Object** and search for the maintenance object that you want to have revision control.
 - b) On the **Algorithms** tab of the maintenance object, add the revision control algorithm F1-REVCTL.

Working with the Revision Control Zones

You use two zones in the dashboard to work with revision controlled objects when revision control is turned on.

The **Revision Control** zone gives you several options for managing the revision of the currently displayed object. This zone also shows when the object was last revised and by whom. This information is linked to the **Revision Control Search** portal which lists all of the versions of the object.

Using the Revision Control zone you can:

- Check out an object in order to change it.
- Check in an object so others will be able to work on it.
- Revert the object back to where it was at the last checkout.
- Force a check in of an object that is checked out by someone else. You need special access rights to force a check in.
- Delete an object.

The **Checked Out** zone lists all of the objects that you currently have checked out. Clicking on an object listed in this zone will take you to the page for that object. The zone is collapsed if you have no objects checked out.

See [Revision Control Search](#) for more information about Check In, Force Check In, and Check Out one or more records simultaneously.

Checking Out an Object

You must check out a revision controlled object in order to change it.

An object must have revision control turned on before you can check it out.

NOTE: When you first create or update an object a dialog box informs you that the object is under revision control. You can select **OK** to check out the object and save your changes, or **Cancel** to stop the update.

1. Go to the object that you want to work on.
2. Select **Check Out** in the **Revision Control** dashboard zone.

Checking In an Object

You must check in a revision controlled object in order to create a new version of it. Checking in an object also allows others to check it out.

1. Select a link in the **Checked Out** dashboard zone to go to the object that you want to check in.
2. Select **Check In** in the **Revision Control** dashboard zone.
3. Provide details about the version:
 - In the **External References** field state the bug number, enhancement number, or a reason for the revision.
 - In the **Detailed Description** field provide additional details regarding the revision.
 - In the **Keep Checked Out** box specify if you want to keep the object checked out. If you keep the object checked out then your revision is a new version that you can restore later.
 - In the **Add To Bundle** box specify if the object belongs to a bundle.
4. Select **OK** to check in the object.

Reverting Changes

Reverting changes will undo any changes you made since you checked out an object.

To revert changes:

1. Go to the object that you want to revert.
2. Select **Revert** in the **Revision Control** dashboard zone.
3. In the confirmation dialog box select **OK** to confirm the action or **Cancel** to return to the object page.

Once reverted, the object can be checked out by another user.

Forcing a Check In or Restore

You can force a check in if an object is checked out by another user and that person is not available to check it in.

You must have proper access rights to force a check in or restore.

To force a check in or restore:

1. Go to the object that is checked out by another user.
2. Select **Force Check In** or **Force Restore** in the Revision Control zone.

The **Force Check In** option is the same as a regular check in. The **Force Restore** option checks in the object and restores it to the previously checked in version.

Deleting an Object

If revision control is turned on for an object, you must use the **Revision Control** zone to delete it.

The object must be checked in before it can be deleted.

To delete a revision controlled object:

1. Go to the object that you want to delete.
2. Select **Delete** in the **Revision Control** zone.

3. Provide details regarding the deletion.
4. Select **OK** to delete the object.

The system creates a revision record before the object is deleted so that the deleted object can be restored.

Restoring an Object

You can restore an older version of either a current object or a deleted object.

An object must be checked in before an older version can be restored.

To restore an object:

1. Go to the **Revision History** portal for the object.
If the object was deleted you must search for it by going to **Admin > Revision Control**.
2. Select the desired entity by clicking the hyperlink in the **Details** column.
3. Locate the row in the version history that has the version that you want to restore and click **Restore**.
4. In the confirmation dialog box select **OK** to confirm the action or **Cancel** to return to the object page.

Working with the Revision Control Portal

The **Revision Control** portal lists information about each version of a revision controlled object.

You can navigate to the **Revision Control** portal from either a link in the **Revision Control** dashboard zone or by going to **Revision Control** portal through **Admin**.

If you want to find the Revision History entry for an earlier version or deleted object, you must search for the object using the **Revision Control Search** portal. Once you select the desired entry, you can restore a previous version of the object clicking **Restore** in the row for the version that you want to restore. You can also see the details of each version by clicking the broadcast icon for that version.

See [Working with Revision Control Zones](#) for more information about tasks that can be performed through Revision Control.

Revision Control Search

The **Revision Control Search** portal allows users to search for entities that have a revision history. The **Search By** dropdown provides additional functionality so that users can search for revisions that are associated to theirs or other's user ID. Users can also **Check In**, **Force Check In**, or **Check Out** one or more entities through this portal.

Zone Options

- **Revision History Search** (*default*) allows the user to query for revised entities based on a combination of criteria.
 - In the **User ID** field, enter the user ID that is associated with a revision.
 - In the **External Reference** field, enter an ID from an external system and is associated with a revision.
 - In the **Maintenance Object** dropdown menu, select the Maintenance Object that is associated with a revision. The options in this list are populated by the Maintenance Objects that are active to track revision.
 - In the **Key 1**, **Key 2**, **Key 3**, **Key 4**, **Key 5** fields, enter the primary identifier(s) for the revised entity. Typically, the entity only requires a single key, but some entities require more than one (for example, Oracle Utilities Customer Care and Billing SA Type require CIS Division and SA Type).
 - In the **Status** dropdown menu, select the entity status for your search.

- **Check In** allows the user to search for entities currently checked out to the logged in user ID and a combination of criteria. Once the search results are returned, the user has the option to select one or more entities and check them in.
 - In the **Maintenance Object** dropdown menu, select the Maintenance Object that is associated with a revision. The options in this list are populated by the Maintenance Objects that are active to track revision.
 - In the **Key** field, enter the primary identifier(s) for the revised entity.
- **Force Check In** allows the user to search for entities that are currently checked out by other user IDs (excluding the logged in user ID) based on a combination of criteria. Once the search results are returned, the user has the option to select one or more entities and check them in.
 - In the **Checked Out By User** field, enter the user ID that has the entity in a Checked Out status.
 - In the **Maintenance Object** dropdown menu, select the Maintenance Object that is associated with a revision. The options in this list are populated by the Maintenance Objects that are active to track revision.
 - In the **Key** field, enter the primary identifier(s) for the revised entity.
- **Check Out** allows the user to search for entities currently checked in user ID and a combination of criteria. Once the search results are returned, the user has the option to select one or more entities and check them out.
 - In the **Maintenance Object** dropdown menu, select the Maintenance Object that is associated with a revision. The options in this list are populated by the Maintenance Objects that are active to track revision.
 - In the **Key** field, enter the primary identifier(s) for the revised entity.

Please see [Working with Revision Control Zones](#) for more information about working with individual entities.

Information Lifecycle Management

Information Lifecycle Management (ILM) is designed to address data management issues, with a combination of processes and policies so that the appropriate solution can be applied to each phase of the data's lifecycle.

Data lifecycle typically refers to the fact that the most recent data is active in the system. As time progresses, the same data becomes old and unused. Older data becomes overhead to the application not only in terms of storage, but also in terms of performance. This older data's impact can be reduced by using advanced compression techniques, and can be put into slower and cheaper storage media. Depending on how often it's accessed, it can be removed from the system to make an overall savings of cost and performance. The target tables for ILM are transactional tables that have the potential to grow and become voluminous over time.

The Approach to Implementing Information Lifecycle Management

This section describes the product approach to implementing ILM for its maintenance objects (MOs).

NOTE: The term archiving is used to cover any of the possible steps an implementation may take in their data management strategy, including compression, moving to cheaper storage, and removing the data altogether.

Age is the starting point of the ILM product implementation for some of its high volume data. In general "old" records are considered eligible to be archived. In the product solution, maintenance objects (MOs) that are enabled for ILM have an ILM Date on the primary table and the date is typically set to the record's creation date. (An MO may have special business rules for setting this date, in which case, a different date may be used to set the initial ILM Date). For implementations that want to use ILM to manage the records in the MO, the ILM date is used for defining partitions for the primary table.

There are cases where a record's age is not the only factor in determining whether or not it is eligible to be archived. There may be some MOs where an old record is still 'in progress' or 'active' and should not be archived. There may be other MOs where certain records should never be archived. To evaluate archive eligibility using information other than the ILM Date, the ILM enabled MOs include an ILM Archive switch that is used to explicitly mark records that have been evaluated and

should be archived. This allows DBAs to monitor partitions based on age and the value of this switch to evaluate data that may be ready to be archived.

Evaluating records to determine their archive eligibility should still occur on “old” records. The expectation is that a large percentage of the old records will be eligible for archiving. The small number that may be ineligible could be updated with a more recent ILM date. This may cause the records to move into a different partition and can delay any further evaluation of those records until more time has passed.

For each MO enabled for ILM, the product provides a batch process to review “old” records and an ILM eligibility algorithm that contains business logic to evaluate the record and mark it eligible for archiving or not. The following sections provide more information about the batch process and algorithm functionality.

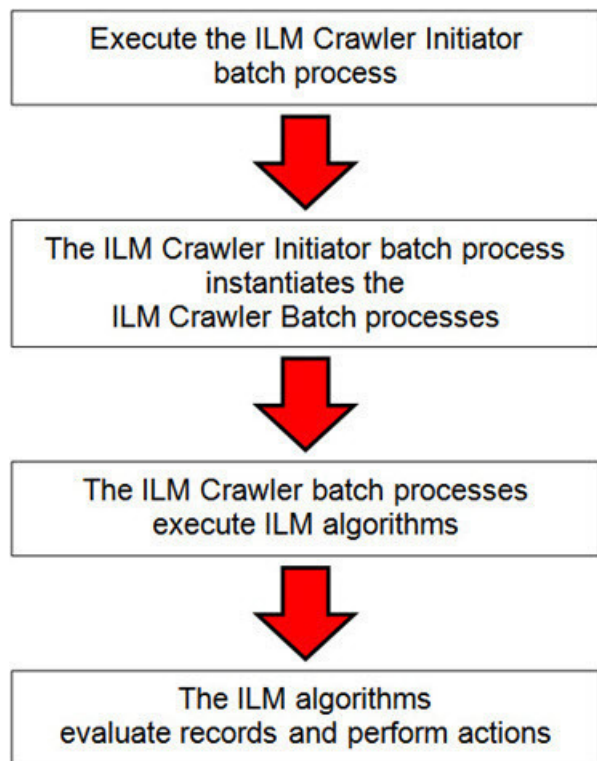
Batch Processes

There are two main types of batch processes that manage data for ILM in the application: ILM Crawler Initiator and individual ILM Crawlers (one for each MO that is configured for ILM).

- **ILM Crawler Initiator: (F1-ILMIN)** - The ILM Crawler Initiator is a *driver* batch process that starts the individual ILM Crawler batch control as defined by the MO’s options.

Restartable: In case of server failure, the ILM Crawler Initiator process can be restarted, which will also restart the ILM Crawler processes.

- **ILM Crawler:** Each maintenance object that is configured for ILM defines an ILM Crawler. These are *child* batch processes that can be started either by the ILM Crawler Initiator or by a standalone batch submission.



The ILM Crawler batch process selects records whose retention period has elapsed and invokes the MO’s ILM eligibility algorithm to determine if the record is ready to be archived or not. The ILM eligibility algorithm is responsible for setting the record’s ILM archive switch to ‘Y’ and updating the ILM date, if necessary.

The retention period defines the period that records are considered active. It spans the system date and cutoff date (calculated as system date - retention days).

The retention days of an MO is derived as follows:

- If the ILM Retention Days MO option is defined, that is used.
- Otherwise, the Default Retention Days from the ILM Master Configuration record is used.

An error is issued if no retention period is found.

The crawler calculates the cutoff date and selects all records whose ILM archive switch is 'N' and whose ILM date is prior to the cutoff date. Each record returned is subject to ILM eligibility.

If the Override Cutoff Date parameter is supplied, it will be used instead of the calculated cutoff date. An error is issued if the override cutoff date is later than the calculated cutoff date. This parameter is useful if an object has many years of historic data eligible for archiving. Setting this parameter allows for widening the retention period and therefore limiting the process to a shorter period for initial processing

NOTE: ILM Crawler batch processes are designed not to interfere with current online or batch processing. Because of this, these batch processes can run throughout the day.

NOTE: Before passing the cut-off date to the algorithm, the ILM crawler ensures that the number of days calculated (System Date – override cut-off date) is more than the retention period specified in the MO option or the Master Configuration. If the number of days calculated is **less than** the retention period specified on the MO option or the Master Configuration, then it throws an error.

Eligibility Algorithm

Algorithms are triggered by the ILM batch crawler for the maintenance object. The key responsibility of the ILM algorithm is to determine whether a record can be marked as ready to be archived or not. If a record is determined to be ready for archive, the algorithm should set the ILM Archive switch to Y. If not, the algorithm leaves the switch set to N and may decide to update the ILM Date to something more recent (like the System Date) to ensure that the record does not get evaluated again until the future.

This algorithm is plugged into the [Maintenance Object — Algorithm](#) collection.

Oracle Utilities Application Framework provides the algorithm **ILM Eligibility Based on Status (F1-ILMELIG)** to support the ILM batch crawler. Refer to the algorithm type description for details about how this algorithm works. If a maintenance object has special business rules that are evaluated to determine the eligibility for ILM, a custom algorithm can be created and applied by the implementation team.

Enabling ILM for Supported Maintenance Objects

In order to enable ILM for one or more maintenance objects, several steps are needed in both the configuration and in the database. This section describes some configuration enabled by default and some steps that must be taken in order to fully implement ILM.

There is some configuration enabled by default, but it won't be used unless ILM is fully configured. Each maintenance object that the product has configured for ILM has the following provided out of the box:

- **Special Table Columns:** Maintenance objects that support ILM include two specific columns: ILM Archive Switch (**ILM_ARCH_SW**) and ILM Date (**ILM_DT**).
- **Crawler Batch Process:** A “crawler” batch process is provided for each maintenance object that supports ILM and it is plugged into the MO as an option. Refer to [Batch Processes](#) for more information.
- **ILM Eligibility Algorithm:** Each maintenance object that is configured for ILM defines an [eligibility algorithm](#) that executes the logic to set the ILM Archive switch appropriately. This is plugged in to the MO algorithm collection.

If an implementation decides to implement ILM, there are steps that need to be followed, which are highlighted below.

Create the Master Configuration Record

The first step when enabling ILM is to create the **ILM Configuration** master configuration record. Navigate using **Admin Menu > Master Configuration**.

The Master Configuration for ILM Configuration defines global parameters for all ILM eligible Maintenance Objects. For example, the Default Retention Period. In addition, your product may implement additional configuration. Refer to the inline help for specific details about the information supported for your product's ILM configuration.

In addition, the user interface for this master configuration record displays summary information about all the maintenance objects that are configured to use ILM.

Confirm the Maintenance Objects to Enable

In viewing the list of maintenance objects that support ILM in the ILM master configuration page, your implementation may choose to enable ILM for only a subset of the supported maintenance objects. For example, some of the maintenance objects may not be relevant for your implementation. Or perhaps, the functionality provided by the maintenance object is used, but your implementation does not expect a high volume of data.

For each maintenance object that your implementation has confirmed for ILM, the following steps should be taken:

- Determine if the maintenance object should have a different default retention days than the system wide value defined on the master configuration. If so, use the MO option **ILM Retention Period in Days** to enter an override option for this maintenance object.
- Review the functionality of the ILM Eligibility algorithm provided by the product for this maintenance object. Each algorithm may support additional configuration based on business needs. If your organization has special business rules that aren't satisfied by the algorithm provided by the product, a custom algorithm may be provided to override the base algorithm.

For each maintenance object that your implementation does not want to enable for ILM, inactivate the eligibility algorithm. This will ensure that the ILM Crawler Initiator background process does not submit the crawler batch job for the maintenance object in question.

- Go to the [Maintenance Object — Algorithm](#) tab for each maintenance object and take note of the **ILM Eligibility** algorithm code.
- Go to the [Maintenance Object — Option](#) tab for the same maintenance object and add an option with an option type of **Inactivate Algorithm** and the value set to the ILM eligibility algorithm noted in the previous step.

Database Administrator Tasks

In order to implement ILM for each of the maintenance objects determined above, your database administrator must perform several steps in the database for the tables related to each MO. The following points are a summary of those steps. More detail can be found in the Information Lifecycle Management section of your product's *Database Administration Guide*.

- **Initializing ILM Date:** Your existing tables that are enabled for ILM may not have the ILM Date and ILM Archive switch initialized on all existing records. When choosing to enable ILM, a first step is to initialize this data based on recommendations provided in the DBA guide.
- **Referential Integrity:** The recommended partitioning strategy for child tables in a maintenance object is referential partitioning. In order to implement this, database referential integrity features must be enabled.
- **Partitioning:** This provides a way in which the data can segregate into multiple table partitions and will help in better management of the lifecycle of the data.

Schedule the ILM Crawler Initiator

The final step of enabling the system for ILM is to schedule the ILM crawler initiator **F1-ILMIN** regularly based on your implementation's need. It is recommended to only schedule this batch process once all the required database activities are complete.

Ongoing ILM Tasks

For an environment where ILM is enabled, besides the periodic execution of the ILM crawler batch processes to review and mark records, your database administrator has ongoing tasks.

The DBA reviews and maintains partitions and identifies partitions that may warrant some type of archiving step. The Information Lifecycle Management section of your product's *Database Administration Guide* provides more information for your DBA.

Archived Foreign Keys

If your DBA choose to archive a partition, there may be records in the system that refer to one of the archived records as a foreign key.

When a user attempts to view a record using a hyperlink or drill down mechanism, if the implementation of the navigation uses FK Reference functionality, the system will first check if the record exists. If not, it will display a message to the user indicating that the record has been archived.

Configuration Tools

This section describes tools to facilitate detailed business configuration. The configuration tools allow you to extend both the front-end user interface as well as create and define specialized back-end services.

Business Objects

A [maintenance object](#) defines the physical tables that are used to capture all the possible details for an entity in the system. A business object is tool provided to further define business rules for a maintenance object.

This section provides an overview of business objects and describes how to maintain them.

The Big Picture of Business Objects

The topics in this section describe background topics relevant to business objects.

What Is A Business Object?

A business object (BO) is a powerful tool used throughout the system. For many maintenance objects, a BO is a key attribute of the record used to define the data it captures, its user interface behavior and its business rules. Some business objects support the definition of a lifecycle, capturing different states that a record may go through, allowing for different business rules to be executed along the way. This type of business object is considered the “identifying” or “governing” business object. We will see later that other types of BOs exist that are different from the “identifying” business object.

The use of business objects allows for extensibility and customization of product delivered maintenance objects. There are many options to adjust the behavior of base delivered business objects. In addition, implementations may introduce their own business objects if the base product delivered objects do not meet their business needs.

NOTE: Not all maintenance objects in the product support business objects as a “identifying” or “governing” tool. This is the standard going forward for new maintenance objects. However, there are some maintenance objects created before this became a standard.

A Business Object Has a Schema

A business object has elements. The elements are a logical view of fields and columns in one of the maintenance object’s tables. The structure of a business object is defined using an XML schema. The main purpose of the schema is to identify all the elements from the maintenance object that are included in the business object and map them to the corresponding maintenance object fields. Every element in the BO schema must be stored somewhere in the maintenance object. The BO may not define elements that are derived.

When defining elements for the primary table or the language table (for an administrative object) there is no need to define the name of the physical table in the schema. The system infers this information. The following is a snippet of a schema:

```
<schema>
  <migrationPlan mapField="MIGR_PLAN_CD" suppress="true" isPrimeKey="true"/>
  <bo mapField="BUS_OBJ_CD" fkRef="F1-BUSOB"/>
  <customizationOwner mapField="OWNER_FLG" suppress="input"/>
  <version mapField="VERSION" suppress="true"/>
  <description mapField="DESCR"/>
  <longDescription mapField="DESCRLONG"/>
  ...

```

Many maintenance objects have child table collections (e.g., a collection of names for a person, or a collection of persons on an account). Depending on the requirements, the business object may define the full collection such that the user will maintain the information in a grid. However, the schema also supports “flattening” records in a child table so that they can be treated as if they were singular elements. The following are examples of each:

Example of a child table. This is a snippet of the Instructions collection on the migration plan business object. You can see that the list attribute defines the child table and all elements within it map to the appropriate column in that table.

```
<migrationPlanInstruction type="list" mapChild="F1_MIGR_PLAN_INSTR">
  <migrationPlan mapField="MIGR_PLAN_CD" suppress="true"/>
  <sequence mapField="PLAN_INSTR_SEQ" suppress="true"/>
  <instructionSequence mapField="INSTR_SEQ"/>
  <instructionType mapField="INSTR_TYPE_FLG"/>
  <parentInstructionSequence mapField="PARENT_INSTR_SEQ"/>
  <businessObject mapField="BUS_OBJ_CD" fkRef="F1-BOMO"/>
  ...

```

Example of a simple “flattened” field. The business object for Status Reason includes an element called Usage, which maps to a pre-defined characteristic of type **F1-SRUSG**. The “row” defines which child table is being flattened and the attributes of the row in that child that uniquely identify it.

```
<usage mdField="STATUS_RSN_USAGE" mapField="CHAR_VAL">
  <row mapChild="F1_BUS_OBJ_STATUS_RSN_CHAR">
    <CHAR_TYPE_CD is="F1-SRUSG"/>
    <SEQ_NUM is="1"/>
  </row>
</usage>

```

Example of a “flattened row”. This business object for Account includes a single row for the Person collection where only the “financially responsible, main” customer is defined. The “accountPerson” attribute defines one field from that row (the Person Id) and includes the ‘flattening’ criteria in the “row” information. In addition, a second field from that same row (“accountRelType”) is defined. Instead of having to repeat the flattening criteria, the “rowRef” attribute identifies the element that includes the flattening.

```
<accountPerson mapField="PER_ID">
  <row mapChild="CI_ACCT_PER">
    <MAIN_CUST_SW is="true"/>
    <FIN_RESP_SW default="true"/>
  </row>
</accountPerson>
<accountRelType mapField="ACCT_REL_TYPE_CD" rowRef="accountPerson" dataType="string"/>

```

Example of a “flattened list”. The business object for Tax Bill Type includes an list of valid algorithms for “bill completion”. The Sequence and the Algorithm are presented in a list. The list element identifies the child table and the ‘rowFilter’ identifies the information about the list that is common.

```
<taxBillCompletion type="list" mapChild="C1_TAX_BILL_TYPE_ALG">
  <rowFilter suppress="true" private="true">
    <TAX_BILL_TYPE_SEVT_FLG is="C1BC"/>
  </rowFilter>
  <sequence mapField="SEQ_NUM"/>
  <algorithm mapField="ALG_CD" fkRef="F1-ALG"/>
</taxBillCompletion>
```

In addition, many maintenance objects support an XML structure field within the entity. These fields may be of data type CLOB or XML. One or more business object elements may be mapped to the MO's XML structure field. These elements may be defined in different logical places in the business object schema based on what makes sense for the business rules. When updating the MO, the system builds a type of XML document that includes all the elements mapped to the XML structure and stores it in one column. The following is an example of elements mapped to an XML column:

```
<filePath mdField="F1_FILE_PATH" mapXML="MST_CONFIG_DATA" required="true"/>
<characterEncoding mdField="F1_CHAR_ENCODING" mapXML="MST_CONFIG_DATA"/>
```

Elements mapped to an MO XML structure field are typically ones that do not have to be exposed for SQL indexing or joining.

NOTE: If the MO's XML structure field is of the data type XML, the database will allow searching for records based on that data, assuming appropriate indexes are defined. If the MO's XML structure field is of the data type CLOB, indexing or joining to elements in this column via an SQL statement is not typically supported. Note that most MOs are currently using the CLOB data type for the XML structure column, if provided.

Some business objects may have child tables that allow data to be stored in an XML structure field. The schema language supports defining elements from those fields in your schema as well.

Besides including information about the physical “mapping” of the element to its appropriate table / field location in the maintenance object, the schema supports additional syntax to provide the ability to define basic validation and data manipulation rules, including:

- Identifying the primary key of the record or the primary key of the a row in a list.
- Identifying which elements are required when adding or changing a record.
- Default values when no data is supplied on an Add.
- For elements that are lookup values, the lookup may be specified to validate that the value of the element is a valid lookup value.
- For elements that are foreign keys to another table, the FK Reference may be specified to validate the data.

The system will check the validity of the data based on the schema definition obviating the need for any special algorithm to check this validation.

In addition, the schema language may include some attributes that are used to auto-render the view of the record on the user interface, such as the **suppress** attribute. Refer to [BO Defines its User Interface](#) for more information.

NOTE: Schema Definition Tips. A context sensitive "Schema Tips" zone appears when you open the [Business Object](#) page to assist you with the schema definition syntax. The zone provides a complete list of the XML nodes and attributes available to you when you construct a schema.

A business object's schema may include a subset of the fields and tables defined in the maintenance object. There are two reasons for this:

- The fields or tables may not be applicable to the type of record the business object is governing. For example, a field that is specific to gas may not be included on a Device business object that is specific to electric meters.

- The information is not maintained through the business object, but rather maintained separately. For example, many BO based maintenance objects include a Log table. The records in the log table are typically not included the BO because they are viewed and maintained separately from the business object.

A Business Object May Define Business Rules

A business object may define business rules that govern the behavior of entities of this type.

- Simple element-level validation is supported by schema attributes. Note that element-level validation is executed before any maintenance object processing. For more sophisticated rules you create **Validation** algorithms and associate them with your business object. BO validation algorithms are only executed after "core validation" held in the MO is passed.
- A **Pre-Processing** algorithm may be used to "massage" a business object's elements prior to any maintenance object processing. For example, although simple element-level defaulting is supported by schema attributes, you may use this type of algorithm to default element values that are more sophisticated.
- A **Post-Processing** algorithm may be used to perform additional steps such as creating a To Do Entry or add a log record as part of the business object logical transaction. These plug-ins are executed after all the validation rules are executed.
- An **Audit** algorithm may be used to audit changes made to entities of this type.

Any time a business entity is added, changed or deleted, the system detects and summarizes the list of changes that took place in that transaction and hands it over to **Audit** plug-ins associated with the business object. These plug-ins are executed after all the post-processing rules are executed. It is the responsibility of such algorithms to log the changes if and where appropriate, for example as a log entry or an entry in an audit trail table or an entry in the [business event log](#)

By default all elements of the business object are subject to auditing. You can however mark certain elements to be excluded from the auditing process using the **noAudit** schema attribute. Marking an element as not auditable will prevent it from ever appearing as a changed element in the business object's audit plug-in spot. In addition, if the only elements that changed in a BO are ones marked to not audit, the audit algorithm is not even called. Refer to the "Schema Tips" context sensitive zone associated with the Business Object maintenance page for more information on this attribute.

Refer to [Business Object - Algorithms](#) for more information on the various types of algorithms.

The system applies business object rules (schema based and algorithms) whenever a business object instance is added, changed or deleted. This is only possible when the call is made via the maintenance object service. For example, when made via business object interaction ("invoke BO"), the MO's maintenance page or inbound web services that reference the BO. In addition the system must be able to [determine the identifying business object](#) associated with the actual object being processed. If the business object cannot be determined for a maintenance object instance business object rules are not applied.

NOTE:

Pre-Processing is special. The pre-processing algorithm plug-in spot is unique in that it only applies during a BO interaction. It is executed prior to any maintenance object processing. It means that when performing add, change or delete via the maintenance object service, the pre-processing plug-in is not executed.

CAUTION: Direct entity updates bypass business rules! As mentioned above, it is the maintenance object service layer that applies business object rules. Processes that directly update entities not via the maintenance object service bypass any business object rules you may have configured.

FASTPATH: Refer to [BO Algorithm Execution Order](#) for a summary of when these algorithms are executed with respect to lifecycle algorithms.

The plug-in spots described above are available for all business objects and they are executed by the system when processing adds or updates to the business object. It is possible for a specific maintenance object to define a special plug-in spot for business objects of that MO. When this happens, the maintenance object identifies the special algorithm entity

lookup value as an [MO option](#): **Valid BO System Event**, causing the BO Algorithm collection to include that system event in its list.

A Business Object Defines its User Interface

One of the responsibilities of an identifying business object is to define its user interface rules for viewing and maintenance of its record. The standard implementation for maintaining a business object is that a [maintenance portal](#) is used to display a record. This portal includes a “map” zone that displays the information about the business object. To add or make changes to a record, the user clicks a button that launches a maintenance BPA script which displays a maintenance “map”.

The display and maintenance “maps” are driven by the business object. The BO may define a full [UI map](#) where all the information is displayed based on the map’s HTML. Note that for a child BO, the maps may be [inherited](#) by a parent BO (or any BO “up the chain”).

The standard going forward is to use schema definition and UI Hints to define user interface behavior so that a full UI map is not needed but rather the HTML is derived. The schema language includes some basic display attributes such as **label** and **suppress**. UI hints provide many additional tags and elements that allow dynamic generation of formatted UI Maps. For more complex behavior in the user interface, for example where javascript is needed, UI map fragments may be defined within the schema via UI hints. In this way only complex UI behavior warrants small snippets of javascript and HTML. However the rendering of standard fields can be dynamically rendered. UI map fragments also allow for derived fields to be included in the user interface.

A business object schema may include [data areas](#) for segments of its schema definition to allow for reusable components. In this case the data area would also include schema attributes and UI hints for the elements that it is including.

NOTE: Schema Definition Tips. The context sensitive "Schema Tips" zone appears when you open the [Business Object](#) page. This includes detailed information about supported UI hint definition syntax.

As mentioned in [Business Object Inheritance](#), schemas are not inherited on a child business object. As such, when using UI hints for automatic UI rendering, the child BO must define the full schema with all the definitions. A good business object hierarchy will be designed for reuse meaning that the child BO will include the parent BO schema or alternatively, the BO schemas will include reusable data areas.

Invoking A Business Object

We have talked about defining a business object. This section describes how business objects are used throughout the system to view, add and update records.

- Various parameters for zones that are used to display data in the system include support for retrieving data by referencing a business object. The zone code will “invoke” the BO, meaning that the record will be retrieved using the referenced BO.
- The system’s [scripting](#) language includes a step type to “invoke BO”. This allows for BPA scripts, service scripts and plug-in scripts to retrieve information and add or update records using BO interaction.
- [Inbound web services](#) may reference a business object in its operations collection. This allows external systems to add or update records in our product via web service interaction.

Often when configuring a zone or writing a script, the BO to use in the “invoke BO” statement should be the identifying BO of the record. As such, often the script will include steps prior to the “invoke BO” step to “[determine the identifying BO of the record](#)” and once the identifying BO is found, the script step will invoke that BO. Note that zones and inbound web services reference a BO directly. In each case, if the BO to use should be dynamic, then the zone / inbound web service should reference a service script that can perform the steps to identify the BO and then invoke that BO.

It should be noted however that the BO used in an “invoke BO” statement (or referenced in an inbound web service) **does not have to match** the identifying BO for the record. Here are some examples of where this may be true:

- A script may only require a subset of elements for a record and not the entire record. In this case, it is better for performance purposes to define a special BO (sometimes called a “lite” BO or a “mini” BO) that only defines the

needed elements. When the system retrieves the data, it will only access the tables that are included in the BO's schema definition. In addition, if there are no elements that map to an XML structure field, the system will skip any parsing of that column. Similarly, if a script is **updating** a subset of elements on a record, it may be beneficial to use a "mini" BO to do the updates.

NOTE: Please note the following with respect to using a mini BO. This BO is only used for its schema. This type of BO would not define algorithms or a lifecycle. Because the BO is special, it often should not be able to be used as any record's identifying BO. To control that, these BOs are often configured to not allow new instances. Refer to [Determine the Identifying BO](#) for more information.

- The maintenance object to be added or updated in a script may not support business objects as "identifying BOs". For example, Batch Control maintenance object does not have an identifying BO. However, scripts may still wish to retrieve data (or make updates) to these types of records. An easy way to achieve that goal is to define a business object and use "invoke BO" to access the data.

NOTE: Not all maintenance objects support being maintained through a business object interaction. This is true in a small number of older objects where the underlying maintenance service includes additional functionality besides simply updating the database tables. These maintenance objects are identified via the [MO option BO Maintenance](#), set to **N**.

- Some functionality may be trying to add or update records for a maintenance object in a 'physical' manner and do not want or need to use the object's identifying BO. Or the MO may not have an identifying BO. For example, revision control takes a snapshot of a record for audit purpose and to be able to restore a previous version. In this case, the system wants to capture a full "physical" view of the record. To do this, a special "physical" BO may be created that includes all (or most of) the columns and the child tables.

NOTE: Like the mini BO, the physical BO would not define algorithms or a lifecycle and should not be able to be used as any record's identifying BO. To control that, these BOs are often configured to not allow new instances. Refer to [Determine the Identifying BO](#) for more information.

NOTE: To reiterate, the BO referenced in the "invoke BO" statement or referenced in an inbound web service does not have to match the identifying BO and does not have to be configured to "allow new instances".

Determine the Identifying BO

As mentioned in other topics, the identifying BO is the business object that governs the business rules for a record. This is the business object that the record will be validated against when any additions or changes are made to the record as long as updates are made via the maintenance service. This includes using "invoke BO" for add or update, using inbound web service interaction and for access to the maintenance page service (via an old style fixed page or via a business service).

How does the system determine the identifying BO? An algorithm plugged into the maintenance object (the **Determine BO** plug-in spot) is responsible for this. If the maintenance object is not configured with an algorithm for this plug-in spot, or no BO is found by the algorithm, no BO business rules are applied.

Most maintenance objects in the system capture the record's identifying BO directly on the record. However, it is possible to define the identifying BO somewhere else. For example, there may be some maintenance objects that are master or transaction objects with an associated "type" object where the identifying BO is defined on its "type" object. Note that the standard Determine BO algorithm plugged into most maintenance objects (**F1-STD-DTMBO — Determine Standard Business Object**) checks for these two conditions.

There may also be cases where a single identifying BO is used for all BOs for a given MO. This may be an option used for some older maintenance object created prior to the business object functionality when implementations wish to introduce custom business rules that are common for all records of that MO. The product provides a base algorithm type (**F1-MOBO — Determine Specific Business Object**) that captures the BO as a parameter.

Base Business Objects

For each maintenance object (MO) that supports an “identifying” business object, the type of business object provided by the product depends on the functionality and expected use by implementations. The following are some common patterns.

- There are MOs where the product provides base BOs that implementations may use if applicable for their business rules. In addition, it is expected that implementation will define custom BOs to support their business needs. Good examples of this type of MO are any of the various “rule” MOs. For example, calculation rule in Oracle Utilities Customer Care and Billing or the usage rule in Oracle Utilities Meter Data Management or the form rule in Oracle Public Sector Revenue Management. The product provides business objects for common rules but each implementation could have special rules that they need to implement and will need to create custom business objects.
- There are MOs where the product provides base BOs that supply common behavior for an object. Implementations may find that supplied the business objects match their business requirements and use the BOs as is. It is expected, however that for many implementations, their business rules will require additional elements to be captured or have special rules to apply. In this case the base business objects may be extended. This scenario may apply to ‘master’ data objects in various products such as the Device or Meter or Tax Role.
- There are MOs where the product may deliver a base BO that is not expected to satisfy most implementations because different jurisdictions or different implementations will typically have their own rules. In this case the base delivered BO can be used as a template or starting point for custom defined BOs. Some examples of this are Rebate Claim in Oracle Utilities Customer Care and Billing or the Appeal object in Oracle Public Sector Revenue Management.
- There are MOs where the expectation is that every implementation will have different requirements for the type of data to capture and the product will not supply base BOs that can be used as the “identifying” BO. However, it may supply a “parent” BO that defines the lifecycle and many of the business rules that it expects all records to follow. In these scenarios, the implementations will create “child” BOs that will serve as the “identifying” BOs and refer to the base “parent” BO for many of its rules through [inheritance](#). Some examples of the are Tax Form in Oracle Public Sector Revenue Management or Activity in Oracle Utilities Mobile Workforce Management.
- There are some scenarios where the base product provides business objects and the expectation is that implementations will use the business objects as delivered with little or no customization. This is a case where the system used business objects to implement product functionality, not because there is an expectation that the implementers will extend the functionality, but because the business object model is the favored development tool even for the product. The objects delivered for Configuration Migration Assistant are an example.

NOTE: Not all maintenance objects in the product support business objects as a “identifying” or “governing” tool. This is the standard going forward for new maintenance objects. However, there are some maintenance objects created before this became a standard.

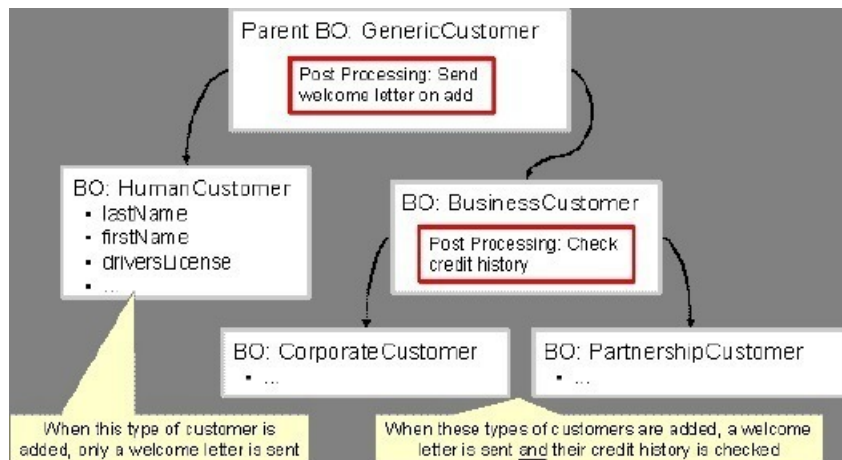
For all maintenance objects, the base product may provide additional BOs that are not meant to be “identifying” BOs, but instead are provided to support functionality to interact with the MO using the BO as a tool as described in [Invoking a BO](#).

- One or more “mini” or “lite” BOs may be supplied for a maintenance object. This may be found when the product has functionality to retrieve a subset of elements for the maintenance object via scripting or via a user interface.
- A “physical” BO may be supplied. This a BO that typically includes all tables and all fields of the maintenance object in there “physical” form. In other words, there is no “flattening” of child tables and any XML structure fields are defined as a single field. Physical BOs are used in system processing where the full record needs to be captured as is. Some functionality that uses a physical BO includes [bundling](#) and [revision control](#).

Business Object Inheritance

A business object may inherit business rules from another business object by referencing the latter as its parent. A child business object can also have children, and so on. A parent's rules automatically apply to all of its children (no compilation - it's immediate). A child business object can always introduce rules of its own but never remove or bypass an inherited rule.

The following is an illustration of multiple levels of business object inheritance.



Notice how the "Business Customer" business object extends its parent rules to also enforce a credit history check on all types of customers associated with its child business objects.

Most types of business object system events allows for multiple algorithms to be executed. For example, you can have multiple **Validation** algorithms defined for a business object. For these, the system executes all algorithms at all levels in the inheritance chain starting from the highest-level parent business object moving on to lower levels.

Other types of system events allows for a single algorithm to be executed. For example, you can only have one **Information** algorithm to format the standard description of a business object instance. For these, the system executes the one at the level nearest to the business object currently being processed.

NOTE: The parent and its children must reference the same maintenance object.

NOTE: Data structures are not inherited. While you can declare schemas on parent business objects, their children will not inherit them. A good practice is to design child business object schemas to **include** their parent business object's schema.

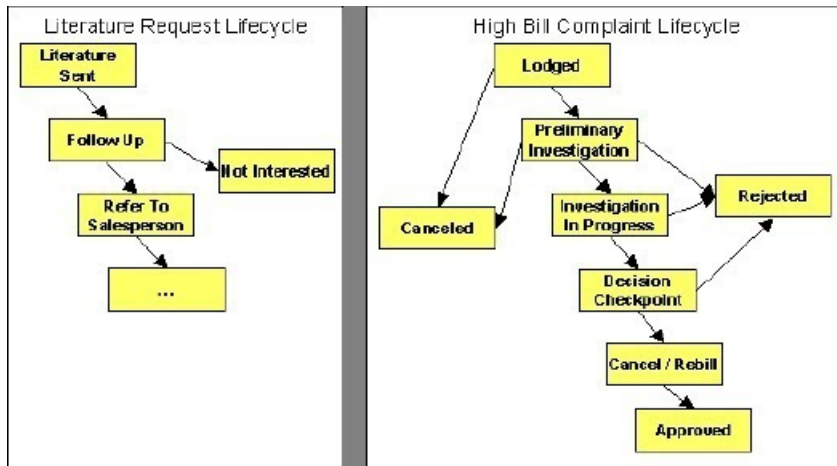
NOTE: User interface maps are inherited. When determining if the business object has a UI map to use for UI rendering, the system looks for display and maintenance maps linked to the BO as options. If the identifying BO does not have maps defined, the system follows "up the chain" of inheritance until it finds a map to use. This allows for a child BO to be used to extend business rules of a parent BO but inherit its user interface behavior. Refer to [Business Object Defines its User Interface](#) for more information.

NOTE: Use Inheritance Wisely. While it is intellectually attractive to abstract behavior into parent BOs to avoid redundant logic and simplify maintenance, before doing this weigh the reuse benefits against the cost in transparency, as it is not easy to maintain a complex hierarchy of business objects.

Each Business Object Can Have A Different Lifecycle

Many maintenance objects have a status column that holds the business entity's current state within its lifecycle. Rules that govern lifecycle state transition (e.g., what is its initial state, when can it transition to another state, etc.) and the behavior associated with each state are referred to as lifecycle rules. Older Maintenance Objects, such as To Do Entry, have predefined lifecycles whose rules are governed by the base-package and cannot be changed. The lifecycle of newer Maintenance Objects exists in business object meta-data and as such considered softly defined. This allows you to have completely different lifecycle rules for business objects belonging to the same maintenance object.

Here are examples of two business objects with different lifecycles that belong to the same maintenance object.



NOTE: A Maintenance Object supports soft lifecycle rules if it is defined with a **Status Field Maintenance Object** option.

The topics that follow describe important lifecycle oriented concepts.

Valid States versus State Transition Rules

The boxes in the above diagram show the potential valid states a business entity of the above business object can be in. The lines between the boxes indicate the state transition rules. These rules govern the states it can move to while in a given state. For example, the above diagram indicates a high bill complaint that's in the **Lodged** state can be either **Canceled** or moved into the **Preliminary Investigation** state.

When you set up a business object, you define both its valid states and the state transition rules.

One Initial State and Multiple Final States

When you set up lifecycle states, you must pick one as the initial state. The initial state is the state assigned to new entities associated with the business object. For example, the above high-bill complaint business object defines an initial state of **Lodged**, whereas the literature request one defines an initial state of **Literature Sent**.

You must also define which statuses are considered to be "final". Typically when an entity reaches a "final" state, its lifecycle is considered complete and no further processing is necessary.

NOTE: Allowing An Entity To Be "Reopened". You can set up your state transition rules to allow a business entity to be "reopened" (i.e., to be moved from a final state to a non-final state). Neither of the above examples allows this, but it is possible if you configure your business object accordingly.

State-Specific Business Rules

For each state in a business object's lifecycle, you can define the following types of business rules.

FASTPATH: Refer to [BO Algorithm Execution Order](#) for a summary of when these lifecycle algorithms are executed with respect to BO level algorithms.

Logic To Take Place When Entering A State

You can define algorithms that execute before a business entity enters a given state. For example, you could develop an algorithm that requires a cancellation reason before an entity is allowed to enter the *Canceled* state.

You can also incorporate state auto-transitioning logic within this type of algorithms. Refer to [auto-transition](#) for more information.

Logic To Take Place When Exiting A State

You can define algorithms that execute when a business entity exists a given state. For example, you could develop an algorithm that clears out error messages when a given entity exits the *Error* state.

Monitor Rules

You can define algorithms to monitor a business entity while it is in a given state. This type of logic is typically used to check if the conditions necessary to [transition](#) the entity to another state exist (and, if so, transition it). For example, transition an entity to the *Canceled* state if it's been in the **Error** state too long. Another common use is to perform ancillary work while an entity is in a given state. For example, update statistics held on the object while it's in the **Active** state.

[Monitor algorithms](#) are invoked when a business entity first enters a state and periodically after that in batch. You have the option to defer the monitoring of a specific state until a specific monitoring batch job runs. You do so by associating the state with a specific monitoring process. In this case the system will only execute the monitoring rules of this state when that specific batch process runs. This is useful when processing one type of record typically creates another type of record. You may want the processing of the second set of records to be deferred to a later time.

A monitor algorithm can carry out any business logic. In addition it can optionally tell the system to do either of the following:

- Stop monitoring and transition to another state. The system will not call any further monitoring algorithm plugged in on the state and attempt to transition the entity to the requested new state.
- Stop monitoring. Same as above except that no transition takes place. You may want to use this option to prevent transitions while some condition is true.

If none of the above is requested the system keeps executing subsequent monitoring algorithms.

FASTPATH: Refer to [Business Object - Lifecycle](#) for more information on how to set up state-specific algorithms.

Inheriting Lifecycle

If a business object references a parent business object, it always [inherits](#) its lifecycle from the highest-level business object in the hierarchy. In other words, only the highest-level parent business object can define the lifecycle and the valid state transitions for each state. Child business objects, in all levels, may still extend the business rules for a given state by introducing their own state-specific algorithms.

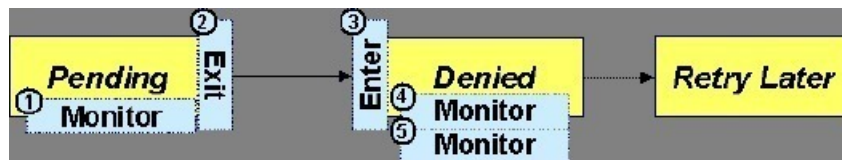
The system executes all state-specific algorithms at all levels in the inheritance chain starting from the highest-level parent business object moving on to lower levels.

Auto-Transition

In a single transition from one state to another, the system first executes the **Exit** algorithms of the current state, transitions the entity to the new state, executes the **Enter** algorithms of the new state followed by its **Monitor** algorithms. At this point if a **Monitor** algorithm determines that the entity should be further automatically transitioned to another state the remaining monitoring algorithm defined for the current state are NOT executed and the system initiates yet another transition cycle.

Notice that an **Enter** algorithm can also tell the system to automatically transition the entity to another state. In this case the remaining **Enter** algorithm as well as all **Monitor** algorithms defined for the current state are NOT executed.

The following illustration provides an example of an auto-transition chain of events.



In this example a business entity is in a Pending state. While in that state a **Monitor** algorithm determines to auto-transition it to the Denied state. At this point the following takes place:

- No further **Monitor** algorithms of the Pending state are executed
- Pending state **Exit** algorithms are executed
- The system transitions the entity to the Denied state
- Denied state **Enter** algorithms are executed. No further auto-transition is requested.
- Denied state **Monitor** algorithms are executed. No further auto-transition is requested.

Keeping An Entity In Its Last Successful State

By default, any error encountered while transitioning a business entity from one state to another rolls back ALL changes leaving the entity in its original state.

When applicable, the Maintenance Object can be configured to always keep an entity in its last successful state rather than rolling all the way back to the original state. This practice is often referred to as "taking save-points". In case of an error, the entity is rolled back to the last successfully entered state and the error is logged on the [maintenance object's log](#). Notice that with this approach no error is returned to the calling process, the error is just logged.

The logic to properly log the error is in a **Transition ErrorMaintenance Object plug-in**. The system considers a maintenance object to practice "save-points" when such an algorithm is plugged into it.

Even if the maintenance object practices "save-points", in case of an error the system will not keep an entity in the last successfully entered state if that state is either marked as [transitory](#) or one of its **Enter** algorithms has determined that the entity should proceed to a next state. The system will roll back to the first previous state that does not match these conditions.

Monitoring Batch Processes

The base package provides a periodic monitoring batch process for each maintenance object that supports soft state transition. The process periodically executes the monitoring algorithms associated with the current state of an entity, excluding states explicitly referencing a deferred monitoring batch process.

A deferred monitoring process works in the same way but only considers entities whose current state references this particular batch control as their monitor process. Deferred monitoring is only needed for certain states based on business requirements.

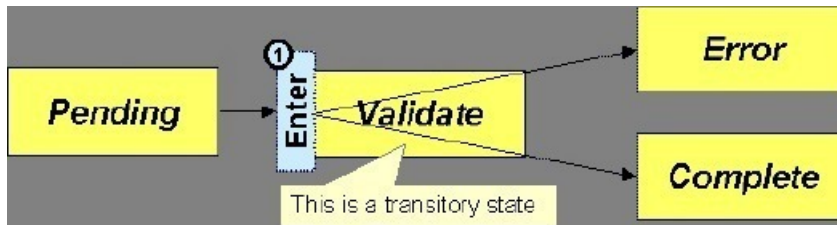
Your business rules will dictate the execution frequency of each monitoring process and the order in which they should be scheduled.

NOTE: Updates to the business object. When the monitor algorithms indicate that the business object should transition, the monitor batch processes are responsible for ensuring the business object is transitioned appropriately and that the appropriate exit, enter and monitor algorithms are executed. Please note that the business object is not updated using a call to the maintenance object service and therefore the [business rules](#) plugged in to the business object are not executed.

Transitory States

You can define a state as **Transitory** if you do not wish the business entity to ever exist in that particular state.

The following illustrates a lifecycle with a transitory Validate state.



In this example, the business entity is saved still not validated in the Pending state. At some point, the user is ready to submit the entity for validation and transitions it into a transitory state **Validate** whose **Enter** rules contain the validation logic. The responsibility of the transitory state's **Enter** algorithms is to decide if the entity is valid or in error and then transitions it into the appropriate final state. In this scenario, you may not ever want the business entity to exist in the **Validate** state.

Let's also assume that the maintenance object in this example is practicing "save-points" and requires the entity to be kept in its last successful state. If an error were to occur during the transition from **Validate** to the next state, the system would roll back the entity back to Pending, and not **Validate** even though the entity has successfully entered the **Validate** state. Refer to the [Auto Transition](#) section for more information.

State Transitions Are Audited

Most Maintenance Objects that support soft lifecycle definition also have a log to hold significant events throughout a business entity's lifecycle. For example, log entries are created to record:

- When the business entity is created (and who created it)
- When its status changes (and who changed it)
- If a transition error occurred (and the error message)
- References to other objects created throughout the entity's lifecycle. For example, if a To Do entry is created as part of processing the entity, the To Do Entry is referenced in the log.
- Manual entries added by a user (think of these as "diary" entries)

When a business entity is first created and when it transitions into a new state the system calls **Transition** algorithm(s) plugged in on the [Maintenance Object](#) to record these events. If the maintenance object supports a log these events can be captured as log entries.

NOTE: Most base package maintenance objects supporting a log may already provide state transition logging as part of their core logic. In this case you only need to provide a **Transition** plug-in if you wish to override base logging logic with your own.

Required Elements Before Entering A State

You can define additional elements that are required before a business entity can enter a given state. For example, let's assume that a Cancel Reason must be defined before an object can enter the *Canceled* state. You do this by indicating that element as a **Required Element** [state-specific option](#) on the appropriate state on the business object.

Capturing a Reason for Entering a State

Some business objects support configuring certain states to allow or require a status reason when an object enters the state. The product provides a centralized status reason table that may be used to define the valid BO status reasons for various business objects and various states. The status reasons are defined using the [Status Reason](#) portal.

The following sections provide additional information about the BO status reason functionality.

Maintenance Object Must Support Status Reason

In order for a business object to use the centralized status reason table to define reasons, the maintenance object must first support the status reason. MOs that support status reason have the following characteristics:

- The primary table includes a column for Status Reason. This represents the status reason for the record's current status, if applicable.
- The log table includes a column for Status Reason. The standard logic for capturing a log record when entering a state also captures the status reason, if applicable. This allows a user to review the history of the changes in status and the status reason captured for a previous state transition, if applicable.
- The maintenance object option collection includes an option that defines the Status Reason field. This setting is a trigger for business objects of this MO to be able to configure states to allow or require status reason.

Business Object State Indicates if Reasons are Applicable

Once the MO is configured to support status reason, configuration on the business object is required to indicate the states where a reason is applicable. States may be configured to require status reasons, allow status reasons as optional or not allow status reasons. With this configuration, the framework will automatically get the list of valid reasons for a state that allows or requires them and then prompt the user for a status reason when a manual state transition occurs for that state. It also automatically triggers an error if the state requires a status reason and no reason is provided.

NOTE: The status reason configuration on the business object state is customizable. That means that for a product owned business object, an implementation may opt to change the delivered configuration.

Status reasons are defined for the parent (or “lifecycle”) business object. All business objects in the hierarchy of the parent business object have the same valid reasons for their states.

The status reason code must be unique for the centralized status reason table. Business object and status are required fields, so it is not possible to share a common reason code (like “Not applicable”) across multiple business objects or states. If multiple BOs / states want to support a reason “Not applicable” then each must define a unique record for it. This point should be considered when planning for your status reasons.

Selectable vs. Not Selectable

When defining a status reason, you may indicate whether it's **Selectable** or **Not Selectable**. When a manual transition is performed and a user is prompted for a status reason, only the **Selectable** reasons are presented. The **Not Selectable** reasons may be defined to support transitions that occur via algorithm processing.

NOTE: The Selectable setting is customizable. That means that if a product provides a base owned status reason for a business object state, an implementation may opt to change whether it is selectable or not. Careful consideration should be made before changing a base delivered status reason from **Not Selectable** to **Selectable** as this may affect base provided algorithm functionality that could be relying on the setting of **Not Selectable**.

Status Reason Business Object

The status reason maintenance object, as with many maintenance objects in the product, references a business object used to define attributes and behavior related to defining status reasons. The framework provides a business object for status reason (**F1-BOStatusReason**). For the business objects that have states that require a status reason (let's call these “transactional BOs”), if there is some special logic required for defining the status reasons, it is possible to define a different status reason BO. In this situation, the override status reason BO to use for capturing status reasons should be defined as a BO option on the transactional BO using the **Status Reason Business Object** option type. If a transactional BO does not define any status reason BO option, then the **F1-BOStatusReason** is used when adding a status reason.

Defining a Usage

The base product status reason BO provides the ability to define a “usage” value. This is useful for algorithms that perform state transitions where a status reason is needed and where the algorithm is usable by more than one business object. In this case, the status reason to use cannot be provided as a parameter because each business object must define its own set of status reasons for each state. The Usage value can be used instead. Each business object can configure the status reason to use in the algorithm and set the appropriate usage value. The algorithm can reference the usage value and retrieve the correct status reason to use based on the record’s transactional BO.

The status reason business object provided with the framework product (**F1-BOStatusReason**) supports capturing a usage. The valid usage values are defined in the **Status Reason Usage** characteristic type.

Alternatives for Defining Reasons

There may be business objects in the system that capture reasons that are defined somewhere besides the BO status reason table. For example, some objects may have an explicit administrative table for status reasons. Some objects may use a Lookup or an Extendable Lookup to capture reasons. Refer to the business object description for information about how valid reasons are defined, if applicable.

If a business object supports a reason that is not related to a state transition (such as a creation reason), the BO status reason would not be used. One of the alternate methods for defining a reason, described above, would be used.

BO Algorithm Execution Summary

This table highlights the processing steps that occur when adding or changing a record that is governed by a business object.

Invoke BO	
Event	Comments
BO Pre-processing algorithms executed	These algorithms are only executed when Invoke BO is used. The business object in the Invoke BO is the one whose rules are executed.
MO Processing	
Event	Comments
Determine if status has changed.	The system keeps a note of the new status value but initially proceeds with the old value.
MO Processing.	Standard MO processing, including MO validation is executed.
Determine BO algorithm executed.	The MO level algorithm is executed to determine the identifying BO .
BO Validation algorithms executed.	
State transition rules are performed if the status has changed.	BO Status Exit algorithms for the “old” status executed. Status updated to the new value. BO Status Enter algorithms for the “new” status executed. If no error — MO Transition algorithms are executed. FASTPATH: Refer to State Transitions are Audited for more information. If error and there are “save points” the MO Transition Error algorithms are executed. FASTPATH: Refer to Keeping An Entity In Its Last Successful State for more information. Otherwise, the error is reported.
BO Status Monitor algorithms are executed.	If the record transitions again, the prior step (State transition rule step) is repeated for the new transition.
BO Post-processing algorithms are executed.	

NOTE: To emphasize, the steps in the MO Processing table are only executed when the maintenance object service is invoked. Any add or update initiated by an "invoke BO" statement will invoke the MO service. This is also true for web service that invoke the business object. Records processed by the [Monitor Batch Process](#) does not invoke the maintenance service. The monitor batch process only executes the monitor algorithms and the state transition rules (if the monitor algorithms indicate that a status change should occur).

NOTE: For records that do not have a status, the state transition rules and the monitor rules are not applicable.

Granting Access To Business Objects

Every business object must reference an [application service](#). When you link a business object to an application service, you are granting all users in the group access to its instances. You can prevent users from transitioning a business object into specific states by correlating each business object status with each application service action (and then don't give the user group rights to the related action).

FASTPATH: Refer to [The Big Picture Of Application Security](#) for information about granting users access rights to an application service.

The system checks if a user has access rights each time the application is invoked to add, change, delete, read, or transition a business object. However, if a business object invokes another business object, we assume that access was controlled by the initial business object invocation and we do not check access for other business objects that it invokes. In other words, access rights are only checked for the initial business object invoked in a service call.

In order to apply business object security the system must be able to determine the business object associated with the actual object being processed. To do that the Maintenance Object itself has to have a **Determine BO** algorithm [plugged in](#). If this algorithm is not plugged in or it cannot determine the BO on the MO, the system will NOT invoke any BO rules. If the business object cannot be determined for a maintenance object instance, business object security is not checked. In this case the system checks the user's access rights using standard maintenance object security.

NOTE: Parent business objects are ignored. If a child business object exists, a user need only have access to the child business object's application service (not to every application service in the business object hierarchy).

Defining Business Objects

The topics in this section describe how to maintain business objects.

Note that several context sensitive dashboard zones appear on this page and are visible on all tabs.

- **Schema Tips.** This zone provides several links to view details related to the valid schema syntax and UI Hint syntax.
- **View UI Rendering.** This zone provides buttons to view the automatic rendering of the Display map or Input map based on the attributes defined in the schema, including UI hints.
- **Generate Schema.** This zone includes a button that can be used to generate a "physical" schema based on the maintenance object definition. The element names are taken from the Java field name for each column. Once generated, adjust the schema as desired.
- **Create a BO Algorithm.** This zone includes a button to create a script based algorithm related to this business object. You are then prompted for information regarding the plug-in spot (BO or BO Status) and the system event, the name, description, etc. Once all the information is provided, the system creates an algorithm type, algorithm, links the algorithm to the business object, creates the script and brings you to the script step to start defining the logic for the plug-in script.

- **BOs Linked to the MO.** This zone displays other business objects for the same maintenance object as the BO currently displayed. You may drill into any of the other BOs by clicking its description.

Business Object - Main

Use this page to define basic information about a business object. Open this page using **Admin > System > Business Object**

Description of Page

Enter a unique **Business Object** name and **Description**. Use the **Detailed Description** to describe the purpose of this business object in detail. **Owner** indicates if this business object is owned by the base package or by your implementation (**Customer Modification**).

CAUTION: Important! If you introduce a new business object, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

Enter the **Maintenance Object** that is used to maintain objects of this type.

Enter a **Parent Business Object** from which to [inherit](#) business rules.

Lifecycle Business Object is only displayed for child business objects, i.e. those that reference a parent business object. It displays the highest-level business object in the inheritance hierarchy. Refer to [Inheriting Lifecycle](#) for more information.

Application Service is the application service that is used to provide security for the business object. Refer to [Granting Access To Business Objects](#) for more information. The application service on the child business object must have the same valid actions as the application service on the parent business object.

Use **Instance Control** to allow or prevent new entities from referencing the business object. Typically only the [identifying BOs](#) are marked to allow new instances.

Click the **View Schema** hyperlink to view the business object's expanded schema definition. Doing this opens the [schema viewer](#) window.

Click the **View XSD** hyperlink to view the business object's expanded schema definition in XSD format.

Click the **View MO** hyperlink to view the maintenance object in the [Maintenance Object Viewer](#). You may find it useful to leave the application viewer window open while defining your business object schema.

The options grid allows you to configure the business object to support extensible options. Select the **Option Type** dropdown to define its **Value**. **Detailed Description** may display additional information on the option type. Set the **Sequence** to **1** unless the option can have more than one value. **Owner** indicates if this option is owned by the base package or by your implementation (**Customer Modification**).

NOTE: You can add new options types. Your implementation may want to add additional option types. For example, your implementation may have plug-in driven logic that would benefit from a new option. To do that, add your new values to the customizable lookup field **BUS_OBJ_OPT_FLG**. If you add a new option type for a business option, you must update its maintenance object to declare this new option type. Otherwise, it won't appear on the option type dropdown. You do that by referencing the new option type as a **Valid BO Option Type**[maintenance object option](#).

Where Used

Follow this link to open the data dictionary to view the tables that reference [FI_BUS_OBJ](#).

Business Object - Schema

Use this page to maintain a business object's schema. Open this page using **Admin > System > Business Object** and then navigate to the **Schema** tab.

Description of Page

The contents of this section describe the zones that are available on this portal.

The **General Information** zone displays main attributes of the business object.

Click the **View Schema** hyperlink to view the business object's expanded schema definition. Doing this opens the [schema viewer](#) window.

Click the **View XSD** hyperlink to view the business object's expanded schema definition in XSD format.

The [Schema Designer](#) zone allows you to edit the business object's schema. The purpose of the schema is to describe the business object's properties and map them to corresponding maintenance object fields.

NOTE: Generating a Schema A context sensitive "Generate Schema" zone is associated with this page. The zone provides a button that allows the user to generate a basic schema that includes all the fields for all the tables for the BO's maintenance object.

NOTE: Schema Definition Tips. A context sensitive "Schema Tips" zone is associated with this page. The zone provides a complete list of the XML nodes and attributes available to you when you construct a schema.

NOTE: View UI Rendering. A context sensitive "View UI Rendering" zone is associated with this page. The zone is useful for [business objects that define the user interface detail](#) using schema attributes and UI Hints. The buttons allow you to view the automatically rendered display and input maps.

The **Schema Usage Tree** zone summarizes all cross-references to this schema. These may be other schemas, scripts, and web services. For each type of referencing entity, the [tree](#) displays a summary node showing a total count of referencing items. The summary node appears if at least one referencing item exists. Expand the node to list the referencing items and use their description to navigate to their corresponding pages.

Business Object - Algorithms

Use this page to maintain a business object's algorithms. Open this page using **Admin > System > Business Object** and then navigate to the **Algorithms** tab.

Description of Page

The **Algorithms** grid contains algorithms that control important functions for entities defined by this business object. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).
- Specify the **Sequence Number** and **Algorithm** for each system event. You can set the **Sequence Number** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.
- If the algorithm is implemented as a script, a link to the **Script** is provided. Refer to [Plug-In Scripts](#) for more information.
- **Owner** indicates if this is owned by the base package or by your implementation (**Customer Modification**).

The following table describes each **System Event**. Refer to [A Business Object May Define Business Rules](#) for more information about these system events.

System Event	Optional / Required	Description
Audit	Optional	Algorithms of this type may be used to audit certain changes made to business object instances. The system hands over to the algorithms a summary of all the elements that were changed throughout a specific call to update

System Event	Optional / Required	Description
Information	Optional	<p>an object. Excluded from this processing are elements explicitly marked on the schema as requiring no audit. For each element its original value before the change as well as its new value are provided.</p> <p>It is the responsibility of the algorithms to record corresponding audit information.</p> <p>The system invokes all algorithms of this type defined on the business object's inheritance hierarchy. Refer to Business Object inheritance for more information.</p> <p>Click here to see the algorithm types available for this system event.</p>
Post-Processing	Optional	<p>We use the term "Business Object Information" to describe the basic information that appears throughout the system to describe an entity defined by the business object. The data that appears in this information description is constructed using this algorithm.</p> <p>The system invokes a single algorithm of this type. If more than one algorithm is plugged-in the system invokes the one with the greatest sequence number found on the business object closest to the current business object in the inheritance hierarchy. Refer to Business Object inheritance for more information.</p> <p>Click here to see the algorithm types available for this system event.</p>
Pre-Processing	Optional	<p>Algorithms of this type may be used to perform additional business logic after a business object instance has been processed.</p> <p>The system invokes all algorithms of this type defined on the business object's inheritance hierarchy. Refer to Business Object inheritance for more information.</p> <p>Click here to see the algorithm types available for this system event.</p>
Validation	Optional	<p>Algorithms of this type further populates a request to maintain a business object instance right before it is processed.</p> <p>The system invokes all algorithms of this type defined on the business object's inheritance hierarchy. Refer to Business Object inheritance for more information.</p> <p>Click here to see the algorithm types available for this system event.</p>
Validation	Optional	<p>Algorithms of this type may be used to validate a business object instance when added, updated or deleted.</p> <p>The system invokes all algorithms of this type defined on the business object's inheritance hierarchy. Refer to Business Object inheritance for more information.</p> <p>Click here to see the algorithm types available for this system event.</p>

NOTE: Generate Algorithm. A context sensitive "Generate a BO Algorithm" zone is associated with this page. Refer to [Defining Business Objects](#) for more information about this zone.

NOTE: You can add new system events. Your implementation may want to add additional business object oriented system events. For example, your implementation may have plug-in driven logic that would benefit from a new system

event. To do that, add your new values to the customizable lookup field **BO_SEVT_FLG**. If you add a new business object system event, you must update the maintenance object to declare this new system event. Otherwise, it won't appear on the system event dropdown. You do that by referencing the new system event as a **Valid BO System Event maintenance object option**.

NOTE: You can inactivate algorithms on base Business Objects. Your implementation may want to use a business object provided by the base product, but may want to inactivate one or more algorithms provided by the base business object. To do that, on the business object where this algorithm is referenced, go to the options grid on Business Object - Main and add a new option, setting the option type to **Inactive Algorithm** and setting the option value to the algorithm code.

Business Object - Lifecycle

Use this page to maintain a business object's lifecycle oriented business rules and options. Open this page using **Admin > System > Business Object** and then navigate to the **Lifecycle** tab.

Description of Page

The **Status** accordion contains an entry for every status in the object's [lifecycle](#). The entry appears differently for a child business object as it can only extend its inherited lifecycle by introducing algorithms and options of its own.

Use **Status** to define the unique identifier of the status. This is NOT the status's description, it is simply the unique identifier used by the system. Only the highest-level business object can define lifecycle statuses. For a child business object the inherited status description is displayed allowing navigation to the corresponding entry on the business object defining the lifecycle.

Use **Description** to define the label of the status. This field is hidden for a child business object.

Use **Access Mode** to define the action associated with this status. Refer to [Access Rights](#) for the details of how to use this field to restrict which users can transition a business entity into this state. This field is hidden for a child business object.

Enter a **Monitor Process** to defer the monitoring of entities in this state until the specific batch process runs. Refer to [Monitor Rules](#) for more information. This field is hidden for a child business object.

The **Status Reason** dropdown indicates if users should be prompted to provide a specific reason when the business object enters this state. This field appears only if the Status Reason Field is configured as an option on the business object's maintenance object. Valid values are blank, **Optional**, and **Required**. The default value is blank (users are not prompted to provide a status reason). See [Configuring Status Reasons](#) for more information about status reasons.

Use **Status Condition** to define if this status is an **Initial**, **Interim** or **Final** state. Refer to [One Initial State and Multiple Final States](#) for more information about how this field is used. This field is hidden for a child business object.

Use **Transitory State** to indicate whether a business entity should ever exist in this state. Only **Initial** or **Interim** states can have a transitory state value of **No**. Refer to [transitory states](#) for more information. This field is hidden for a child business object.

Use **Alert Flag** to indicate that being in this state warrants an application alert. This may be used by custom logic to provide an alert to a user that entities exist in this state. This field is hidden for a child business object.

Use **Display Sequence** to define the relative order of this status for display purposes. For example when displayed on the status accordion and on the summary tab page. This field is hidden for a child business object.

Algorithms

The **Algorithms** grid contains algorithms that control important functions for a given status. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).

- Specify the **Sequence Number** and **Algorithm** for each system event. You can set the **Sequence Number** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.
- If the algorithm is implemented as a script, a link to the **Script** is provided. Refer to [Plug-In Scripts](#) for more information.
- **Owner** indicates if this is owned by the base package or by your implementation (**Customer Modification**).

The following table describes each **System Event**.

System Event	Optional / Required	Description
Enter	Optional	<p>Algorithms of this type apply business rules when a business object instance enters a given state.</p> <p>The system invokes all algorithms of this type defined on the business object's inheritance hierarchy. Refer to Business Object Inheritance for more information.</p> <p>Click here to see the algorithm types available for this system event.</p>
Exit	Optional	<p>Algorithms of this type apply business rules when a business object instance exits a given state.</p> <p>The system invokes all algorithms of this type defined on the business object's inheritance hierarchy. Refer to Business Object Inheritance for more information.</p> <p>Click here to see the algorithm types available for this system event.</p>
Monitor	Optional	<p>Algorithms of this type monitor a business object instance while in a given state. Typically these are used to auto-transition it to another state.</p> <p>The system invokes all algorithms of this type defined on the business object's inheritance hierarchy. Refer to Business Object Inheritance for more information.</p> <p>Click here to see the algorithm types available for this system event.</p>

NOTE: Generate Algorithm. A context sensitive "Generate a BO Algorithm" zone is associated with this page. Refer to [Defining Business Objects](#) for more information about this zone.

NOTE: You can inactivate status level algorithms on base Business Objects. Your implementation may want to use a business object provided by the base product, but may want to inactivate one or more of the status oriented algorithms provided by the base business object. To do that, on the business object and status where this algorithm is referenced, go to the options grid and add a new option, setting the option type to **Inactive Algorithm** and setting the option value to the algorithm code.

Next Statuses

Use the **Next Statuses** grid to define the valid statuses a business entity can transition into while it's in this state. This section is hidden for a child business object. Refer to [Valid States versus State Transition Rules](#) for more information. Please note the following about this grid:

- **Status** shows the statuses for the top-level business object, the **Status Code**, the **Lifecycle BO description**, and the **Status description** for each status.
- Use **Action Label** to indicate the verbiage to display on the button used to transition to this status.

- **Sequence** controls the relative order of one status compared to others for display purposes. This information may be used to control the order in which buttons are presented on a user interface.
- **Default** controls which next state (if any) is the default one. This information may be used by an **Enter** or **Monitor** algorithm to determine an auto-transition to the default state. It may also be used to also mark the associated button as the default one on a user interface.
- **Transition Condition** may be configured to identify a common transition path from the current state. By associating a given "next status" with a transition condition value, you can design your auto-transition rules to utilize those flag values without specifying a status particular to a given business object. Thus, similar logic may be used across a range of business objects to transition a business entity into, for example, the next **Ok** state for its current state. You'll need to add your values to the customizable lookup field **BO_TR_COND_FLG**.
- **Transition Role** controls whether only the **System** or both **System and User** have the ability to transition a business entity into a given "next status".
- When you initially set up a business object lifecycle, none of the statuses will reside on the database and therefore you can't use the search to define a "next status". We recommend working as follows to facilitate the definition of this information:
 - Leave the Next Statuses grid blank when you initially define a business object's statuses
 - After all statuses have been saved on the database, update each status to define its Next Statuses (this way, you can use the search to select the status).

Options

The options grid allows you to configure the business object status to support extensible options. Select the **Option Type** drop-down to define its **Value**. **Detailed Description** may display additional information on the option type. Set the **Sequence** to **1** unless the option can have more than one value. **Owner** indicates if this option is owned by the base package or by your implementation (**Customer Modification**).

NOTE: You can add new options types. Your implementation may want to add additional option types. For example, your implementation may have plug-in driven logic that would benefit from a new option. To do that, add your new values to the customizable lookup field **BO_OPT_FLG**. If you add a new option type for a status, you must update the business object's maintenance object to declare this new option type. Otherwise, it won't appear on the option type dropdown. You do that by referencing the new option type as a **Valid BO Status Option Type** [maintenance object option](#).

Business Object - Summary

This page summarizes business object information in a high level. Open this page using **Admin > System > Business Object > Search** and then navigate to the **Summary** tab.

Description of Page

The contents of this section describe the zones that are available on this portal.

The **General Information** zone displays main attributes of the business object.

Click the **View Schema** hyperlink to view the business object's expanded schema definition. Doing this opens the [schema viewer](#) window.

Click the **View XSD** hyperlink to view the business object's expanded schema definition in XSD format.

The **Business Object Hierarchy** zone displays in a tree view format the [hierarchy](#) of child business object associated with the current business object. It also shows the current business object's immediate parent business object.

For business objects with a lifecycle, the **Lifecycle Display** zone shows a graphical depiction of the lifecycle. Refer to the inline help of that zone for more information.

The **Options** zone summarizes business object and state specific options throughout the [inheritance](#) chain.

The **Rules** zone summarizes business object and state specific rules throughout the [inheritance](#) chain.

Defining Status Reasons

Status Reasons are used to provide more information about why a business object transitioned to a given state. The status reason table provides a centralized place where status reasons can be defined across many different business objects and states.

NOTE: Refer to [Defining Reasons for Entering a State](#) for overview information.

If a business object has one or more states that are configured to capture a status reason, you may configure the valid reasons by navigating to the status reason portal using **Admin > System > Status Reason**.

The topics in this section describe the base-package zones that appear on the Status Reason portal.

Business Objects with Status Reason List

This zone displays the business objects that have one or more status values that allow status reasons to be defined.

Click the broadcast icon to open other zones that contain more information about the business object's status reasons.

Status Reasons

The **Status Reasons** zone contains a list of the existing status reasons for the broadcasted business object.

Business Services

A business service is used to expose a back-end service so that it may be invoked by a script or a zone or a map to retrieve information or perform functions, depending on the related service.

As with the business object, the business service's interface to the internal service is defined using its schema. The schema maps the business service's elements to the corresponding elements in the internal service program's XML document. Just as a business object can simplify the schema of its maintenance object by only defining elements that it needs and "flattening" entries in a child collection to be defined as a singular element, a business service schema may simplify its service XML in a similar way.

NOTE: Schema Definition Tips. A context sensitive "Schema Tips" zone appears when you open the [Business Service](#) page to assist you with the schema definition syntax. The zone provides a complete list of the XML nodes and attributes available to you when you construct a schema.

[Inbound web services](#) and [scripts](#) support interaction with business services. You can also invoke a business service from a Java class.

Service Program

This transaction defines services available in the system. These include user interface services as well as stand-alone services that perform a specific function. A service may be referenced by a business service. Use this transaction to view existing service and introduce a new stand-alone service to be made available to a Business Service.

Select **Admin > System > Service Program** to maintain service programs.

Description of Page

Service Name is the unique identifier of the service.

CAUTION: Important! When adding new service programs, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

Owner indicates if this service is owned by the base package or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add a service. This information is display-only.

Description describes the service.

Service Type indicates whether the service is a **Java Based Service** or a **Java (Converted) Service**.

This **Program Component** grid shows the list of program user interface components associated with the service. For a stand-alone service, this list is typically not applicable.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_MD_SVC](#).

Defining Business Services

The topics in this section describe how to maintain business services.

Note that several context sensitive dashboard zones appear on this page and are visible on all tabs.

- **Schema Tips.** This zone provides several links to view details related to the valid schema syntax.
- **Generate Schema.** This zone includes a button that can be used to generate the schema based on the XML of its related Service. Once generated, adjust the schema as desired.

Business Service - Main

Use this page to define basic information about a Business Service. Open this page using **Admin > System > Business Service**

Description of Page

Enter a unique **Business Service** name and **Description**. Use the **Detailed Description** to describe the purpose of this business service in detail. **Owner** indicates if the business service is owned by the base package or by your implementation (**Customer Modification**).

CAUTION: Important! If you introduce a new business service, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

Enter the internal **Service Name** being called when this business service is invoked.

Enter the **Application Service** that is used to provide security for the business service. The application service must have an Access Mode of Execute.

Click the **View Schema** to view the business service's expanded schema definition. Doing this opens the [schema viewer](#) window.

Click the **View XSD** hyperlink to view the business object's expanded schema definition in XSD format.

Click the **View XML** hyperlink to view the XML document used to pass data to and from the service in the [Service XML Viewer](#). You may find it useful to leave the application viewer window open while defining your business service schema.

NOTE: XML document may not be viewable. If you create a new service program and do not regenerate the application viewer, you will not be able to view its XML document.

Where Used

Follow this link to open the data dictionary to view the tables that reference [FI_BUS_SVC](#).

Business Service - Schema

Use this page to maintain a Business Service's schema and to see where the Business Service is used in the system. Open this page using **Admin > System > Business Service** and then navigate to the **Schema** tab.

Description of Page

The contents of this section describe the zones that are available on this portal.

The **General Information** zone displays main attributes of the business service.

The [Schema Designer](#) zone allows you to edit the business service's schema. The purpose of the schema is to map the business service's elements to the corresponding fields of the back-end service program it rides on.

NOTE: Generating a Schema A context sensitive "Generate Schema" zone is associated with this page. The zone provides a button that allows the user to generate a basic schema that includes all the elements that are found in the XML of the BS's service.

NOTE: Schema Definition Tips. A context sensitive "Schema Tips" zone is associated with this page. The zone provides a complete list of the XML nodes and attributes available to you when you construct a schema.

The **Schema Usage Tree** zone summarizes all cross-references to this schema. These may be other schemas, scripts, and web services. For each type of referencing entity, the [tree](#) displays a summary node showing a total count of referencing items. The summary node appears if at least one referencing item exists. Expand the node to list the referencing items and use their description to navigate to their corresponding pages.

User Interface (UI) Maps

The User Interface (UI) map holds HTML to be rendered within [portal zones](#) and [Business Process Assistant \(BPA\) scripts](#). UI maps allow your implementation to create input forms and output maps that closely match your customer's business practices. In other words, the UI Map is designed to facilitate the capture and display of your [business objects](#) and [business services](#).

The UI map is a repository for a single HTML document paired with an XML schema where the schema defines the data that the HTML document displays and/or modifies. The UI Map HTML gives you the ability to craft the display by any method that an html document can support, including JavaScript and full CSS functionality.

Configuration tool support for UI Maps hinges around the ability to inject and extract an XML document from the HTML. For more information on the specialized support for HTML and JavaScript functionality - please refer to the [HTML tips document](#) (more on this below).

```

<html>
<head>
<title>Output Personal Information</title>
<link rel="stylesheet" type="text/css" href="cm_templates/cmStyles.css"/>
</head>
<body>

<table width="100%" border="0" cellpadding="2" style="margin-top:15px;" >

  <tr>
    <td/>
    <td/> <!-- locate the edit button on the bottom of the third column -->
    <td rowspan="99" style="vertical-align:bottom; margin-left:5px">
      <input type="button" value="edit" onClick="oraRunScript('HumanInfoU','personId');"/>
    </td>
  </tr>

  <tr>
    <td class="outputLabel">Name:</td>
    <td><span oraField="name" class="outputData"></span></td>
  </tr>
  <tr>
    <td class="outputLabel">Home Phone:</td>
    <td><span oraField="homePhone" class="outputData"></span></td>
  </tr>
  <tr>
    <td class="outputLabel">Business Phone:</td>
    <td><span oraField="businessPhone" class="outputData"></span></td>
  </tr>
  <tr>
    <td class="outputLabel">Cell Phone:</td>
    <td><span oraField="cellPhone" class="outputData"></span></td>
  </tr>
  <tr>
    <td class="outputLabel">FAX:</td>
    <td><span oraField="fax" class="outputData"></span></td>
  </tr>
  <tr>
    <td class="outputLabel">Social Security:</td>
    <td><span oraField="socialSecurity" class="outputData"></span></td>
  </tr>
  <tr>
    <td class="outputLabel">Drivers License:</td>
    <td><span oraField="driversLicense" class="outputData"></span></td>
  </tr>
  <tr>
    <td class="outputLabel">Email:</td>
    <td><span oraField="email" class="outputData"></span></td>
  </tr>

</table>

</body>

<xml>
<root>
  <name>Greer, Johan</name>
  <email>jurgen.greer@media.com</email>
  <socialSecurity>939-30-3939</socialSecurity>
  <driversLicense>C8392020</driversLicense>
  <homePhone>(838) 030-0303</homePhone>
  <cellPhone>(444) 444-4040</cellPhone>
  <businessPhone>(737) 393-3838</businessPhone>
  <fax>(373) 939-3939</fax>
  <personId>1239997654</personId>
</root>
</xml>
</html>

```

Figure 6: HTML to Display Customer Business Object

Name:	Greer, Johan	
Home Phone:	(838) 030-0303	
Business Phone:	(737) 393-3838	
Cell Phone:	(444) 444-4040	
FAX:	(373) 939-3939	
Social Security:	939-30-3939	
Drivers License:	C8392020	
Email:	jurgen.greer@media.com	<input type="button" value="edit"/>

Figure 7: Customer HTML Rendered (Output Data for Zone)

UI maps are typically crafted as output tables when used in conjunction with portal zones - please refer to [Map Zones](#) for more information. When referenced within [BPA scripts](#), UI maps are typically crafted as forms for the capture and update of data.

Edit Personal Information

[err](#)

Name:
Last-name suffix, prefix first-name middle-name/initial

Home Phone: -

Business Phone: -

Cell Phone: -

FAX: -

Social Security: - -

Drivers License:

Email:

Figure 8: HTML Input Form Rendered (for BPA Script)

Portal zones can reference a UI map for the zone header. They may also utilize a UI map to define their filter area. This type of UI map is not a complete HTML document, but is instead configured as a UI Map "fragment".

NOTE: UI Map Tips. A context sensitive "UI Map Tips" zone appears when you open the [UI Map](#) page to assist you with the HTML document and schema definition syntax. The "tips" describe methods to take advantage of the configuration tools, however, they are not an HTML reference.

NOTE: Editing HTML. You can use a variety of HTML editors to compose your HTML, which you can then cut, and paste into your UI map. In addition, the zone provides a complete list of the XML schema nodes and attributes available to you when you construct the map's data schema.

Defining UI Maps

The topics in this section describe how to maintain UI Maps.

UI Map - Main

Use this page to define basic information about a user interface (UI) Map. Open this page using **Admin > System > UI Map**

Description of Page

Enter a unique **Map** name. **Owner** indicates if the UI map is owned by the base package or by your implementation (**Customer Modification**).

CAUTION: Important! If you introduce a new UI map, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

Use **UI Map Type** to indicate whether the map is a **Complete HTML Document** or an **HTML Fragment**. Portal zones can reference a UI map to describe a fragment of their HTML, for example the zone header or filter area. In this case the UI map is not a complete HTML document, but is instead configured as a UI Map "fragment".

NOTE: A context sensitive "UI Map Tips" zone is associated with this page to assist you with the HTML document definition syntax. Refer to the tips zone for more information on how to use fragment UI maps to construct portal zone HTML components.

Enter a **Description**. Use the **Detailed Description** to describe how this map is used in detail.

Click on the **View Schema** to view the UI map's expanded schema definition. Doing this opens the [schema viewer](#) window.

Use the **Test UI Map** hyperlink to render your html in a test window.

NOTE: The **Test UI Map** hyperlink also exercises the proprietary functionality that binds an xml element with an html element so you can get immediate feedback on your html syntax.

Where Used

Follow this link to open the data dictionary to view the tables that reference [F1_MAP](#).

UI Map - Schema

Use this page to maintain a UI Map's HTML and schema and to see where the UI Map is used in the system. Open this page using **Admin > System > UI Map** and then navigate to the **Schema** tab.

Description of Page

The contents of this section describe the zones that are available on this portal.

The **General Information** zone displays main attributes of the UI Map.

The **HTML Editor** zone allows you to edit the HTML document of the map.

NOTE: HTML Definition Tips. A context sensitive "UI Map Tips" zone is associated with this page to assist you with the HTML document definition syntax. The "tips" describe good ways to produce simple HTML, however, they are not an HTML reference. Note that you can use a variety of HTML editors to compose your HTML, which you can then cut and paste into your UI map.

NOTE: Providing Help. A [tool tip](#) can be used to display additional help information to the user. This applies to section elements as well as individual elements on a map. Refer to the tips context sensitive zone for more information on how to enable and provide UI map help.

The [Schema Designer](#) zone allows you to edit the data schema part of the map. The purpose of the schema is to describe the data elements being displayed by the map.

NOTE: Schema Definition Tips. The same "UI Map Tips" zone above also provides a complete list of the XML nodes and attributes available to you when you construct the map's data schema.

The **Schema Usage Tree** zone summarizes all cross-references to this schema. These may be other schemas, scripts, and web services. For each type of referencing entity, the [tree](#) displays a summary node showing a total count of referencing items. The summary node appears if at least one referencing item exists. Expand the node to list the referencing items and use their description to navigate to their corresponding pages.

Ensuring Unique Element IDs for UI Maps

The following describes how to modify JavaScript code to ensure the proper rendering of unique element IDs for UI Maps.

The modification is required only for code that renders HTML using a `getElementById()` (or similar) function to generate list IDs and avoid account verification or related errors.

The following sample snippet contains the necessary modifications:

```
...
function getElementsFromList(namePrefix) {
    var ret = [];
    var elements = document.getElementsByTagName("INPUT");
    for(var i=0;i<elements.length;i++) {
        var elemID = elements[i].id;
        if((id) && (id.startsWith(namePrefix + '_'))) {
            ret.push(elements[i]);
        }
    }
    ...
return ret;
}
```

Since IDs aren't necessarily unique in generated UI Map IDs, the code shown above ensures uniqueness at runtime by appending an underscore and row number (e.g., `myField_1`, `myField_2`) for proper handling by Framework in the rendered HTML, while still allowing you to reference the unmodified IDs contained in the generated UI Map.

A switch in the `spl.properties` file also permits you to disable the generation of unique IDs for elements in a grid (as described below), though, for standards compliance reasons, it is highly recommended that this switch be left at its default value.

```
Property Name: spl.runtime.compatibility.uiMapDisableGenerateUniqueHtmlIDs
File Name: spl.properties (under web project in FW)
Default Value: false
Accepted Values: true or false
Description: This property controls the generation of unique IDs for all input elements inside a list. When t
compliant.
```

Maintaining Managed Content

The Managed Content maintenance object is used to store content such as XSL files used to create vector charts, JavaScript include files, and CSS files. These files may then be maintained in the same manner as the HTML in UI Maps.

The topics in this section describe the Managed Content portal.

Managed Content - Main

This page is used to define basic information about the content. Open this page using **Admin > System > Managed Content**.

Description of Page

Enter a unique name for the content in the **Managed Content** field.

Owner indicates if the content is owned by the base package or by your implementation.

Use **Managed Content Type** to indicate the type of content, for example, XSLT or JavaScript.

Enter a **Description**.

Use the **Detailed Description** to describe in detail how this map is used.

Managed Content - Schema

This page is used to create and maintain the managed content. Open this page using **Admin > System > Managed Content** and then navigate to the **Schema** tab.

Description of Page

The General Information zone displays the main attributes of the content. The Editor zone allows you to edit the content.

Data Areas

The data area has no business purpose other than to provide a common schema location for re-used schema structures. It exists solely to help eliminate redundant element declaration. For example, if you have multiple schemas that share a common structure, you can set up a stand-alone data area schema for the common elements and then include it in each of the other schemas.

Be aware that a stand-alone data area can hold elements that are mapped to true fields. For example, you might have 50 different types of field activities and all might share a common set of elements to identify where and when the activity should take place. It would be wise to declare the elements that are common for all in a stand-alone data area and then include it in the 50 field activity business objects.

It's strongly recommended that you take advantage of stand-alone data areas to avoid redundant data definition!

CAUTION: Dynamic inclusion! When the system renders a schema, all schemas included within it are expanded real-time. This means that any change you make to a data area will take effect immediately within all schemas it is referenced within.

NOTE: Schema Definition Tips. A context sensitive "Schema Tips" zone appears when you open the [Data Area](#) page to assist you with the schema definition syntax. The zone provides a complete list of the XML nodes and attributes available to you when you construct a schema.

Data areas may be included in a business object that does not define a full UI map for display or input. Rather, it is using auto-rendering by defining UI attributes in its schema and via UI hints.

NOTE: View UI Rendering. A context sensitive "View UI Rendering" zone appears on this page. It may be used for a data area that is part of a business object that is using auto-rendering for the display and input maps. The buttons allow you to view the rendered UI for the segment of the schema that is defined by the data area.

Defining Data Areas

The topics in this section describe how to maintain Data Areas.

Data Area - Main

Use this page to define basic information about a data area. Open this page using **Admin > System > Data Area**.

Description of Page

Enter a unique **Data Area** name and **Description**. Use the **Detailed Description** to describe what this data area defines in detail. **Owner** indicates if the data area is owned by the base package or by your implementation (**Customer Modification**).

CAUTION: Important! If you introduce a new data area, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

Click the **View Schema** to view the data area's expanded schema definition. Doing this opens the [schema viewer](#) window.

Click the **View XSD** hyperlink to view the business object's expanded schema definition in XSD format.

To extend another data area, reference that data area in the **Extended Data Area** field. By extending a data area you can add additional elements to a base product data area.

Here's an example of an extended data area:

- The product releases with data area A, which contains elements a, b, and c.
- Your implementation creates data area CM-A, which contains element z, and references data area A as the extended data area.
- At run time, everywhere data area A is included it will contain elements a, b, c, and z.

Where Used

Follow this link to open the data dictionary to view the tables that reference [FI_DATA_AREA](#).

Data Area - Schema

Use this page to maintain a Data Area's schema and to see where the data area is used in the system. Open this page using **Admin > System > Data Area** and then navigate to the **Schema** tab.

Description of Page

The contents of this section describe the zones that are available on this portal.

The **General Information** zone displays the main attributes of the data area.

The [Schema Designer](#) zone allows you to edit the data area's schema. The purpose of the schema is to describe the structure and elements of the data area.

NOTE: Schema Definition Tips. A context sensitive "Schema Tips" zone is associated with this page. The zone provides a complete list of the XML nodes and attributes available to you when you construct a schema.

NOTE: View UI Rendering. A context sensitive "View UI Rendering" zone is associated with this page. The zone is useful for data areas that are to be included in [business objects that define the user interface detail](#) using schema attributes and UI Hints. The buttons allow you to view the automatically rendered display and input maps.

The **Schema Usage Tree** zone summarizes all cross-references to this schema. These may be other schemas, scripts, and web services. For each type of referencing entity, the [tree](#) displays a summary node showing a total count of referencing items. The summary node appears if at least one referencing item exists. Expand the node to list the referencing items and use their description to navigate to their corresponding pages.

Schema Designer

The Schema Designer is a user-friendly interface for performing the following common schema editing tasks:

- Displaying existing schemas.
- Creating schema elements.
- Moving elements within a schema.
- Adding attribute values.

The designer provides the ability to toggle between **Design View** and **Source View** views by clicking the icons on the **Schema Designer** title bar:



The **Source View** shows the schema elements and their attributes written in the proper syntax.


In the graphical **Design View** mode, the zone is split into two areas. The left pane displays the elements of the schema in a tree-like presentation, while the right pane displays all valid attributes and values for selected schema elements.

The following points highlight some functionality available in the design view.

- If the schema definition refers to another schema using an “include” statement, one can expand that schema in the left panel to view the elements within that schema. Clicking an element in that expanded view shows information about the element’s definition in the right panel. Changes cannot be made to elements in this view.
- To move an existing element within the schema in the left panel, simply drag and drop.
- Right clicking an element in the left panel causes a pop-up to display which offers options based on the position of the element clicked on and based on the type of element.
 - **Add element above | below**
 - **Delete this element**
 - **Move this element up | down**
 - **Add child element**

When adding a new element, you are prompted for the element type. The following lists the possible element types. Most are self explanatory and represent standard schema options.

- Comment. This adds a comment to the schema.
- Embedded HTML. This is specific to a schema enabled for UI Hints. It is used to include a UI map fragment.
- Field
- Flattened Field
- Flattened List
- Group
- Include BO Schema
- Include BS Schema
- Include DA Schema
- Include Map Schema
- Include SS Schema
- Input Map Title. This is specific to a schema enabled for UI Hints. It is used to define a title element for the map.
- List
- Nested Flattened Field. This is a flattened field from a child table.
- Raw Element. This element is used to capture text as is. It is typically used to capture an XML structure without any details of the definition of the individual nodes.
- Section. This is specific to a schema enabled for UI Hints. It is used to define a section within the map.
- Simple Field. This is a special type of Field element that is used to define an element that is mapped to an XML column.

Context-sensitive embedded help is provided for fields and controls in the right pane by clicking the Help icon 

You can also use links in the **Schema Tips** zone in the dashboard area to open a navigable Help window containing schema-related topics and tips. To search topic content, press `Ctrl+F` when the Schema Tips Help window has focus.

The **Schema Designer** is available by choosing the **Schema** tab on the [Business Object](#), [Data Area](#), [UI Map](#), [Business Service](#), and [Script](#) pages.

Schema Viewer

The schema viewer shows a tree-view presentation of a schema in its expanded form.

The schema illustrates the structure to be used when communicating with the schema's associated object. The following takes place when a schema is expanded:

- If the schema definition includes references to other schemas, these references are replaced with the corresponding schema definitions.
- Also, if the schema definition contains **private** elements, they are omitted from this view.

Clicking on any node on the tree populates the text box on the top with the node's absolute XPath expression. You will find this feature very useful when writing scripts interacting with schema-based objects. [Scripting](#) often involves referencing elements in a schema-based XML document using their absolute XPath expression. You can use this feature on the schema viewer to obtain the XPath expression for an element and copy it over to your script.

Business Event Log

Business Event Log may be viewed as a tool designed to capture any type of business event worth noting. You configure business objects to represent the various types of events your application calls for. The following type of details may be captured for each event:

- The business object representing the type of event.
- The date and time the event took place and who initiated it.
- The business entity for which this event is logged.
- Standard application message to describe the event.
- Additional context information that is available at the time of the event and varies for each type of event. The Business Event Log maintenance object supports a standard characteristics collection as well as an XML storage (CLOB) field. The event's business object determines where each piece of information resides. Refer to [Business Objects](#) for more information.

One common type of event may be the audit of changes made to sensitive data, for example, tracking an address change. Whenever an entity associated with a business object is added, changed, or deleted the system summarizes the list of changes that took place in that transaction and hands them over to **Audit** business object algorithms to process. You may design such an algorithm to audit the changes as business event logs. Refer to [a business object may define business rules](#) for more information.

You can also allow users to initiate business event logs to capture important notes about a business entity by exposing a [BPA Script](#) to invoke the event's corresponding business object.

Bottom line is that any process can create a business event log by invoking the business object representing the appropriate type of event.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [FI_BUS_EVT_LOG](#).

To Do Lists

Certain events that occur within the system will trigger messages describing work that requires attention. For example, if a bill segment has an error, the system generates a To Do message to alert the person responsible for correcting such errors.

Each type of message represents a To Do list. For example, there are To Do lists for bill segment errors, payment errors, customer contact reminder, etc.

We refer to each message as a **To Do Entry**. Each To Do entry is assigned a specific **To Do Role**. The role defines the users who may work on the entry. A To Do entry has a **To Do log** that maintains record of the progress on the To Do entry. For example, the To Do log indicates when the To Do entry was created, when it was assigned to a user and to whom it was assigned, and when and by whom it was completed.

FASTPATH: Refer to [To Do Processing](#) for a description of end-user queries and tools assisting in reviewing, assigning and processing To Do entries.

The Big Picture of To Do Lists

The topics below provide more information about To Do configuration.

To Do Entries Reference A To Do Type

Every [To Do entry](#) references a To Do type. The To Do type controls the following functions:

- The To Do list on which the entry appears.
- The page into which a user is taken when they drill down on an entry.
- The message that appears in the user's To Do list. Note this message can be overridden for specific To Do messages by specifying a different message number in the process that creates the specific To Do entry. For example, the process that creates To Do entries associated with bill segments that are in error displays the error message rather than a generic "bill segment is in error" message.
- The To Do list's sort options. Sort options may be used on the To Do list page to sort the entries in a different order. For example, when a user looks at the bill segment error To Do list, they have the option of sorting it in error number order, account name order, or in customer class order. Note the default sort order is also defined on To Do type.
- Whether (and how) the To Do entry is downloaded to an external system (e.g., an email system).
- The roles to which an entry may be reassigned.
- The default priority of the To Do list in respect of other To Do lists.
- The To Do list's usage, which indicates whether a To Do of that type may be created manually by a user.
- The algorithms used to perform To Do list specific business rules.
- The characteristics applicable to the To Do list.

To Do Entries Reference A Role

Every [To Do entry](#) references a role. The role defines the users who may be assigned to **Open** entries.

The permissible roles that may be assigned to a To Do entry are defined on the entry's To Do type. After an entry is created, its role may be changed to any role defined as valid for the entry's To Do type.

An entry's initial role is assigned by the background process or algorithm that creates the entry. Because you can create your own processes and algorithms, there are an infinite number of ways to default an entry's role. However, the base package processes and algorithms use the following mechanisms to default an entry's role:

- The system checks if an entry's message category / number is suppressed (i.e., not created). If so, the entry is not created. Refer to [To Do Entries Can Be Rerouted Or Suppressed Based On Message Number](#) for more information.
- The system checks if an entry's message category / number is rerouted to a specific role. If so, it defaults this role. Refer to [To Do Entries Can Be Rerouted Or Suppressed Based On Message Number](#) for more information.
- Your specific product may introduce additional criteria for assigning a role, for example perhaps important accounts are assigned to a kind of account management group and the account management group includes configuration for special role for certain To Do types. Refer to [Set Additional Information Before a To Do is Created](#) for more information.
- If a Role wasn't determined in one of the previous steps and a Role is provided by the initiating process, the entry is created with that Role.
- If the entry does not have a role after the above takes place, the entry's To Do type's default role is assigned to the entry.

NOTE:

At installation time, the system provides a default role assigned to the system To Do types when first installed called **F1_DFLT**. This is done to allow testing of the system prior to implementing of appropriate To Do roles for your organization. The recommendation is to configure all the To Do Types with appropriate business oriented To Do roles once they are defined.

CAUTION: Important! Most organizations have the notion of a supervisor who is responsible for all entries assigned to a given role. It's important for this user (or users) to be part of all such roles. Refer to [To Do Supervisor Functions](#) for information about transactions that can be used by supervisors to review and assign work to users.

To Do Entries Can Be Rerouted (Or Suppressed) Based On Message Number

Consider the To Do type used to highlight bill segments that are in error. To Do entries of this type reference the specific bill segment error number so that the error message can be shown when the Bill Segments in Error To Do list is displayed.

NOTE: Message Category / Message Number. Every error in the system has a unique message category / number combination. Refer to [The Big Picture of System Messages](#) for more information about message categories and numbers.

If you want specific types of errors to be routed to specific users, you can indicate such on the To Do type. For example, if certain bill segment errors are always resolved by a given rate specialist, you can indicate such on the To Do type. You do this by updating the [To Do type's message overrides](#). On this page you specify the message category / number of the error and indicate the To Do role of the user(s) who should work on such errors. Once the To Do type is updated, all new To Do entries of this type that reference the message number are routed to the desired role.

NOTE: Reroute versus suppression. Rather than reroute an entry to a specific role, you can indicate that an entry with a given message number should be suppressed (i.e., not created). You might want to do this if you have a large volume of certain types of errors and you don't want these to clutter your users' To Do lists.

Obviously, you would only reroute those To Do types that handle many different types of messages. In other words, if the To Do type already references a specific message category / number rerouting is not applicable.

We do not supply documentation of every possible message that can be handled by a given To Do type. The best way to build each To Do type's reroute list is during the pre-production period when you're testing the system. During this period, compile a list of the messages that should be routed to specific roles and add them to the To Do type.

Keep in mind that if a message number / category is not referenced on a To Do type's reroute information, the entry is routed as described under [To Do Entries Reference A Role](#).

NOTE: Manually created To Do entries cannot be rerouted or suppressed. The rerouting occurs as part of the batch process or algorithm processing when the To Do is created. The role or user to whom a manual To Do should be assigned is specified when the To Do is created online. A manually created To Do may also be forwarded to another user or role.

The Priority Of A To Do Entry

Some To Do entries may be more urgent to resolve than others. A To Do entry is associated with a priority level representing its relative processing order compared to other entries.

Priority level is initially assigned as follows:

- If a **Calculate Priority** plug-in is defined on the To Do entry's type, the system calls it to determine the entry's priority. You may want to use this method if an entry's priority is based on context or time-based factors. For example, when priority takes into consideration account specific attributes. When applicable, you may design a process that triggers priority recalculation of To Do entries "at will". For example, when priority is reassessed periodically based on factors external to the To Do entry's information. Refer to [To Do Type](#) for more information on priority calculation algorithms.
- If a priority value has not been determined by a plug-in, the system defaults a To Do entry's initial priority to the value specified on its type.

A user may manually override a To Do entry's priority at any time. Notice that once a To Do entry's priority is overridden, **Calculate Priority** plug-ins are no longer called so as to not override the value explicitly set by the user.

NOTE: The system does not use priority values to control order of assignment nor processing of To Do entries. Priority is available to assist your organization with supporting a business practice that ensures higher priority issues are worked on first.

Working On A To Do Entry

A user can drill down on a To Do entry. When a user drills down on an entry, the user is transferred to the transaction associated with the entry. For example, if a user drills down on a bill segment error entry, the user is taken to the Bill Segment - Main page. Obviously, the page to which the user is taken differs depending on the type of entry.

It is also possible to configure the To Do type to launch a [script](#) when a user drills down on an entry rather than taking the user to a transaction. The script would walk the user through the steps required to resolve the To Do entry. Refer to [Launching Scripts When A To Do Is Selected](#) for more information.

After finishing work on an entry, the user can mark it as **Complete**. Completed entries do not appear on the To Do list queries (but they are retained on the database for audit purposes). If the user cannot resolve the problem, the user can forward the To Do to another user.

Launching Scripts When A To Do Is Selected

Users can complete many To Do entries without assistance. However, you can set up the system to launch a [script](#) when a user selects a To Do entry. For example, consider a To Do entry that highlights a bill that's in error due to an invalid mailing address. You can set up the system to execute a script when this To Do entry is selected by a user. This script might prompt the user to first correct the customer's default mailing address and then re-complete the bill.

A script is linked to a To Do type based on its message number using the [To Do type's message overrides](#). Refer to [Executing A Script When A To Do Is Selected](#) for more information.

To Do Entries Have Logs

Each [To Do entry](#) has a To Do log that maintains a record of the To Do's progress in the system. For example, the To Do log indicates when the To Do entry was created, when it was assigned to a user and to whom it was assigned, and when and by whom it was completed. Users can view the log to see who assigned them a particular To Do and whether any work has already been done on the To Do.

A log entry is created for all actions that can be taken on a To Do entry. Log entries are created for the following events:

- A To Do entry is created (either by the system or by a user)
- A To Do entry is completed (either by the system or by a user)
- A user takes an open To Do entry
- A supervisor assigns a To Do entry
- A user forwards an entry to another user or role
- A user sends back a To Do to the user who forwarded it
- A user manually adds a log entry to record details about the To Do's progress
- A user manually overrides the To Do entry's priority

FASTPATH: For information about the contents of log entries for each of the events, refer to [Log Entry Events](#).

How Are To Do Entries Created?

A To Do Entry may be created in the following ways:

- A [background process](#) can create To Do Entries.
- An [algorithm](#) can create entries of a given type. Because the use of algorithms is entirely dependent on how you configure the control tables, the number of types of such entries is indeterminate.
- A user can create entries of To Do types that have a **Manual** usage. Refer to [To Do Entries Created Manually](#) for information about setting up manual To Do types.

For any base product process that includes logic to create a To Do entry, the system supplies a sample To Do type that may be used. Although the To Do types provided by the product are system data, the following information related to each To Do type may be customized for an implementation and is not overwritten during an upgrade:

- The creation process. If the To Do is created by a background process where the background process is referenced on a To Do type. Refer to [To Do Entries Created By Background Processes](#) for more information.
- The routing process. Refer to [To Do Entries May Be Routed Out of the System](#) for more information.
- The priority. Refer to [To Do Type - Main](#) for more information.
- The [roles](#) that may be associated with the To Do type. Refer to [To Do Entries Reference a Role](#) for more information.
- The [message override](#) information. Refer to [To Do Entries Can Be Rerouted \(Or Suppressed\)](#) and [Launching Scripts When a To Do Is Selected](#) for more information.

To Do Entries Created By Background Processes

There are different types of To Do entries created by background processes:

- To Do entries created by dedicated To Do background processes
- To Do entries created for object-specific errors detected in certain background processes

- To Do entries created based on a specific condition

Dedicated To Do Background Processes

There are To Do entries that are created by system background processes whose main purpose is to create To Do entries based on a given condition. For these background processes, the To Do Type indicates the creation background process.

NOTE: If you don't schedule the background process, the entries will NOT be created! The To Do entries of this type will only be created if you have scheduled the associated background process. Therefore, if you want the system to produce a given entry, schedule the background process.

To Dos Created for Object-Specific Error Conditions

A system background process may create a To Do entry when an error is detected during object-specific processing and it is not possible to create an exception record. (I.e., either no exception record exists for the process or the error is not related to the entity reported in the exception record.)

For these background processes, the To Do Type must reference the creation background process. To have the system create To Do entries for some or all of the errors generated by one of these processes, you must do the following:

- If you want the system to generate To Do entries for errors detected by one of the background processes below, go to the appropriate To Do type and populate the creation background process.
- If you want the system to generate To Do entries for some errors for the process, but not for all errors, populate the creation background process and then proceed to the [message overrides](#) tab to [suppress](#) certain messages. To this by indicating the message category and message number you want to suppress. Any error that is suppressed is written to the [batch run tree](#).

If you do not populate the creation background process, the errors are written to the [batch run tree](#).

NOTE: Errors received while creating a To Do entry. If the background process cannot successfully create a To Do entry to report an object-specific error, the error is written to the batch run tree along with the error encountered while attempting to create the To Do entry.

NOTE: System errors are not included. To Do entries are not created for a system error, for example an error related to validation of input parameter. These errors are written to the [batch run tree](#). Refer to [Processing Errors](#) for more information.

To Dos Created by Background Processes for Specific Conditions

There are some system background processes that create a To Do entry when the process detects a specific condition that a user should investigate. For each background process, the To Do type is an input parameter to the process. The system provides To Do types for each base package background process that may create a To Do entry.

NOTE: No Creation Process. These To Do types do not need (and should not have) a **Creation Process** specified.

To Do Entries Created By Algorithms

There are To Do entries that are created by algorithm types supplied with the base package. The system supplies a To Do Type for each of these To Do entries that you may use.

If you want to take advantage of these types of entries for system algorithm types, you must do the following:

- Create an [algorithm](#):

- This algorithm must reference the appropriate Algorithm Type.
- These algorithms have a parameter of the To Do Type to be created. You should specify the To Do Type indicated in the table.
- Plug the algorithm into the respective control table.

To Do Entries Created Manually

You must set up manual To Do entry types if you want your users to be able to create To Do entries online. Users may create a manual To Do entry as a reminder to themselves to complete a task. Online To Do entries may also be used like electronic help tickets in the system. For example, if a user is having a problem starting service, the user can create a To Do that describes the problem. The To Do can be assigned to a help resolution group that could either resolve the problem or send the To Do back to the initiating user with information describing how to resolve the problem.

If you want to take advantage of manual To Do entries, create a To Do type and specify the following information.

On the Main tab:

- Set the **To Do Type Usage** flag to **Manual**.
- Set the **Navigation Option** to **toDoEntryMaint** (To Do entry maintenance).
- Set the **Message Category** and **Message Number** to the message you want to be used for To Do entries of this type. The system will populate the message parameter with the Subject. To show only the subject in the To Do's message, use a message with "%1" as its text.

On the Roles tab:

- Specify the **To Do roles** that may be assigned to To Do entries of this type.
- Indicate the To Do role that should be defaulted when you create To Do entries of this type.

On the Sort Keys tab:

When a user adds a manual To Do entry, the system creates an entry with three sort key values. (Sort keys may be used on the To Do list page to sort the entries in a different order.) The To Do type should be set up to reference the sort keys as follows:

Sequence	Description
1	Created by user ID
2	Created by user name
3	Subject

We recommend that the keys have an **Ascending** sort order and that the Subject is made the default sort key.

NOTE: It is possible to define additional sort keys and use a To Do Post Processing algorithm to populate the values. In this case, the base sort keys defined above should still be defined.

On the Drill Keys tab:

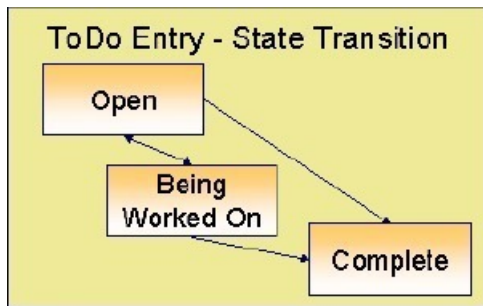
When a user adds a manual To Do entry, it is created with a drill key value equal to the To Do entry's ID. When the user clicks the Go To button next to the message in the To Do list, the system uses the drill down application service (defined on the main tab) and the drill key to display the associated To Do entry.

The To Do type must be set up with a drill key that reference the To Do entry table and the To Do entry ID:

Sequence	Table	Field
1	CI_TD_ENTRY	TD_ENTRY_ID

The Lifecycle Of A To Do Entry

The following state transition diagram will be useful in understanding the lifecycle of a To Do entry.



A To Do entry is typically created in the **Open** state. Entries of this type can be viewed by all users belonging to the entry's role. Refer to [How Are To Do Entries Created?](#) for information about how entries are created.

An **Open** entry becomes **Being Worked On** when it is assigned to a specific user or when a user proactively assumes responsibility for the entry. While an entry is **Being Worked On**, it is visible on the To Do Summary page only by the user who is assigned to it.

NOTE: To Do entries may be created in the **Being Worked On** state. Some To Do background processes may create To Do entries in the **Being Worked On** state. When a user adds a To Do entry online and assigns the To Do to a user (as opposed to a role), the To Do entry is also created in the **Being Worked On** state.

A **Being Worked On** entry may be forwarded to a different user or role. If the entry is forwarded to a role, it becomes **Open** again.

When an entry becomes **Complete**, it is no longer visible in the To Do list queries (but it remains on the database for audit purposes). There are two ways an entry can become **Complete**:

- A user can manually indicate it is **Complete** (there are several ways to do this).
- For To Do entries that are logically associated with the state of some object, the system automatically marks the entry **Complete** when the object is no longer in the respective state. For example, an entry that's created when an account doesn't have a bill cycle is completed when the account has a bill cycle.

CAUTION: Important! The automatic completion of To Do entries occurs when the background process responsible for creating entries of a given type is executed. Therefore, if you only run these processes once per day, these entries remain **Being Worked On** even if the object is no longer in the respective state.

Linking Additional Information To A To Do Entry

Additional information may be linked to a To Do entry using characteristics. For example, when creating a manual To Do entry, a user may define the account related to the To Do.

When creating an automatic To Do entry, the program that generates the To Do may link related data to the To Do using characteristics. Use system [algorithm](#) to link related entities. For manually created To Dos, the valid characteristic types that may be linked to the To Do entry must be defined on the [To Do type](#) for that To Do entry.

If your To Do entries reference characteristics that are related to your global context data, you may want to configure an [Alerts](#) to display an alert if a related entry is **Open** or **Being Worked On**.

Implementing Additional To Do Entry Business Rules

If your business practice calls for additional validation rules or processing steps to take place after a To Do Entry is created or updated, you may want to take advantage of the **To Do Post Processing** plug-ins defined on [To Do type](#).

For example, you may want to validate that To Do entries are only assigned to users with the proper skill levels needed to resolve them. Refer to [F1-VAL-SKILL](#) for a sample algorithm handling such validation.

To Do Entries May Be Routed Out Of The System

A To Do type can be configured so that its entries are interfaced to another system.

For example, a given To Do type can be configured to create an email message whenever a new To Do entry is created. The following points describe how to do this:

- Define the name of the background process responsible for interfacing the new To Do entries to another system on the respective To Do type. The base package contains a batch process called [F1-TDEER](#) that can be used for most situations. This batch process invokes the **External Routing algorithms** defined on each entry's To Do type.
- Plug in an appropriate **External Routing** algorithm on the respective To Do type. The logic in this type of algorithm performs the interface efforts for a specific To Do entry. For example, if an email message should be created for a To Do entry, the logic in the algorithm would compose and send the email message(s) for a specific To Do entry.

Click [here](#) to see the algorithm types available for this system event.

Periodically Purging To Do Entries

Completed To Do entries should be periodically purged from the system by executing the [F1-TDPG](#) background process. This background process offers you the following choices:

- You can purge all To Do entries older than a given number of days.
- You can purge To Do entries for a specific list of To Do types that are older than a given number of days.
- You can purge all To Do entries except for a specific list of To Do types that are older than a given number of days.

We want to stress that there is no system constraint as to the number of **Completed** To Do entries that may exist. You can retain these entries for as long as you desire. However, you will eventually end up with a very large number of **Completed** entries and these entries will cause the various To Do background processes to degrade over time. Therefore, you should periodically purge **Completed** To Do entries as they exist only to satisfy auditing and reporting needs.

NOTE: Different retention periods for different types of To Do entries. Keep in mind that the purge program allows you to retain different types of entries for different periods of time.

Setting Up To Do Options

The topics in this section describe how to set up To Do management options.

Installation Options

The following section describes configuration setup on the installation options.

To Do Information May Be Formatted By An Algorithm

A **To Do Information** algorithm may be plugged in on the [installation record](#) to format the standard To Do information that appears throughout the system. This algorithm may be further overridden by a corresponding plug-in on the [To Do Type](#).

Set Additional Information Before A To Do Is Created

A **To Do Pre-creation** algorithm may be plugged in on the [installation record](#) to set additional information for a To Do entry before it is created. Algorithms of this type are used for two common purposes:

- Linking context specific data to the To Do entry using characteristics. For example, Oracle Utilities Customer Care and Billing provides an algorithm that attempts to link a related person, account, premise, service agreement or service point to a To Do entry based on its drill key value. Note, before you can set up this algorithm, you must define the characteristic types that you'll use to hold each of these entities. Also note that it is not necessary to define these characteristics as valid characteristic types on the To Do type.
- Overriding the Role of a To Do entry based on specific configuration related to the To Do's context data. For example, Oracle Utilities Customer Care and Billing provides an algorithm to determine a To Do role based on overrides related to the account's account management group or division.

Alerts

If your To Do entries reference characteristics related to your global context data and your product supports dashboard alerts generated by algorithms, you may want configure an algorithm to display an alert if the entry is **Open** or **Being Worked On** for the data currently in context.

Refer to your product's documentation to determine if these types of alerts are supported.

Next Assignment Algorithm

If your organization opts to use the next assignment feature supported by the Current To Do dashboard zone, you need to plug-in a **Next To Do Assignment** algorithm into the [installation options](#) to determine the next To Do entry the user should work on. Make sure you provide users with security access rights to the zone's next assignment action.

FASTPATH: Refer to the [Current To Do](#) zone for more information.

Messages

You need only set up new messages if you use algorithms to create To Do entries or prefer different messages than those associated with the base package's To Do types.

Feature Configuration

The base package is provided with a generic **Activity Queue Management Feature Configuration** type. You may want to set up a feature configuration of this type to define any To Do management related options supporting business rules specific to your organization.

For example, the base package provides the following plug-ins to demonstrate a business practice where To Do entries are only assigned to users with the proper skill levels to work on them.

- The base **To Do Post Processing** To Do Type algorithm [F1-VAL-SKILL](#) validates that a user has the necessary skill levels required to work on a given To Do entry.
- The base **Next To Do Assignment** installation options algorithm [F1-NEXT-ASSG](#) only assigns To Do entries to users that have the proper skills to work on them. This plug-in is only applicable if your organization practices [work distribution](#) "on demand."

You must set up such an **Activity Queue Management** feature configuration if you want to use any of the above base package plug-ins.

The following points describe the various **Option Types** provided with the base package:

- **Skill.** This option provides a reference to a skill category. For example, if you were using characteristics to represent skill categories then you should reference each characteristic type using this option.
- **Override Skill.** This option provides an override skill information reference for a specific message. For example, if you were using a To Do Type characteristic to specify an override skill category and level for a specific message category / number then you would reference this characteristic type using this option.

NOTE: Skill Setup. Refer to the description of the above base package algorithms for further details on how to setup skill level information.

NOTE: More Options. Your implementation may define additional options types. You do this by add new lookup values to the lookup field **F1QM_OPT_TYP_FLG**.

NOTE: Only one. The system expects only one **Activity Queue Management** feature configuration to be defined.

Defining To Do Roles

This section describes the control table used to maintain To Do roles.

To Do Role - Main

The **Main** page is used to define basic information about a To Do role.

To maintain this information, select **Admin > General > To Do Role**.

Description of Page

Enter a unique **To Do Role** and **Description** for the To Do role.

The grid contains the ID of each **User** that may view and work on entries assigned to this role. The First Name and Last Name associated with the user is displayed adjacent.

NOTE: System Default Role. The system supplies a default role **F1_DFLT** linked to each system To Do type. This is done so that To Do functionality may be tested prior to the creation of appropriate business oriented To Do roles.

Where Used

Follow this link to view the tables that reference [CI_ROLE](#) in the data dictionary schema viewer.

In addition, various "type" objects or algorithms may reference a To Do role to use when creating a To Do for a given business scenario. This is dependent on your specific product.

To Do Role - To Do Types

The **To Do Types** page defines the To Do types that may be viewed and worked on by users belonging to a given To Do role.

To maintain this information, select **Admin > To Do Role > Search** and navigate to the **To Do Types** page.

Description of Page

Enter the ID of each **To Do Type** whose entries may be viewed and worked on by the role.

Use As Default is a display-only field that indicates if the role is assigned to newly created entries of this type. You may define the default role for a given To Do type on the To Do Type maintenance page.

CAUTION: If you remove a To Do type where this role is the default, you must define a new role as the default for the To Do type. You do this on the To Do Type maintenance page.

Defining To Do Types

This section describes the control table used to maintain To Do types.

To Do Type - Main

The **Main** page is used to define basic information about a To Do type.

FASTPATH: Refer to [The Big Picture Of To Do Lists](#) for more information about To Do types and To Do lists in general.

To maintain this information, select **Admin > General > To Do Type**.

CAUTION: Important! If you introduce a To Do type, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

Description of Page

Enter a unique **To Do Type** and **Description** for the To Do type.

Owner indicates if this entry is owned by the base package or by your implementation (**Customer Modification**).

Use the **Detailed Description** to provide further details related to the To Do Type.

Enter the default **Priority** of To Do entries of this type in respect of other To Do types. Refer to [The Priority Of A To Do Entry](#) for more information.

For **To Do Type Usage**, select **Automatic** if To Dos of this type are created by the system (i.e., a background process or algorithm). Select **Manual** if a user can create a To Do of this type online.

Define the **Navigation Option** for the page into which the user is transferred when drilling down on a To Do entry of this type.

Use **Creation Process** to define the background process, if any, that is used to manage (i.e., create and perhaps complete) entries of this type. A **Creation Process** need only be specified for those To Do types whose entries are created by a background process. Refer to [To Do Entries Created By Background Processes](#) for more information.

Use **Routing Process** to define the background process that is used to download entries of a given type to an external system, if any. A **Routing Process** need only be specified for those To Do types whose entries are routed to an external system (e.g., an Email system or an auto-dialer). Refer to [To Do Entries May Be Routed Out Of The System](#) for more information.

Use **Message Category** and **Message Number** to define the message associated with this To Do type's entries. Note: this message will only be used if the process that creates the To Do entry does not supply a specific message number. For example, the process that creates To Do entries that highlight bill segments that are in error would not use this message; rather, the entries are marked with the message associated with the bill segment's error.

Use the characteristics collection to define a **Characteristic Type** and **Characteristic Value** common to all To Do entries of this type. You may enter more than one characteristic row for the same characteristic type, each associated with a unique **Sequence** number. If not specified, the system defaults it to the next sequence number for the characteristic type.

Where Used

Follow this link to view the tables that reference [CI_TD_TYPE](#) in the data dictionary schema viewer.

To Do Type - Roles

The **Roles** page defines the roles who may view and work on entries of a given To Do type.

To maintain this information, select **Admin > To Do Type > Search** and navigate to the **Roles** page.

Description of Page

Enter each **To Do Role** that may view and work on entries of a given type. Turn on **Use as Default** if the role should be assigned to newly created entries of this type. Only one role may be defined as the default per To Do type.

FASTPATH: Refer to [To Do Entries Reference A Role](#) for more information about roles and To Do entries.

To Do Type - Sort Keys

The **Sort Keys** page defines the various ways a To Do list's entries may be sorted. For example, when you look at the bill segment error To Do List, you have the option of sorting the entries in error number order, account name order, or in customer class order.

To maintain this information, select **Admin > To Do Type > Search** and navigate to the **Sort Keys** page.

CAUTION: Do not change this information unless you are positive that the process / algorithm that creates entries of a given type stores this information on the entries.

Description of Page

The following fields display for each sort key.

Sequence is the unique ID of the sort key.

Description is the description of the sort key that appears on the To Do list.

Use as Default indicates the default sort key (the one that is initially used when a user opens a To Do list). Only one sort key may be defined as the default per To Do type.

Sort Order indicates whether the To Do entries should be sorted in **Ascending** or **Descending** order when this sort key is used.

Owner indicates if this entry is owned by the base package or by your implementation (**Customer Modification**).

To Do Type - Drill Keys

The **Drill Keys** page defines the keys passed to the application service (defined on the Main page) when you drill down on an entry of a given type.

To maintain this information, select **Admin > To Do Type > Search** and navigate to the **Drill Keys** page.

CAUTION: Do not change this information unless you are positive that the process / algorithm that creates entries of a given type stores this information on the entries.

Description of Page

Navigation Option shows the page into which the user is transferred when drilling down on a To Do entry of this type.

The following fields display for each drill key.

Sequence is the unique ID of the drill key.

Table and **Field** are passed to the application service when you drill down on an entry of a given type.

Owner indicates if this entry is owned by the base package or by your implementation (**Customer Modification**).

To Do Type - Message Overrides

The **Message Overrides** page is used if you want To Do entries that reference a given message category / number to be routed to a specific To Do role (or suppressed altogether) or if you want to associate a script to a given message category / number.

FASTPATH: Refer to [To Do Entries Reference A Role](#) and [To Do Entries Can Be Rerouted Or Suppressed](#) for more information.

To maintain this information, select **Admin > To Do Type > Search** and navigate to the **Message Overrides** page.

Description of Page

The following fields display for each override.

Message Category and **Number** allow the message to be overridden.

Exclude To Do Entry indicates if a To Do entry of this type that references the adjacent **Message Category** and **Number** should NOT be created.

Override Role indicates the to do role to which a To Do entry of this type that references the adjacent **Message Category** and **Number** should be addressed. This field is protected if **Exclude To Do Entry** is on.

Script indicates the script that should execute when a user drills down on a To Do entry of this type that references the adjacent **Message Category** and **Number**. This field is protected if **Exclude To Do Entry** is on. Refer to [Working On A To Do Entry](#) for more information.

To Do Type - To Do Characteristics

The **To Do Characteristics** page defines characteristics that can be defined for To Do entries of this type. The characteristic types for characteristics that are linked to the To Do entry as a result of a pre-creation algorithm do not need to be defined here.

To maintain this information, select **Admin > To Do Type > Search** and navigate to the **To Do Characteristics** page.

Turn on the **Required** switch if the **Characteristic Type** must be defined on To Do entries of this type.

Enter a **Characteristic Value** to use as the default for a given **Characteristic Type** when the **Default** switch is turned on. Use **Sequence** to control the order in which characteristics are defaulted.

To Do Type - Algorithms

The **To Do Algorithms** page defines the algorithms that should be executed for a given To Do type.

To maintain this information, select **Admin > To Do Type > Search** and navigate to the **Algorithms** page.

Description of Page

The grid contains **Algorithms** that control important To Do functions. If you haven't already done so, you must [set up the appropriate algorithms](#) in your system. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).
- Specify the **Sequence Number** and **Algorithm** for each system event. You can set the **Sequence Number** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

The following table describes each **System Event**.

System Event	Optional / Required	Description
Calculate Priority	Optional	<p>Algorithms of this type may be used to calculate a To Do entry's priority. They are called initially when a To Do entry is created and each time it gets updated so long as the To Do entry's priority has not been manually overridden. Once overridden, these algorithms are not called anymore.</p> <p>Note that it is not the responsibility of the algorithms to actually update the To Do entry with the calculated priority value but rather only return the calculated value. The system carries out the update as necessary.</p> <p>If more than one algorithm is plugged-in the system calls them one by one until the first to return a calculated priority.</p> <p>Click here to see the algorithm types available for this system event.</p>
External Routing	Optional	<p>Algorithms of this type may be used to route a To Do entry to an external system.</p> <p>The base package F1-TDEER background process invokes the algorithms for every To Do entry that its type references the process as the Routing Process and that the entry was not already routed. The background process marks an entry as routed by updating it with the batch control's current run number.</p> <p>If more than one algorithm is plugged-in the batch process calls them one by one until the first to indicate the To Do entry was routed.</p> <p>Click here to see the algorithm types available for this system event.</p>
To Do Information	Optional	<p>We use the term "To Do information" to describe the basic information that appears throughout the system to describe a To Do entry. The data that appears in "To Do information" is constructed using this algorithm.</p> <p>Plug an algorithm into this spot to override the "To Do information" algorithm on installation options or the system default "To Do information" if no such algorithm is defined on installation options.</p> <p>Click here to see the algorithm types available for this system event.</p>
To Do Post-Processing	Optional	<p>Algorithms of this type may be used to validate and/or further process a To Do entry that has been added or updated.</p> <p>Click here to see the algorithm types available for this system event.</p>

List of System To Do Types

The To Do types available to use with the product are found in the To Do Type page. In addition, they may be viewed in the [application viewer's To Do type](#) viewer. If your implementation adds To Do types, you may [regenerate](#) the application viewer to see your additions reflected there.

Implementing The To Do Entries

To enable the To Do entries visible in the To Do Type page and application viewer, you must configure the system as follows:

- Define the To Do roles associated with each To Do type and link the appropriate users to them. Once you have defined the roles appropriate for your organization's To Do types, remove the reference to this system default role **F1_DFLT**. Refer to [To Do Entries Reference A Role](#) for more information.
- For any To Do Type that is provided for a specific background process, the To Do simply needs to reference the appropriate Creation Background Process. When the background process is scheduled, To Dos are created based on the logic of the related background process. This applies to [To Dos Created for Object-Specific Error Conditions](#) and [Dedicated To Do Background Processes](#).
- For any To Do Type that is provided for creation by an algorithm or other process, there may be configuration required to populate that To Do type as an algorithm parameter or as an attribute on a control table.

NOTE: Refer to the description of the To Do type for more information.

Background Processes

This chapter covers various topics related to background processes. Besides providing an overview of background process functionality, the various tools available within the application to define, submit and monitor background processes are covered.

NOTE: Your specific source application may have additional background process topics. Please refer to the documentation section that applies to your source application for more information.

The Big Picture of Background Processes

This section describes various topics related to the background processes that perform many important functions throughout your product such as:

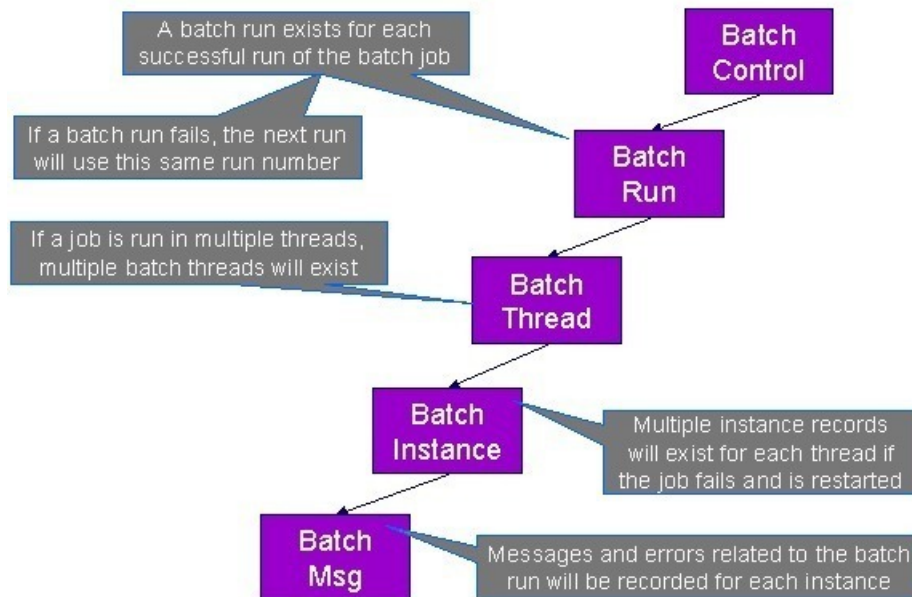
- Processing To Do Entries
- Monitor processes that select records in a given state to progress them to their next state in their lifecycle
- Processes that purge data
- Processes that extract data
- And many more...

Background Processing Concepts

While the system relies on a scheduler to secure and execute its background processes, there are additional issues that you should be familiar with:

- Batch control records are used for the following purposes:
 - Define the code that executes the logic associated with the background process.
 - For processes that extract information, the batch control record defines the next batch number to be assigned to new records that are eligible for extraction. For example, the batch control record associated with the process that routes To Do entries to an external system defines the next batch number to be assigned to new To Do entries that are configured with this batch control. When this To Do external routing process next runs, it selects all To Do entries marked with the current batch number (and increments the next batch number).
 - The batch control record for each background process organizes audit information about the historical execution of the background process. The system uses this information to control the restart of failed processes. You can use this information to view error messages associated with failed runs.
 - Many processes have been designed to run in parallel in order to speed execution. For example, the process that applies updates for a migration data set import for CMA can be executed so that multiple "threads" are processing a different subset of records (and multiple threads can execute at the same time). Batch control records associated with this type of process organize audit information about each thread in every execution. The system uses this information to control the restart of failed threads. Refer to [Parallel Background Processes](#) for more information.
 - Some processes define extra parameters. These parameters are defined with the batch control. Default values may also be captured for each parameter. They will be used when the [background process is submitted on-line](#).

The following diagram illustrates the relationships that exist for batch control records.



Results of each batch run can be viewed using the [Batch Run Tree](#) page.

Parameters Supplied To Background Processes

All background processes receive the following parameters.

- **Batch code.** Batch code is the unique identifier of the background process.
- **Batch thread number.** Thread number is only used for background processes that can be run in multiple parallel threads. It contains the relative thread number of the process. For example, if the billing process has been set up to run in 20 parallel threads, each of the 20 instances receives its relative thread number (1 through 20). Refer to [Optimal Thread Count for Parallel Background Processes](#) for more information.
- **Batch thread count.** Thread count is only used for background processes that can be run in multiple parallel threads. It contains the total number of parallel threads that have been scheduled. For example, if the billing process has been set up to run in 20 parallel threads, each of the 20 instances receives a thread count of 20. Refer to [Optimal Thread Count for Parallel Background Processes](#) for more information.
- **Batch rerun number.** Rerun number is only used for background processes that download information that belongs to given run number. It should only be supplied if you need to download an historical run (rather than the latest run).
- **Batch business date.** Business date is only used for background processes that use the current date in their processing. For example, billing using the business date to determine which bill cycles should be downloaded. If this parameter is left blank, the system date is used. If supplied, this date must be in the format YYYY-MM-DD. Note: this parameter is only used during QA to test how processes behave over time.
- **Override maximum records between commits.** This parameter is optional and overrides each background process's Standard Commit. You would reduce this value, for example, if you were submitting a job during the day and you wanted more frequent commits to release held resources. You might want to increase this value when a background process is executed at night (or weekends) and you have a lot of memory on your servers.
- **Override maximum minutes between cursor re-initiation.** This parameter is optional and overrides each background process's Standard Cursor Re-Initiation Minutes. You would reduce these values, for example, if you were submitting a job during the day and you wanted more frequent commits to release held resources (or more frequent cursor initiations). You might want to increase these values when a background process is executed at night (or weekends) and you have a lot of memory on your servers.

NOTE: The maximum minutes between cursor re-initiation is relevant for Oracle implementations only. Most of the system background processes contain an outermost loop / cursor. The cursor is opened at the beginning of the process and closed at the end. If Oracle feels that the cursor is open for too long, it may incorrectly interpret this as a problem and may issue an error that the snapshot is too old. The commit processing for the background processes is designed to refresh the cursor based on the minutes between cursor re-initiation in order to prevent this error.

- **User ID.** Please be aware of the following in respect of user ID:
 - The user ID is a user who should have access to all application services in the system. This is because some batch processes update records via services that may be check security.
 - Any batch process that stamps a user ID on a record it creates or updates uses this user ID.
 - This user ID's [display profile](#) controls how dates and currency values are formatted in messages.
- **Password.** Password is not currently used.
- **Language Code.** Language code is used to access language-specific control table values. For example, error messages are presented in this language code.
- **Trace program at start (Y/N), trace program exit (Y/N), trace SQL (Y/N) and output trace (Y/N).** These switches are only used during QA and benchmarking. If trace program start is set to Y, a message is displayed whenever a program is started. If trace program at exist is set to Y, a message is displayed whenever a program is exited. If trace SQL is set to Y, a message is displayed whenever an SQL statement is executed. If output trace is set to Y, special messages formatted by the background process are written.

NOTE: The information displayed when the output trace switch is turned on depends on each background process. It is possible that a background process displays no special information for this switch.

Override Maximum Errors in Batch Process Parameter

Each of the batch processes has, as part of its run parameters, a preset constant that determines how many errors that batch process may encounter before it is required to abort the run. A user can override this constant with an optional additional parameter (defined as either **MAX-ERRORS** or **maxErrors**). If a user chooses not to enter a value for the parameter, the process uses its own preset constant.

The input value must be an integer that is greater than or equal to zero. The maximum valid value for this parameter is 999,999,999,999,999.

The syntax for entering this additional parameter when submitting the batch process is "MAX-ERRORS|maxErrors=PARAM VALUE", where the PARAM VALUE is the desired value (e.g., **MAX-ERRORS=50** or **maxErrors=50**).

Extra Parameters

Some background processes receive additional parameters that are specific to their functionality. When a process receives additional parameters, they are defined and documented in the batch control entry in the application. They are also visible in the [batch control](#) viewer (part of the [application viewer](#)).

The syntax for entering these parameters when submitting the batch process is "PARAM-NAME=PARAM VALUE", where PARAM-NAME is the name of the parameter as indicated in the batch control record. For example, if the batch control includes the **ADD-WORK-DAYS**, with possible values of **Y** and **N**, and you want to pass a value of **N**, enter the following when prompted: **ADD-WORK-DAYS=N**.

Indicating a File Path

Some of the system background processes use extra parameters to indicate a File Path and/or File Name for an input file or an output file. For example, most extract processes use File Path and File Name parameter to indicate where to place the output file.

When supplying a FILE-PATH variable, the directory specified in the FILE-PATH must already exist and must grant write access to the administrator account for the product. You may need to verify a proper location with your systems administrator.

The syntax of the FILE-PATH depends on the platform used for the product application server. Contact your system administrator for verification. For example, if the platform is UNIX, use forward slashes and be sure to put a trailing slash, for example [/spltemp/filepath/](#).

Processing Errors

When a background process detects an error, the error may or may not be related to a specific object that is being processed. For example, if the program finds an error during batch parameter validation, this error is not object-specific. However, if the program finds an error while processing a specific bill, this error is object-specific. The system reports errors in one of the following ways:

- Errors that are not object-specific are written to the error message log in the [Batch Run Tree](#).
- Some batch processes create entries in an "exception table" for certain object-specific errors. For example, an error detected in the creation of a bill in Oracle Utilities Customer Care and Billing may be written to the bill exception table. If an error is written to an exception table, it does not appear in the batch run tree. For each exception table, there is an associated to do entry process that creates a To Do Entry for each error to allow a user to correct the problem on-line.
- For some background processes, errors that do not result in the creation of an exception record may instead generate a To Do entry directly. For these processes, if you wish the system to directly create a To Do entry, you must configure the To Do type appropriately. Refer to [To Do entry for object-specific errors](#) for information about configuring the To Do

type. If the background process detects an object specific error AND you have configured the system to create a To Do entry, the error is not written to the batch run tree. If you have configured your To Do type to not create To Do entries for certain errors, these errors are written to the [batch run tree](#).

NOTE: Some processes create exceptions and To Do entries. It is possible for a background process to create entries in an exception table and create To Do entries directly, depending on the error. Consider batch billing in Oracle Utilities Customer Care and Billing; any conditions that cause a bill or bill segment to be created in **error** status result in a record added to the bill exception table or the bill segment exception table. However, any object-specific error that is not related to a specific bill or bill segment or any error that prevents a bill or bill segment from being created may result in a To Do entry for the object-specific error.

Parallel Background Processes

Many processes have been designed to run in parallel in order to speed execution. For example, the billing process in Oracle Utilities Customer Care and Billing can be executed so that bills are produced in multiple "threads" (and multiple threads can execute at the same time).

NOTE: The detailed description in the metadata for each batch control provided with the system should indicate if it may be run in parallel.

By default the system distributes the data for the multiple threads using the primary key of the main table of the process. For example, if you are running BILLING in Oracle Utilities Customer Care and Billing with multiple threads, the batch process distributes the accounts across the multiple threads. The Account ID is 10 digits long so if you run it with 4 threads, the records are processed as follows:

- Thread 1: Account IDs 0000000001 - 2500000000
- Thread 2: Account IDs 2500000001 - 5000000000
- Thread 3: Account IDs 5000000001 - 7500000000
- Thread 4: Account Ids 7500000001 - 9999999999

Note that the multi-threading logic relies on the fact that primary keys for master and transaction data are typically system generated random keys.

NOTE: Overriding the thread ranges. Your implementation has the ability to override the thread ranges if certain data in your system takes longer to process. For example, imagine you have a single account in Oracle Utilities Customer Care and Billing that has thousands of service agreements (maybe the account for a large corporation or a major city). You may want to set up the thread ranges to put this large account into its own thread and distribute the other accounts to the other threads. To do this, you should create the appropriate batch thread records ahead of time in a status of **Thread Ready (50)** with the key ranges pre-populated. Note that the base product does not provide the ability to add batch thread records online. If you are interested in more information about this technique, contact Customer Support.

Optimal Thread Count

Running a background process in multiple threads is almost always faster than running it in a single thread. The trick is determining the number of threads that is optimal for each process.

NOTE: A good rule of thumb is to have one thread for every 100 MHz of application server CPU available. For example if you have four 450 MHz processors available on your application server, you can start with 18 threads to begin your testing: $(450 * 4) / 100 = 18$.

This is a rule of thumb because each process is different and is dependent on the data in your database. Also, your hardware configuration (i.e., number of processors, speed of your disk drives, speed of the network between the database server and

the application server) has an impact on the optimal number of threads. Please follow these guidelines to determine the optimal number of threads for each background process:

- Execute the background process using the number of threads dictated by the rule of thumb (described above). During this execution, monitor the utilization percentage of your application server, database server and network traffic.
- If you find that your database server has hit 100% utilization, but your application server hasn't one of the following is probably occurring:
 - There may be a problematic SQL statement executing during the process. You must capture a database trace to identify the problem SQL.
 - It is also possible that your commit frequency may be too large. Commit frequency is a parameter supplied to every background process. If it is too large, the database's hold queues can start swapping. Refer to [Parameters Supplied to Background Processes](#) for more information about this parameter.
- It is normal if you find that your application server has hit 100% utilization but your database server has not. This is normal because, in general, all processes are CPU bound and not IO bound. At this point, you should decrease the number of threads until just under 100% of the application server utilization is achieved. And this will be the optimal number of threads required for this background process.
- If you find that your application server has NOT hit 100% utilization, you should increase the number of threads until you achieve just under 100% utilization on the application server. And remember, the application server should achieve 100% utilization before the database server reaches 100% utilization. If this proves not to be true, something is probably wrong with an SQL statement and you must capture an SQL trace to determine the culprit.

Another way to achieve similar results is to start out with a small number of threads and increase the number of threads until you have maximized throughput. The definition of "throughput" may differ for each process but can be generalized as a simple count of the records processed in the batch run tree. For example, in the Billing background process in Oracle Utilities Customer Care and Billing, throughput is the number of bills processed per minute. If you opt to use this method, we recommend you graph a curve of throughput vs. number of threads. The graph should display a curve that is steep at first but then flattens as more threads are added. Eventually adding more threads will cause the throughput to decline. Through this type of analysis you can determine the optimum number of threads to execute for any given process.

Timed Batch Processes

Most batch jobs are submitted via a batch scheduler. In the absence of a scheduler, a batch control may be configured as "timed" triggering the framework to monitor and schedule these batch jobs as defined by the timer interval. The timer interval defines the desired interval between starts (in seconds). The system schedules new batch runs at each interval if the last instance of the job has completed.

When configuring a batch control as "timed", other default information must be provided, including the User ID and Language to use for submitting the job and the email address for notification, if desired.

Timed batch controls also include an Active setting, allowing for an implementation to temporarily stop further executions of the batch job (but retain the other timer settings).

Timed jobs are controlled by the default threadpool and not by a scheduler. When the **DEFAULT** threadpoolworker starts it will start executing any job for a Batch Control configured as **Timed** with the Timer Active set to **Yes**. This is whether the batch daemon or batch server is enabled or not.

How to Re-extract Information

If you need to recreate the records associated with an historical execution of an extract process, you can - simply supply the desired batch number when you request the batch process.

How to Submit Batch Jobs

Most batch jobs are submitted via a batch scheduler.

In addition, you can manually submit your adhoc background processes or submit a special run for one of your scheduled background processes using the online [batch job submission](#) page.

How to Track Batch Jobs

You can track batch jobs using the batch process pages, which show the execution status of batch processes. For a specified batch control id and run id, the tree shows each thread, the run-instances of each thread, and any messages (informational, warnings, and errors) that might have occurred during the run.

FASTPATH: For more information, refer to [Tracking Batch Processes](#).

How to Restart Failed Jobs and Processes

Every process in the system can be easily restarted if it fails (after fixing the cause of the failure). All you have to do is resubmit the failed job; the system handles the restart.

Assessing Level of Service

For some background processes, an implementation may wish to supply an algorithm that checks some conditions to assess whether or not the process is performing as expected. This algorithm could be simply verifying that the batch process is running as frequently as expected. Or it could be used to check the performance of the job to see if it is running as efficiently as expected. Or it could analyze the data processed by the background process to assess whether there may be some problem with the quality of the data.

The system provides a Level of Service plug-in spot on [batch control](#) to configure the appropriate algorithm for a given background process, if desired. The algorithm is expected to return a value to indicate the 'level of service' determined along with a message indicating the reason for the value. The following Level of Service values are supported:

- **Normal.** Indicates that the algorithm did not detect any issues.
- **Warning.** Indicates that the algorithm found some issues that may or may not indicate a problem.
- **Error.** Indicates that the algorithm found some issues that should be investigated.
- **Disabled.** Indicates that the algorithm could not properly execute the level of service logic.

When viewing a [batch control](#) record, if there is a level of service algorithm configured, its logic is executed and the results are displayed. The level of service is also part of the [Health Check](#) service.

System Background Processes

NOTE: List of system background processes. The list of background processes provided in the base product may be viewed in the [application viewer's batch control](#) viewer. In addition if your implementation adds batch control records, you may [regenerate](#) the application viewer to see your additions reflected there.

Defining Batch Controls

The system is delivered with all necessary batch controls. Implementations may define default values for parameters. In addition, implementations may define their own background processes.

To view background processes, open **Admin > System > Batch Control**. Refer to [Background Processing Concepts](#) for more information.

CAUTION: Important! If you introduce a new batch process, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

Description of Page

Enter an easily recognizable **Batch Process** and **Description** for each batch process.

Owner indicates if this batch control is owned by the base package or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add a batch control. This information is display-only.

Use the **Detailed Description** to describe the functionality of the batch process in detail.

Use **Batch Control Type** to define the batch process as either **Timed** or **Not Timed**. A [Timed batch process](#) will be automatically initialized on a regular basis. A Not Timed process needs to be run manually or through a scheduler.

Use **Batch Control Category** to categorize the process for documentation purposes. The base values provided are as follows:

- **Ad Hoc.** Processes of this type are run on an ad hoc basis, only when needed. For example, if there is a process to do a mass cancel / correction of data, it would only be run when a situation occurs requiring this.
- **Extract.** Extract processes extract information that is interfaced out of the system. Processes of this type typically extract records marked with a given run number. If the requester of the process does not supply a specific run number, the system assumes that the latest run number should be extracted. If you need to re-extract an historical batch, you simply supply the respective run number when you request the batch process.
- **ILM. [Information Lifecycle Management](#)** jobs are crawler background processes that are associated with the ILM based storage solution.
- **Monitor.** Processes of this type are processes related to business objects with a lifecycle state that defines [monitor algorithms](#). The monitor process selects records in a given state and executes its algorithms, which may cause the record to transition to another state or may trigger some other logic to occur. Using configuration, the monitor process may target only specific records. Refer to [Monitoring Batch Processes](#) for more information. Note that these types of background processes can be considered a subset of **Process What's Ready**
- **Process What's Ready.** Processes of this type create and update records that are “ready” for processing. The definition of “ready” differs for every process. For example, a payment upload process creates payments for every record that is **pending**. An overdue event monitor activates pending overdue events that have reached their trigger date.
- **To Do Entry.** Processes of this type are used to detect a given situation and create a To Do Entry. Refer to [To Do Entries Created by Background Processes](#) for more information.
- The following categories are related to the data conversion / migration processes, which are not applicable to all products:
 - **Conversion.** Processes of this type are dedicated to converting or migrating data from external applications into the product.
 - **Object Validation.** Processes of this type are dedicated to validate data within objects for conversion or migration purposes.
 - **Referential Integrity.** Processes of this type are dedicated to validate referential integrity within objects for conversion or migration purposes.

NOTE: Additional categories may be introduced by your specific product.

If the batch process is Timed, then the following fields are available:

- **Timer Interval** is the number of seconds between batch process submissions. The system will start the next run this many seconds from the start time of the previous run.
- **User ID** is the ID under which the batch process will run.
- **Email Address** is the Email address to be used for notification if the batch process fails.
- **Timer Active** allows you to temporarily switch off the timer while retaining the other settings for the timed job.
- **Batch Language** is the language associated with the batch process.

Use **Program Type** to define if the batch process program is written in **Java** or **Java (Converted)**, meaning that the program has been converted into Java.

NOTE: **Java (Converted)** program types are not applicable to all products.

Use **Program Name** to define the Java class / program associated with your batch process:

NOTE: View the source. If the program is shipped with the base package, you can use the adjacent button to display the source code of this program in the [Java docs viewer](#).

Level of Service shows the output of the [level of service](#) algorithm for the batch control. It shows the level of service lookup value along with a message indicating the reason for the output value. Note that if no level of service algorithm is found, then the value **Disabled** is shown with a message indicating that no algorithm is provided for this batch control.

The **Last Update Timestamp**, **Last Update Instance** and **Next Batch Nbr** are used for audit purposes.

Turn on **Accumulate All Instances** to control how this batch control is displayed in the [Batch Run Tree](#). If checked, the run statistics (i.e., "Records Processed" and "Records in Error") for a thread are to be accumulated from all the instances of the thread. This would include the original thread instance, as well as any restarted instances. If this is not turned on, only the ending (last) thread instance's statistics are used as the thread's statistics. This may be preferred for certain types of batch processes where the accumulation would create inaccurate thread statistics, such as those that process flat files and, therefore, always start at the beginning, even in the case of a restart.

The following fields are default values that are used when a batch job is submitted for the batch control:

- Use **Thread Count** to control whether a background process is run single threaded or in multiple parallel threads. This value defines the total number of threads that have been scheduled.
- Select **Trace Program Start** if you want a message to be written whenever a program is started.
- Select **Trace SQL** if you want a message to be written whenever an SQL statement is executed.
- Use **Override Nbr Records to Commit** to define the default number of records to commit. This is used as the default value for timed jobs as well as online submission of jobs that are not timed.
- Select **Trace Program Exit** if you want a message to be written whenever a program is exited.
- Select **Trace Output** if you want a message to be displayed for special information logged by the background process.

For more information about these fields, see [Batch Job Submission - Main](#)

The parameter collection is used to define additional parameters required for a particular background process. The following fields should be defined for each parameter:

Sequence. Defines the relative position of the parameter.

Parameter Name. The name of the parameter as defined by the background process program.

Description. A description of the parameter.

Detailed Description. A more detailed description of the parameter.

Required. Indicates whether or not this is a required parameter.

Parameter Value. The default value, if applicable. Note that an implementation may define a default value for base provided batch controls.

Security. Indicates whether the system should **Encrypt** the parameter value or not. A value of **Encrypt** means that the parameter value is stored in the database and written to the log files using encryption. In addition, the parameter is written to the log files with asterisks. The setting applies to values entered here as well as in the online [Batch Submission](#). If there is no need to secure the parameter value, use the default setting of **None**.

Owner Indicates if this batch process is owned by the base package or by your implementation (Customer Modification). The system sets the owner to **Customer Modification** when you add a batch process. This information is display-only.

Batch Control - Algorithms

Use this page to maintain a batch control's algorithms. Open this page using **Admin > System > Batch Control** and then navigate to the **Algorithms** tab.

Description of Page

The **Algorithms** grid contains algorithms that control important functions for instances of this batch control. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).
- Specify the **Sequence Number** and **Algorithm** for each system event. You can set the **Sequence Number** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

The following table describes the **System Events**.

System Event	Optional / Required	Description
Level of Service	Optional	Algorithms of this type are called to determine the Level of Service provided by a batch control. Refer to Assessing Level of Service for more information. Click here to see the algorithm types available for this system event.

On-Line Batch Submission

The on-line [batch submission](#) page enables you to request a specific background process to be run. When submitting a background process on-line, you may override standard system parameters and you may be required to supply additional parameters for your specific background process. After submitting your background process, you may use this page to review the status of the submission.

The following topics further describe logic available for on-line submission of background processes.

Batch Submission Creates a Batch Run

When you request a batch job to be submitted from on-line, the execution of the desired background process will result in the creation of a batch run. Just as with background processes executed through your scheduler, you may use the [Batch Run Tree](#) page to view the status of the run, the status of each thread, the run-instances of each thread, and any messages that might have occurred during the run.

NOTE: Your on-line submission record is assigned a status value so that you may know whether your job has been submitted and whether or not it has ended, however, it will not contain any information about the results of the background process itself. You must navigate to the [Batch Run Tree](#) page to view this detail.

Jobs Submitted in the Background

When you save a record on the batch job submission page, the batch job does not get submitted automatically. Rather, it saves a record in the batch job table. A special background process will periodically check this table for pending records and will execute the batch job. This background process will update the status of the batch job submission record so that a user can determine when their job is complete.

NOTE: At installation time, your system administrator will set up this special background process to periodically check for pending records in the batch job submission table. Your administrator will define how often the system will look for pending records in this table.

It should be noted that this special background process only submits one pending batch job submission record at a time. It submits a job and waits for it to end before submitting the next pending job.

NOTE: If you request a batch job to be run multi-threaded, the special background process will submit the job as requested. It will wait for all threads to complete before marking the batch job submission record as **ended**. Refer to [Running Multi-threaded Processes](#) for more information.

Email Notification

If you wish the system to inform you when the background process completes, you may supply your email address. The email you receive will contain details related to the batch job's output; similar to the job results you would see from your batch scheduler.

NOTE: This assumes that during the installation process, your system administrator configured the system to enable email notification. Your administrator may also override the amount of detail included in the email notification.

Running Multi-Threaded Processes

Many of the system background processes may be run multi-threaded. When submitting a background process on-line, you may also run a multi-threaded process or run a single thread of a multi-threaded process. The fields Thread Count and Thread Number on the [batch submission](#) page control the multi-threaded process requests:

- To run a multi-threaded process, indicate the number of threads in **Thread Count** and enter **0** in the **Thread Number**.
- To run a single thread in a multi-threaded process, indicate the number of threads in Thread Count and indicate the Thread Number you would like to run.
- To run a process as a single thread, enter Thread Count = **1** and Thread Number = **1**. This will execute the background process single-threaded.

NOTE: When running a multi-threaded process, the special background process will wait until all threads are complete before marking the batch job submission record as **Ended**.

Batch Jobs May End in Error

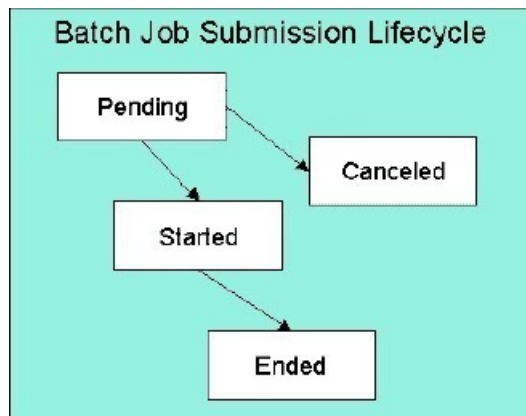
It is possible for your background process to end with an error. When this occurs, your batch job submission record will still be marked as **Ended**. You will need to navigate to the [Batch Run Tree](#) page to determine the status of the batch run.

Submitting Jobs in the Future

If you wish to request a batch job to be submitted in the future, you may do so when creating your batch job submission record by entering a future submission date. The special background process, which looks for pending records in the batch job submission table, will only submit batch jobs that do not have a future submission date.

Lifecycle of a Batch Job Submission

The following diagram illustrates the lifecycle of a batch job submission record.



Pending — Records are created in **Pending** status. Records in this state are put in a queue to be submitted.

Canceled— Users may cancel a pending record to prevent the batch job from being submitted.

Started— Once a pending record has been submitted for processing, its status will be changed to **Started**. Records in this status may not be canceled.

Ended— When the batch job has finished processing, its status will be changed to **Ended**. Note that records in **Ended** status may have ended in error. Refer to [Batch Jobs May End in Error](#) for more information.

Batch Job Submission - Main

This page allows you to submit a batch job on-line. Navigate to this page using **Menu > Batch > Batch Job Submission**.

Description of Page

The **Batch Job ID** is a system generated random number that identifies a particular submission.

To submit a batch job, choose the **Batch Code** for the process you wish to submit.

NOTE: List of system background processes. The list of background processes provided in the base product may be viewed in the [application viewer's batch control](#) viewer.

The following parameters are provided with each background process:

Thread Number is used to control whether a background processes is run single threaded or in multiple parallel threads. It contains the relative thread number of the process. For example, if the process has been set up to run in 20 parallel threads, each of the 20 instances receives its relative thread number (1 through 20). Refer to [Running Multi-threaded Processes](#) for more information about populating this field.

NOTE: Not all processes may be run multi-threaded. Refer to the description of a batch control to find out if it runs multi-threaded.

Thread Count is used to control whether a background processes is run single threaded or in multiple parallel threads. It contains the total number of threads that have been scheduled. For example, if the process has been set up to run in 20 parallel threads, each of the 20 instances receives a thread count of 20. Refer to [Running Multi-threaded Processes](#) for more information about populating this field.

Batch Rerun Number is only used for background processes that download information that belongs to given run number. It should only be supplied if you need to download an historical run (rather than the latest run).

Batch Business Date is only used for background processes that use a date in their processing. In Oracle Utilities Customer Care and Billing, for example, a billing process can use the business date to determine which bill cycles should be downloaded. If this parameter is left blank, the system date is used at the time the background process is executed.

NOTE: Saving a record on this page does not submit the batch job immediately. A special background process will run periodically to find pending records and submit them. Depending on how often the special process checks for pending records and depending on how many other pending records are in the 'queue', there may be a slight lag in submission time. If the desired execution date/time is close to midnight, it is possible that your batch job will run on the day after you submit it. If you have left the business date blank in this case, keep in mind that your business date would be set to the day after you submit the job.

Override Nbr Records To Commit and **Override Max Timeout Minutes**. These parameters are optional and override each background process's Standard Commit Records and Standard Cursor Re-Initiation Minutes. (Each background process's Standard Commit Records / Standard Cursor Re-Initiation Minutes should be documented in the detailed description of the batch control record). Note that Max Timeout Minutes corresponds to the Cursor Re-initiation Minutes.

FASTPATH: Refer to [Parameters Supplied to Background Processes](#) for more information.

User ID is the user ID associated with the run of the background process. Refer to [Parameters Supplied to Background Processes](#) for more information about the significance of the user id.

NOTE: This field defaults to the id of the current user.

Password is not currently used.

Language Code is used to access language-specific control table values. For example, error messages are presented in this language code.

If you wish the system to notify you when the batch job is complete, enter your **Email ID**. Refer to [Email Notification](#) for more information.

NOTE: This field defaults to the email address for the current user, if populated on the [user](#) record.

The **Desired Execution Date/Time** defaults to the current date and time. Override this information if you wish the background process to be executed at some future date and time. Refer to [Submitting Jobs in the Future](#) for more information.

The **Batch Job Status** indicates the current status of the batch job. Refer to [Lifecycle of a Batch Job Submission](#) for more information.

The **Program Name** associated with the batch control code is displayed.

The following trace parameters may also be supplied to a background process and are only used during QA and benchmarking.

- **Trace Program Start** Turn on this switch if you wish a message to be written whenever a program is started.
- **Trace Program Exit** Turn on this switch if you wish a message to be written whenever a program is exited.

- **Trace SQL** Turn on this switch if you wish a message to be written whenever an SQL statement is executed.
- **Trace Output** Turn on this switch if you wish a message to be displayed for special information logged by the background process.

NOTE: The information displayed when the trace output switch is turned on depends on each background process. It is possible that a background process displays no special information for this switch.

NOTE: The location of the output of this trace information is defined by your system administrator at installation time.

If additional parameters have been defined for this background process on the Batch Control page, the **Parameter Name**, **Description**, **Detailed Description** and an indicator of whether or not the parameter is **Required** are displayed.

If a default parameter value is configured on the batch control configuration, that value is shown and may be overridden. Confirm or enter the appropriate **Parameter Value** for each parameter. Note that if the parameter value is configured to be Encrypted on the batch control configuration, the value here will be shown encrypted.

Once you have entered all the desired values, **Save** the record in order to include it in the queue for background processes.

If you wish to duplicate an existing batch job submission record, including all its parameter settings, display the submission record you wish to duplicate and use the **Duplicate and Queue** button. This will create a new Batch Job Submission entry in pending status. The new submission entry will be displayed.

If you wish to cancel a **Pending** batch job submission record use the **Cancel** button. The button is disabled for all other status values.

Tracking Batch Processes

The batch process pages show the execution status of batch processes. For a specified batch control id and run id, the tree shows each thread, the run-instances of each thread, and any messages (informational, warnings, and errors) that might have occurred during the run. Refer to [Defining Batch Controls](#) for more information on how batch control codes are defined.

Batch Run Tree - Main

This page allows you to view the status of a specific execution of a batch job. Navigate to this page using **Menu > Batch > Batch Run Tree**.

Description of Page

Select a **Batch Control** process and **Batch Number** to view information and statistics on the batch run's "threads". The following points should help understand this concept:

- Many batch jobs cannot take advantage of your hardware's processing power when they are run singularly. Rather, you'll find that a large percentage of the CPU and/or disk drives are idle.
- In order to minimize the amount of idle time (and increase the throughput of your batch processes), we allow you to set up your batch processes so that multiple instances of a given batch job are executed at the same time. For example, in Oracle Utilities Customer Care and Billing when you schedule the **billing** process, you can indicate that multiple parallel instances should be executed (rather than just one instance). You'd do this so that the processing burden of creating bills for your customers can be spread over multiple processes.
- We refer to each parallel execution of a batch process as a "thread".
- Statistics and information messages are displayed in respect of each thread. Why? Because each thread is a separate execution and therefore can start and end at different times.

The tree includes a node that displays the total number of records processed for the batch run, the total number of records in error for the batch run and the batch run elapsed time. The elapsed time is the longest elapsed time among the batch thread(s). The message is red if there are any records in error.

If the background process has been enabled to create [To Do entries for object specific errors](#), information about the To Do entries are displayed in the tree. This information is not displayed for each thread, but rather all the To Do entries created for the batch run are grouped together. The To Do entries are grouped by their status.

The messages that appear under a thread always show the start and end times of the execution instance. If errors are detected during the execution of the thread, these error messages may also appear in the tree. Refer to [Processing Errors](#) for information about the types of errors that appear in the batch run tree.

Batch Run Tree - Run Control

By default, if a batch process fails, it will restart. This notebook tab allows you to modify the restart status of a failed run.

Navigate to this page using **Menu > Batch > Batch Run Tree** search for the desired batch control and then navigate to the **Run Control** page.

Description of Page

On the main page, you must select a **Batch Control**, **Batch Number**, and **Batch Rerun Number** to view a tree of the batch run. On this page, the following information is displayed:

- **Date Time** contains the date and time this batch run last start or last completed.
- **Batch Business Date** is the business date that was supplied to the background process (this date is used as the "system date" by the process).

Run Status indicates the status of the batch run. Valid values are: **In Progress**, **Error**, and **Complete**.

If the **Run Status** is **Error**, the system will attempt to restart this run when you attempt to execute the **Batch Control**.

In most situations, this is exactly what you want to happen. However, there are rare situations where you do not want the system to execute a given batch run (e.g., if this run is somehow corrupt and you cannot correct the data for whatever reasons). If you want the system to skip the execution of a batch run (and proceed to the next run), turn on **Do Not Attempt Restart**.

Service Health Check

The system provides a service that returns an overall assessment of the system's health. The overall response is expressed using an HTTP code. The codes supported by the service are defined in the lookup **HEALTH_RESPONSE_FLG** and are as follows:

- A value of 500 (One or More Critical Functions Degraded) is returned if there is any critical issue detected by the service.
- A value of 203 (Non-Critical Function Degraded) is returned if no critical issues are found and there is at least one non-critical issue detected by the service.
- Otherwise a value of 200 (All Checks Successful) is returned.

This health check service is accessible through the business service **F1-HealthCheck**. Also note that the system provides an [Inbound Web Service](#) for this business service (also called **F1-HealthCheck**) allowing external systems to use a web service to retrieve this information.

In addition, the product provides a portal that allows a user to view the detailed results of the health check service.

Navigate using **Admin > System > Health Check** to view this portal.

The zone on the portal displays the following:

Overall Response is the HTTP response code returned by the business service as described above.

The grid displays the detail of each individual component checked as part of the system health check. See below for more detail about what components are checked.

- **Health Component** indicate the type of health check performed. Currently the system supports **Batch Level of Service** component type.
- **Health Component Details** provides information about the individual entity checked. The information displayed here depends on the type of health component. If supported by the type of component, the column may be hypertext, allowing the user to drill into the entity itself.
- **Status** displays the status returned by the health component check for this individual entity. The information displayed here depends on the type of health component.
- **Status Reason** provides more detail about why the individual entity is in the indicated status. The information displayed here depends on the type of health component.
- **Response** shows the health check response code assigned to this individual entity's health check response. It is mapped based on the status returned by the health component check.

Health Component Type Details

The details returned by the business service for this portal depends on the health component type. The system supports the **Batch Level of Service** health component type.

This health component type finds all the batch controls that are configured with a [level of service](#) algorithm and invokes the algorithm for each batch control. The business service populates the output for this health service for each batch control as follows:

- The **Health Component Detail** is populated with the Batch Control code and description. In addition, the navigation information for being able to drill into the batch control are provided and used to build the column as hypertext.
- The **Status** is populated with the description of the Level of Service lookup value returned by the level of service algorithm for this batch control.
- The **Status Reason** is populated with the expanded text of the status reason returned by the level of service algorithm for this batch control.
- The **Response** is populated based on the value of the Level of Service status. It is set to **All Checks Successful** (200) when the Level of Service is **Normal** or **Disabled; Non-Critical Function Degraded** (203) when the Level of Service is **Warning** and **One or More Critical Functions Degraded** (500) when the Level of Service is **Error**.

The Big Picture of Requests

Requests enable an implementer to design an ad-hoc batch process using the configuration tools.

An example of such a process might be to send an email to a group of users that summarizes the To Do entries that are assigned to them. This is just one example. The request enables many types of diverse processing.

Request Type Defines Parameters

For each type of process that your implementation wants, you must configure a request type to capture the appropriate parameters needed by the process.

Previewing and Submitting a Request

To submit a new request, go to **Menu > Batch > Request > Add**. You must select the appropriate request type and then enter the desired parameter values, if applicable.

After entering the parameters, the following actions are possible:

- Click **Save** to submit the request.

- Click **Cancel** to cancel the request.
- Click **Preview** to see a sample of records that satisfy the selection criteria for this request. This information is displayed in a separate map. In addition, the map displays the total number of records that will be processed when the request is submitted. From this map you can click **Save** to submit the request, **Back** to adjust the parameters, or **Cancel** to cancel the request.

When a request is saved, the job is not immediately submitted for real time processing. The record is saved with the status **Pending** and a monitor process for this record's business object is responsible for transitioning the record to **Complete**.

As long as the record is still **Pending**, it may be edited to adjust the parameters. The preview logic described above may be repeated when editing a record.

The actual work of the request, such as generating an email, is performed when transitioning to **Complete** (using an enter processing algorithm for the business object).

To Do Summary Email

The base product includes a sample request process that sends an email to users that have incomplete To Dos older than a specified number of days.

The following configuration tasks are required to use this process:

- Define an Outbound Message Type. The business object usually defined for the Outbound Message Type is **F1-EmailMessage**.
- Define an External System that contains the Outbound Message Type in one of its steps. In the configuration determine if the communication is through SOA, batch, or real-time processing method when sending the email notification. Refer to [Outbound Messages](#) for more information about configuration needed for the different processing methods.
- Create a Request Type that includes the Outbound Message Type and the corresponding External System.
- Create a Request for the created Request Type.

Exploring Request Data Relationships

Use the following links to open the application viewer where you can explore the physical tables and data relationships behind the request functionality:

- Click [F1-REQ-TYPE](#) to view the request type maintenance object's tables.
- Click [F1-REQ](#) to view the request maintenance object's tables.

Defining a New Request

To design a new ad-hoc batch job that users can submit via Request, first create a new Request Type business object. The base product BO for request type, **F1-ToDoSumEmailTyp**, may be used as a sample.

The business object for the request includes the functionality for selecting the records to process, displaying a preview map for the user to review, and for performing the actual processing. The base product BO for request, **F1-ToDoSumEmailReq**, may be used as a sample. The following points highlight the important configuration details for this business object:

- Special BO options are available for request BOs to support the Preview Request functionality.
 - **Request Preview Service Script.** This script retrieves the information that is displayed when a user asks for a preview of a request.
 - **Request Preview Map.** This map displays the preview of a request.
- The enter algorithm plugged into the Complete state is responsible for selecting the records that satisfy the criteria and processing the records accordingly.

Setting Up Request Types

Use the Request Type portal to define the parameters to capture when submitting a request. Open this page using **Admin > Request Type > Add**.

This topic describes the base-package zones that appear on the Request Type portal.

Request Type List. The Request Type List zone lists every request type. The following functions are available:

- Click a **broadcast** icon to open other zones that contain more information about the request type.
- Click **Add** in the zone's title bar to add a new request type.

Request Type. The Request Type zone contains display-only information about a request type. This zone appears when a request type has been broadcast from the Request Type List zone or if this portal is opened via a drill down from another page. The following functions are available:

- Click **Edit** to start a business process that updates the request type.
- Click **Delete** to start a business process that deletes the request type.
- Click **Deactivate** start a business process that deactivates the request type.
- Click **Duplicate** to start a business process that duplicates the request type.
- State transition buttons are available to transition the request type to an appropriate next state.

Please see the zone's help text for information about this zone's fields.

Maintaining Requests

Use the Request transaction to view and maintain pending or historic requests.

Open this page using **Menu > Batch > Request > Search**. This topic describes the base-package portals and zones on this page.

Request Query. Use the query portal to search for an existing request. Once a request is selected, you are brought to the maintenance portal to view and maintain the selected record.

Request Portal. This portal appears when a request has been selected from the Request Query portal. The following base-package zones appear on this portal:

- **Actions.** This is a standard actions zone.
- **Request.** The Request zone contains display-only information about a request. Please see the zone's help text for information about this zone's fields.
- **Request Log.** This is a standard log zone.

Defining Algorithms

In this section, we describe how to set up the user-defined algorithms that perform many important functions including:

- Validating the format of a phone number entered by a user.
- Validating the format of a latitude/longitude geographic code entered by a user.
- In products that support payment and billing:
 - Calculating late payment charges.
 - Calculating the recommended deposit amount.

- Constructing your GL account during the interface of financial transactions to your GL
- And many other functions...

The Big Picture Of Algorithms

Many functions in the system are performed using a user-defined algorithm. For example, when a CSR requests a customer's recommended deposit amount, the system calls the deposit recommendation algorithm. This algorithm calculates the recommended deposit amount and returns it to the caller.

NOTE: Algorithm = Plug-in. We use the terms plug-in and algorithm interchangeably throughout this documentation.

So how does the system know which algorithm to call? When you set up the system's control tables, you define which algorithm to use for each component-driven function. You do this on the control table that governs each respective function. For example:

- You define the algorithm used to validate a phone number on your phone types.
- You define the algorithm in Oracle Utilities Customer Care and Billing used to calculate late payment charges on each Service Agreement Type that has late payment charges.
- The list goes on...

The topics in this section provide background information about a variety of algorithm issues.

Algorithm Type Versus Algorithm

You have to differentiate between the type of algorithm and the algorithm itself.

- An **Algorithm Type** defines the program that is called when an algorithm of this type is executed. It also defines the types of parameters that must be supplied to algorithms of this type.
- An **Algorithm** references an **Algorithm Type**. It also defines the value of each parameter. It is the algorithm that is referenced on the various control tables.

FASTPATH: Refer to [How to Add A New Algorithm](#) for an example that will further clarify the difference between an algorithm and an algorithm type.

How To Add A New Algorithm

Before you can add a new algorithm, you must determine if you can use one of the sample algorithm types supplied with the system. Refer to [List of Algorithm Types](#) for a complete list of algorithm types.

If you can use one of the sample algorithm types, simply add the algorithm and then reference it on the respective control table. Refer to [Setting Up Algorithms](#) for how to do this.

If you have to add a new algorithm type, you may have to involve a programmer. Let's use an example to help clarify what you can do versus what a programmer must do. Assume that you require an additional geographic type validation algorithm. To create your own algorithm type you must:

- Write a new program to validate geographic type in the appropriate manner. Alternatively, you may configure a plug-in script to implement the validation rules. The advantage of the latter is that it does not require programming. Refer to [plug-in script](#) for more information.
- Create an Algorithm Type called **Our Geographic Format** (or something appropriate). On this algorithm type, you'd define the name of the program (or the plug-in script) that performs the function. You'd also define the various parameters required by this type of algorithm.

- After creating the new Algorithm Type, you can reference it on an Algorithm.
- And finally, you'd reference the new Algorithm on the Geographic Type that requires this validation.

Minimizing The Impact Of Future Upgrades

The system has been designed to use algorithms so an implementation can introduce their own logic in a way that's 100% upgradeable (without the need to retrofit logic). The following points describe strong recommendations about how to construct new algorithm type programs so that you won't have to make program changes during future upgrades:

- Do not alter an algorithm type's hard parameters. For example, you might be tempted to redefine or initialize parameters defined in an algorithm type's linkage section. Do not do this.
- Follow the naming conventions for the new algorithm type code and your source code, i.e., both the source code and the algorithm type should be prefixed with "CM". The reason for this naming convention is to make it impossible for a new, base-package algorithm type from overwriting your source code or algorithm type meta-data (we will never develop a program or introduce meta-data beginning with CM).
- Avoid using inline SQL to perform insert/update/delete. Rather, invoke the base-package's object routines or common routines.
- Avoid using base messages (outside of common messages, i.e., those with a message number < 1000) as we may deprecate or change these messages in future releases. The most common problem is caused when an implementation clones a base package algorithm type program because they need to change a few lines of logic. Technically, to be 100% upgradeable, you should add new messages in the "90000" or greater category (i.e., the category reserved for implementation-specific messages) for every message in your new program even though these messages may be duplicates of those in the base package.

Setting Up Algorithm Types

The system provides many algorithm types to support base product functionality. If you need to introduce a new type of algorithm, open **Admin > System > Algorithm Type**.

FASTPATH: Refer to [The Big Picture Of Algorithms](#) for more information.

CAUTION: Important! If you introduce a new algorithm type, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

Description of Page

Enter an easily recognizable **Algorithm Type** and **Description**.

Owner indicates if this algorithm type is owned by the base package or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add an algorithm type. This information is display-only.

Enter a **Detailed Description** that describes, in detail, what algorithms of this type do.

Use **Algorithm Entity** to define where algorithms of this type can be "plugged in". If a detailed description about an algorithm entity is available, a small help icon is visible adjacent to the dropdown. Click the icon to view the information.

NOTE: The values for this field are customizable using the [lookup](#) table. This field name is **ALG_ENTITY_FLG**.

Use **Program Type** to define if the algorithm's program is written using **Java**, a **Plug-In Script**, or **Java (Converted)**, meaning the program has been converted to Java.

NOTE: **Java (Converted)** program types are not applicable to all products.

Use **Program Name** to define the program to be invoked when algorithms of this type are executed:

- If the Program Type is **Java (Converted)**, enter the name of the converted program.
- If the Program Type is **Java**, enter the Java class name.
- If the Program Type is **Plug-In Script**, enter the plug-in [script](#) name. Only plug-in scripts defined for the algorithm entity may be used.

NOTE: View the source. If the program is shipped with the base package, you can use the adjacent button to display the source code of this program in the [Java docs viewer](#). For plug-in scripts, drill into the plug-in script to view the details.

Use the **Parameter Types** grid to define the types of parameters that algorithms of this type use. The following fields should be defined for each parameter:

- Use **Sequence** to define the relative position of the **Parameter**.
- Use **Parameter** to describe the verbiage that appears adjacent to the parameter on the Algorithm page.
- Indicate whether the parameter is **Required**. This indicator is used when parameters are defined on algorithms that reference this algorithm type.
- **Owner** indicates if the parameter for this algorithm type is owned by the base package or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add an algorithm type with parameters. This information is display-only.

NOTE: When adding a new algorithm type that is for a Java program, the parameters are automatically generated based on the Java code. Once an algorithm type exists, any additional parameters defined in the Java code should be manually added to the algorithm type. For other program types, algorithm type parameters must be manually defined.

NOTE: When a new algorithm type parameter is added for any program type, existing algorithms for the algorithm type do not automatically get updated with the new parameter. The algorithms must be manually updated.

Where Used

An Algorithm references an Algorithm Type. Refer to [Setting Up Algorithms](#) for more information.

List of Algorithm Types

The algorithm types available to use with the product may be viewed in the algorithm type page and in the [application viewer's algorithm](#) viewer. If your implementation adds algorithm types or adds algorithms to reference existing algorithm types, you may [regenerate](#) the application viewer to see your additions reflected there.

Setting Up Algorithms

If you need to introduce a new algorithm, open **Admin > System > Algorithm**. Refer to [The Big Picture Of Algorithms](#) for more information.

Description of Page

Enter an easily recognizable **Algorithm Code** and **Description** of the algorithm. **Owner** indicates if this algorithm is owned by the base package or by your implementation (**Customer Modification**).

CAUTION: Important! If you introduce a new algorithm, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

Reference the **Algorithm Type** associated with this algorithm.

FASTPATH: Refer to [Algorithm Type Versus Algorithm](#) for more information about how an algorithm type controls the type of parameters associated with an algorithm.

The parameters available for an algorithm are defined on the algorithm type. The system allows a set of parameter values to change over time. Use the parameter scroll to view parameter values for a given **Effective Date**. The **Owner** of the collection of parameters is displayed. The collection shows the **Parameter** description, the **Sequence** and the **Value** for each parameter.

NOTE: If the product delivers an algorithm with parameter values defined, an implementation may override the base provided parameter values by adding an additional effective dated collection of parameters.

NOTE: If an algorithm is defined and subsequently a new parameter is added to the algorithm type, existing algorithms for the algorithm type should be updated as follows: Click the “+” to add a new effective dated entry to the parameter collection. At this point the latest list of parameters for the algorithm type are visible. Configure the parameters accordingly.

Where Used

Algorithms are plugged in on control tables throughout the system. Each algorithm type’s Algorithm Entity indicates the name of the control table where it is plugged in. The [algorithm](#) viewer in the application viewer may also be used to see a list of plug-in spots along with their algorithm types and algorithms.

Defining Script Options

We use the term "script" to define processing rules that your implementation sets up to control both front-end and back-end processing:

- Rules that control front-end processing are defined using [Business Process Assistant](#) (BPA) scripts. For example, your implementation could set up a BPA script to guide a user through your organization's payment cancellation process.
- Rules that control back-end processing are defined using [Server-based](#) scripts. For example, your implementation could set up a server-based script to control the processing that executes whenever a given type of adjustment is canceled.

The topics in this section describe how to configure your scripts.

The Big Picture Of Scripts

This section describes features and functions that are shared by both BPA scripts and server-based scripts.

Scripts Are Business Process-Oriented

To create a script, you must analyze the steps used to implement a given business process. For example, you could create a "stop auto pay" BPA script that:

- Asks the user to select the customer / taxpayer using an appropriate search page
- Asks the user to define the date on which the person would like to stop making automatic payments
- Invokes a server-based script that populates the end-date on the account's latest automatic payment instructions.

After you understand the business process, you can set up a script to mimic these steps. If the business process is straightforward (e.g., users always perform the same steps), the script configuration will be straightforward. If the business process has several logic branches, the composition of the script may be more challenging.

A Script Is Composed Of Steps

A script contains one or more steps. For example, a "stop auto pay" BPA script might have three steps:

- Ask the user to select the customer / taxpayer using an appropriate search page
- Ask the customer the date on which they'd like to stop making automatic payments (and default the current date)
- Invoke a server-based script that, in turn, updates the account's auto pay options.

Each step references a step type. The step type controls what happens when a step executes. It might be helpful to think of a script as a macro and each step as a "line of code" in the macro. Each step's step type controls the function that is executed when the step is performed.

FASTPATH: Refer to [How To Set Up Each Step Type](#) for a detailed description of all available step types and how to set them up.

Designing And Developing Scripts

Constructing a script is similar to writing a computer program. We recommend that you follow the approach outlined below when you construct scripts:

- Thoroughly understand the business process to be scripted
- Thoroughly understand how the transactions and services that your script uses work
- Design the steps for the script "on paper"
- Determine the most maintainable way to set up your scripts. Rather than creating complex, monolithic scripts, we recommend dividing scripts into smaller sections. For example:
 - For BPA scripts,
 - Determine if several scripts have similar steps. If so, set up a script that contains these common steps and invoke it from the main scripts. For example, if the main script were a BPA script, this would be invoked via a **Perform script** step.
 - Determine if a large script can be divided into logical sections. If so, set up a small script for each section and create a "master script" to invoke each. For example, if the main script were a BPA script, this would be invoked via a **Transfer control** step.
 - For server-based script, you can segregate reusable steps into "service scripts" and then use **Invoke service script** steps to execute the common logic.
- Add the script using [Script Maintenance](#)
- Thoroughly test the script

A Script May Declare Data Areas

Both BPA and server-based scripts may have one or more [data areas](#):

- If the script contains steps that exchange XML documents, you must declare a data area for each type of XML document. For example, if a BPA script has a step that invokes a service script, the BPA script must declare a data area that holds the XML document that is used to pass information to and from the service script.
- You can use a data area as a more definitive way to declare your temporary storage. For example, you can describe your script's temporary storage variables using a [stand-alone data area](#) schema and associate it with your script.

Various step types involve referencing the script's data areas as well as support the ability to compare and move data to and from field elements residing in the data areas.

An [Edit Data](#) step supports the syntax to dynamically declare data areas as part of the step itself. This technique eliminates the need to statically declare a data area. Refer to [Designing Generic Scripts](#) for an example of when this technique may be useful.

Designing Generic Scripts

Scripts may be designed to encapsulate an overall procedure common across different business objects.

For example, BPA scripts may implement a standard procedure to maintain business entities in which the first step obtains the entity's data, the second step presents its associated UI map to the user for update, and the last step updates the entity. Notice that in this case the only difference from one object to another is the data to capture. Rather than designing a dedicated BPA script with static data areas and invocation steps for each business object, you can design a single generic script that dynamically invokes a business object and its associated UI map.

This functionality is available only within an [Edit Data](#) step. With this technique, the name of the schema-based object is held in a variable or an XPath schema location and is used to both declare and invoke the object.

NOTE: Tips. Refer to the tips context zone associated with the script maintenance page for more information on edit data commands and examples.

Securing Script Execution

The system supports the ability to secure the execution of scripts by associating the script with an Application Service. Refer to [The Big Picture of Application Security](#) for more information. Application security is optional and applies to service scripts and user-invocable BPA scripts only. If a script is not associated with an application service, all users may execute the script. Otherwise, only users that have **Execute** access to the application service may execute the script.

The Big Picture Of BPA Scripts

FASTPATH: Refer to [The Big Picture Of Scripts](#) to better understand the basic concept of scripts.

Users may require instructions in order to perform certain tasks. The business process assistant allows you to set up scripts that step a user through your business processes. For example, you might want to create scripts to help users do the following:

- Add a new person to an existing account
- Set up a customer to pay automatically
- Modify a customer who no longer wants to receive marketing information
- Modify a customer's web password
- Record a trouble order
- Merge two accounts into one account
- Fix a bill with an invalid rate
- ... (the list is only limited by your time and imagination)

Users execute these scripts via the [business process assistant](#) (BPA). Users can also define their favorite BPA scripts in their [user preferences](#). By doing this, a user can execute a script by pressing an accelerator key (Ctrl + Shift + a number).

Don't think of these scripts as merely a training tool. BPA scripts can also reduce the time it takes to perform common tasks. For example, you can write a script that reduces the "number of clicks" required to add a new person to an existing account.

CAUTION: Future upgrade issues. Although we make every effort not to remove fields or tab pages between releases, there may be times when changes made by the base-package will necessitate changes to your scripts. Please refer to the release notes for a list of any removed fields or tab pages.

CAUTION: Scripts are not a substitute for end-user training. Scripts minimize the steps required to perform common tasks. Unusual problems (e.g., a missing meter exchange) may be difficult to script as there are many different ways to resolve such a problem. However, scripts can point a user in the right direction and reduce the need to memorize obscure business processes.

The topics in this section describe background topics relevant to BPA scripts.

How To Invoke Scripts

Refer to [Initiating Scripts](#) for a description of how end-users initiate scripts.

Developing and Debugging Your BPA Scripts

You may find it helpful to categorize the step types into two groups: those that involve some type of activity in the [script area](#), and those that don't. The following step types cause activity in the script area: **Height**, **Display text**, **Prompt user**, **Input data**, **Input Map**, **Set focus to a field**. The rest of the step types are procedural and involve no user interaction. For debugging purposes, you can instruct the system to display text in the script area for the latter step types. Also note, for debugging purposes, you can display an entire data area (or a portion thereof) in the script area by entering `%+...+%` where ... is the name of the node whose element(s) should be displayed.

NOTE: Time saver. When you develop a new BPA script, change your [user preferences](#) to include the script as your first "favorite". This way, you can press `Ctrl+Shift+1` to invoke the script (eliminating the need to select it from the [script menu](#)).

Launching A Script From A Menu

You can create menu items that launch BPA scripts rather than open a page. To do this, create a [navigation option](#) that references your script and then add a menu item that references the navigation option.

If the navigation option is referenced on a [context menu](#) and the navigation option has a "context field", a temporary storage variable will be created and populated with the unique identifier of the object in context. For example, if you add a "script" navigation option to the bill context menu and this navigation option has a context field of `BILL_ID`, the system will create a temporary storage variable called `BILL_ID` and populate it with the respective bill id when the menu item is selected.

Launching A Script When Starting The System

You can set the system to launch a script upon startup.

For example, imagine that through an interactive voice response system, a customer has keyed in their account ID and has indicated that they would like to stop an automatic payment. At the point when the IVR system determines that the customer must speak to a user, the interface can be configured to launch the application. When launched it passes the script name and account ID. It can also pass a [navigation option](#) to automatically load the appropriate page (if this information is not part of the script).

To do this, parameters are appended to the standard system URL. The following parameters may be supplied:

- script=<scriptname>
- ACCT_ID=<account id>
- location=<navigation option>

Parameters are added to the standard system URL by appending a question mark (?) to the end and then adding the "key=value" pair. If you require more than one parameter, use an ampersand (&) to separate each key=value pair.

For example, the following URLs are possible ways to launch the **StopAutoPay** script at startup, assuming your standard URL for launching the system is http://system-server:1234/cis.jsp:

- http://system-server:1234/cis.jsp?script=StopAutoPay
- http://system-server:1234/cis.jsp?script=StopAutoPay&ACCT_ID=1234512345
- http://system-server:1234/cis.jsp?script=StopAutoPay&ACCT_ID=1234512345&location=accountMaint

It doesn't matter in which order the parameters are provided. The system processes them in the correct order. For example, the following examples are processed by the system in the same way:

- http://system-server:1234/cis.jsp?ACCT_ID=1234512345&script=StopAutoPay&location=accountMaint
- http://system-server:1234/cis.jsp?ACCT_ID=1234512345&location=accountMaint&script=StopAutoPay

These parameters are kept in a common area accessible by any script for the duration of the session. To use these parameters on a script you may reference the corresponding **%PARM-<name>** [global variables](#). In this example, after the system is launched any script may have access to the above account ID parameter value by means of the **%PARM-ACCT_ID** global variable. Also note, these parameters are also loaded into temporary storage (to continue the example, there'd also be a temporary storage variable called **ACCT_ID** that holds the passed value).

Executing A Script When A To Do Entry Is Selected

The system creates [To Do entries](#) to highlight tasks that require attention (e.g., records in error). Users can complete many of these tasks without assistance. However, you can set up the system to automatically launch a script when a user selects a To Do entry. For example, consider a To Do entry that highlights a bill that's in error due to an invalid mailing address. You can set up the system to execute a script when this To Do entry is selected by a user. This script might prompt the user to first correct the customer's default mailing address and then re-complete the bill.

The following points provide background information to help you understand how to implement this functionality:

- Every To Do entry references a [To Do type](#) and a [message category and number](#). The To Do type defines the category of the task (e.g., bill errors). The message number defines the specific issue (e.g., a valid address can't be found.). Refer to [The Big Picture of System Messages](#) for more information about message categories and numbers.
- When a user drills down on a To Do entry, either a script launches OR the user is transferred to the transaction associated with the entry's To Do type. You control what happens by configuring the To Do type accordingly:
 - If you want to launch a script when a user drills down on an entry, you link the script to the [To Do type and message number](#). Keep in mind that you can define a different script for every message (and some To Do types have many different messages).
 - If the system doesn't find a script for an entry's To Do type and message number, it transfers the user to the To Do type's default transaction.

NOTE: How do you find the message numbers? We do not supply documentation of every To Do type's message numbers (this list would be impossible to maintain and therefore untrustworthy). The best way to determine which message numbers warrant a script is during pre-production when you're testing the system. During this period, To Do entries will be generated. For those entries that warrant a script, simply display the entry on [To Do maintenance](#). On this page, you will find the entry's message number adjacent to the description.

- These types of scripts invariably need to access data that resides on the selected To Do entry. Refer to [How To Use To Do Fields](#) for the details.

The Big Picture Of Script Eligibility Rules

You can configure [eligibility criteria](#) on the scripts to limit the scripts that a user sees in the [script search](#). For example, you could indicate a script should only appear on the script menu if the user belongs to the level 1 customer service representative group. You may also indicate that a script should only appear if the data a user is viewing has certain criteria. For example, if you are using Oracle Utilities Customer Care and Billing, you can indicate that a script should only appear if the current account's customer class is residential. By doing this, you avoid presenting the user with scripts that aren't applicable to the current data in context or the user's role.

The topics in this section describe eligibility rules.

Script Eligibility Rules Are Not Strictly Enforced

The [script search](#) gives a user a choice of seeing all scripts or only scripts that are eligible (given the current data in context and their user profile). This means that it's possible for a script that isn't eligible for the given context data / user to be executed via this search. In other words, the system does not strictly enforce a script's eligibility rules.

It might be more helpful to think of eligibility rules as "highlight conditions". These "highlight conditions" simply control whether the script appears in the [script search](#) when a user indicates they only want to see eligible scripts.

You Can Mark A Script As Always Eligible

If you don't want to configure eligibility rules, you don't have to. Simply indicate that the script is always eligible.

You Can Mark A Script As Never Eligible

If you have scripts that you do not want a user to select from the script menu, indicate that it is never eligible. An example of a script that you wouldn't want a user to select from the menu is one that is [launched when a To Do entry is selected](#). These types of scripts most likely rely on data linked to the selected To Do entry. As a result, a user should only launch scripts of this type from the To Do entry and not from the script menu.

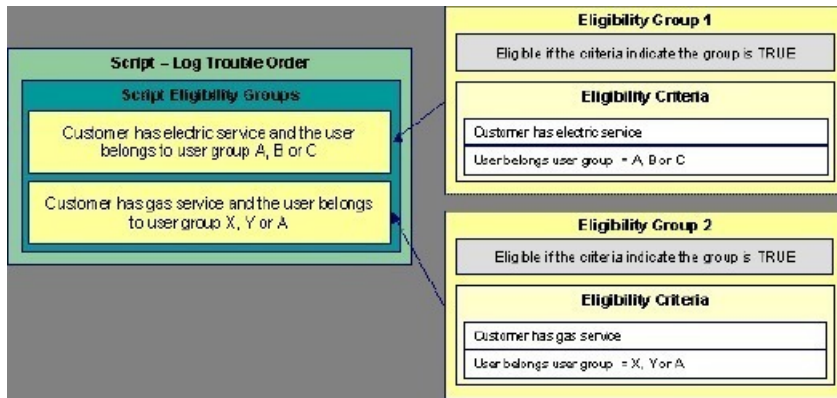
Criteria Groups versus Eligibility Criteria

Before we provide concrete examples of eligibility criteria, we need to explain two concepts: Criteria Groups and Eligibility Criteria. A script's criteria groups control whether a user is eligible to choose a script. At a high level, it works like this:

- A criteria group has one or more eligibility criteria. A group's criteria control whether the group is considered true or false.
- When you create a group, you define what should happen if the group is true or false. You have the following choices:
 - The user is eligible to choose the script
 - The user is not eligible to choose the script
 - The next group should be checked

We'll use the following example from Oracle Utilities Customer Care and Billing to help illustrate these points. Assume a script is only eligible if:

- The customer has electric service and the user belongs to user group A, B or C
- OR, the customer has gas service and the user belongs to user group X, Y or A



This script requires two eligibility groups because it has two distinct conditions:

- IF (Customer has electric service AND (User belongs to user group A, B or C))
- IF (Customer has gas service AND (User belongs to user group X, Y or A))

If either condition is true, the script is eligible.

You would need to set the following criteria groups in order to support this requirement:

Group No.	Group Description	If Group is True	If Group is False
1	Customer has electric service and the user belongs to user group A, B or C	Eligible	Check next group
2	Customer has gas service and the user belongs to user group X, Y or A	Eligible	Ineligible

The following criteria are required for each of the above groups:

Group 1: Customer has electric service and the user belongs to user group A, B or C				
Seq	Logical Criteria	If Eligibility Criteria is True	If Eligibility Criteria is False	If Insufficient Data
10	Customer has electric service	Check next condition	Group is false	Group is false
20	User belongs to user group A, B or C	Group is true	Group is false	Group is false

Group 2: Customer has gas service and the user belongs to user group X, Y or A				
Seq	Logical Criteria	If Eligibility Criteria is True	If Eligibility Criteria is False	If Insufficient Data
10	Customer has gas service	Check next condition	Group is false	Group is false
20	User belongs to user group X, Y or A	Group is true	Group is false	Group is false

The next section describes how you might configure the specific logical criteria in each of the groups.

Defining Logical Criteria

When you set up an eligibility criterion, you must define two things:

- The field to be compared
- The comparison method

You have the following choices in respect of identifying the *field to be compared* :

- You can execute an algorithm to retrieve a field value from somewhere else in the system.

- Some products may support choosing a characteristic linked to an appropriate object in the system (such as an account or person).

You have the following choices in respect of identifying the *comparison method*:

- You can choose an operator (e.g., >, <, =, BETWEEN, IN, etc.) and a comparison value.
- You can execute an algorithm that performs the comparison (and returns True, False or Insufficient Data). This is also a very powerful feature, but it's not terribly intuitive. We'll present a few examples later in this section to illustrate the power of this approach.

The [Examples Of Script Eligibility Rules](#) provide examples to help you understand this design.

Examples Of Script Eligibility Rules

The topics in this section provide examples about how to set up script eligibility rules.

A Script With A Time Span Comparison

A script that is only eligible for senior citizens has the following eligibility rules:

- Customer class = Residential
- Birth date equates to that of a senior citizen

These rules require only one eligibility group on the script. It would look as follows:

Group No.	Group Description	If Group is True	If Group is False
1	Residential and Senior Citizen	Eligible	Ineligible

The following criteria will be required for this group:

Group 1: Residential, Calif, Senior					
Seq	Field to Compare	Comparison Method	If True	If False	If Insufficient Data
10	Algorithm: retrieve account's customer class	= R	Check next condition	Group is false	Group is false
30	Person characteristic: Date of Birth	Algorithm: True if senior	Group is true	Group is false	Group is false

The first criterion is easy; it calls an algorithm that retrieves a field on the current account. This value, in turn, is compared to a given value. If the comparison results in a True value, the next condition is checked. If the comparison doesn't result in a True value, the **Group is false** (and, the group indicates that if the group is false, the script isn't eligible). Refer to SECF-ACCTFLD in the product documentation for an example of an algorithm type that retrieves a field value from an account.

The last criterion contains a time span comparison. Time span comparisons are used to compare a date to something. In our example, we have to determine the age of the customer based on their birth date. If the resultant age is > 65, they are considered a senior citizen. To pull this off, you can take advantage of a comparison algorithm supplied with the base script as described below.

- Field to Compare. The person characteristic in which the customer's birth date is held is selected.
- Comparison Method. We chose a comparison algorithm that returns a value of **True** if the related field value (the customer's date of birth) is greater than 65 years (refer to [SECC-TIMESPN](#) for an example of this type of algorithm).

You'll notice that if a value of **True** is returned by the **True if senior** algorithm, the group is true (and we've set up the group to indicate a true group means the script is eligible).

NOTE: The time span algorithm can be used to compare days, weeks, months, etc. Refer to [SECC-TIMESPN](#) for more information about this algorithm.

A Script With Service Type Comparison

Imagine a script that is only eligible if the current customer has gas service and the user belongs to user groups A, B or C. This script would need the following eligibility rules:

- Customer has gas service
- User belongs to user group A, B, or C

These rules require only one eligibility group on the script. It would look as follows:

Group No.	Group Description	If Group is True	If Group is False
1	Has gas service and user is part of user group A, B or C	Eligible	Ineligible

The following criteria are required for this group:

Group 1: Has gas service and user is part of user group A, B, or C					
Seq	Field to Compare	Comparison Method	If True	If False	If Insufficient Data
10	Algorithm: check if customer has gas service	= True	Check next condition	Group is false	Group is false
20	Algorithm: check if user belongs to user group A, B or C	= True	Group is true	Group is false	Group is false

Both criteria are similar - they call an algorithm that performs a logical comparison. These algorithms are a bit counter intuitive (but understanding them provides you with another way to implement complex eligibility criteria):

The first criterion works as follows:

- **Field to Compare.** We chose a "field to compare" algorithm that checks if the current account has service agreements that belong to a given set of service types. It returns a value of **True** if the customer has an active service agreement that matches one of the service types in the algorithm. In our example, the "check if customer has gas service" algorithm returns a value of **True** if the customer has at least one active service agreement whose SA type references the gas service type. The "check if customer has electric service" algorithm is almost identical, only the service type differs.
- **Comparison Method.** We simply compare the value returned by the algorithm to True and indicate the appropriate response.

The second criterion works similarly:

- **Field to Compare.** We chose a "field to compare" algorithm that checks if the user belongs to any user group in a set of user groups. It returns a value of **True** if the user belongs to at least one user group defined in parameters of the algorithm. Refer to [SECF-USRNGRP](#) for an example of this type of algorithm.
- **Comparison Method.** We simply compare the value returned by the algorithm to True and indicate the appropriate response.

NOTE: Bottom line. The "field to compare" algorithm isn't actually returning a specific field's value. Rather, it's returning a value of **True** or **False**. This value is in turn, compared by the "comparison method" and the group is set to true, false or check next accordingly.

The Big Picture Of Server-Based Scripts

FASTPATH: Refer to [The Big Picture Of Scripts](#) to better understand the basic concept of scripts.

Server-based scripts allow an implementation to configure backend business processes. The system supports two types of server-based scripts, **Plug-In** scripts and **Service** scripts.

- Plug-in scripts allow an implementation to develop routines that are executed from the system's various plug-in spots without coding. For example, an implementation could configure a plug-in script that is executed every time an adjustment of a given type is frozen.
- Service scripts allow an implementation to develop common routines that are invoked from both front-end and back-end services. For example, an implementation could create a service script that terminates an account's automatic payment preferences. This service script could be invoked from a BPA script initiated by an end-user when a customer asks to stop paying automatically, and it could also be executed from a plug-in script if a customer fails to pay their debt on time.

The topics in this section describe background topics relevant to server-based scripts.

Plug-In Scripts

NOTE: This section assumes you are familiar with the notion of plug-in spots (algorithm entities) and plug-ins. See [The Big Picture Of Algorithms](#) for more information.

Rather than write a java program for a plug-in spot, you can create a plug-in using the scripting "language". In essence, this is the only difference between a program-based plug-in and a script-based one. Obviously, this is a significant difference as it allows you to implement plug-ins without programming (and compilers).

The following topics describe basic concepts related to plug-in scripts.

A Plug-In Script's API

Like program-based plug-ins, plug-in scripts:

- Run on the application server
- Have their API (input / output interface) defined by the plug-in spot (i.e., plug-in scripts don't get to declare their own API)
- Can only be invoked by the "plug-in spot driver"

The best way to understand a plug-in script's API is to use the **View Script Schema** hyperlink to view its parameters data area schema.

```

<schema>
  <parm type="group">
    <soft type="list">
      <value/>
    </soft>
    <hard type="group">
      <action use="input"/>
      <businessObject type="group" use="input">
        <id/>
      </businessObject>
    </hard>
  </parm>
</schema>

```

Notice the two groups: soft and hard. If you are familiar with plug-in spots, you'll recognize these as the classic soft and hard parameters:

- The **soft** parameters are the values of the parameters defined on the algorithm. Notice they are not named - if you want to reference them in your plug-in script, you must do it by position (this is equivalent to what a Java programmer does for plug-ins written in Java).
- The **hard** parameters are controlled by the plug-in spot (i.e., the algorithm entity). Notice that this plug-in spot has a single input parameter called "**businessObject/id**". Also notice the **use=** attribute - this shows that this parameter is input-only (i.e., you can't change it in the plug-in script).

NOTE: XPath. You can click on an element name to see the XPath used to reference the element in your script.

Setting Up Plug-In Scripts

You can write plug-in scripts for all plug-in spots that have been Java-enabled. The following points describe how to implement a plug-in script:

- Compose your plug-in [script](#), associating it with the appropriate algorithm entity (plug-in spot).
- Create a new algorithm type for the respective algorithm entity, referencing your plug-in script as the program to carry out the algorithm type's function. Only plug-in scripts associated with the algorithm entity may be referenced on the algorithm type.
- Set up an algorithm for the respective algorithm type and plug it in where applicable. Refer to [Setting Up Algorithm Types](#) for more information.

NOTE: No plug-in scripts are shipped with the base-package.

Service Scripts

BPA scripts run on the client's browser and guide the end-users through business processes. Service scripts run on the application server and perform server-based processing for [BPA scripts](#), [zones](#) and more. You may want to think of a service script as a common routine that is set up via the scripting (rather than programming).

The following topics describe basic concepts related to service scripts.

A Service Script's API

As with any common routine, a service script must declare its input / output parameters (i.e., its API). A service script's API is defined on its [schema](#).

NOTE: Schema Definition Tips. A context sensitive "Script Tips" zone appears when you open the [Script](#) page to assist you with the schema definition syntax. The zone provides a complete list of the XML nodes and attributes available to you when you construct a schema.

Invoking Service Scripts

Any type of script may [invoke a service script](#):

- A BPA script may invoke a service script to perform server-based processing.
- Plug-in scripts may invoke a service script (like a "common routine").
- A service script may call another service script (like a "common routine").

Map zones may be configured to invoke service scripts to obtain the data to be displayed. Refer to [Map Zones](#) for more information.

[Inbound web services](#) support interaction with service scripts allowing the outside world to interact directly with a service script.

You can also invoke a service script from a Java class.

Debugging Server-Based Scripts

The server can create log entries to help you debug your server scripts. These logs are only created if you do the following:

- Start the system in **?debug=true** mode
- Turn on the **Global debug** checkbox in the upper corner of the browser. When you click this check box, a pop-up appears asking you what you want to trace - make sure to check box that causes script steps to be traced.

The logs contain a great deal of information including the contents of the referenced data area for **Move data**, **Invoke business object**, **Invoke business service** and **Invoke service script** steps.

You can view the contents of the logs by pressing the **Show User Log** button appears at the top of the browser.

Please note that all log entries for your user ID are shown (so don't share user IDs).

Maintaining Scripts

The script maintenance transaction is used to maintain your scripts. The topics in this section describe how to use this transaction.

FASTPATH: Refer to [The Big Picture Of Scripts](#) for more information about scripts.

Script - Main

Use this page to define basic information about a script. Open this page using **Admin > System > Script**.

NOTE: Script Tips. A context sensitive "Script Tips" zone is associated with this page. The zone provides useful tips on various topics related to setting up scripts.

Description of Page

Enter a unique **Script** code and **Description** for the script. Use the **Detailed Description** to describe the purpose of this script in detail. **Owner** indicates if the script is owned by the base package or by your implementation (**Customer Modification**).

CAUTION: Important! If you introduce a new script, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

Script Type indicates if this is a **BPA Script**, **Plug-In Script** or **Service Script**. Refer to [The Big Picture Of BPA Scripts](#) and [The Big Picture Of Server Based Scripts](#) for more information.

Accessibility Option appears only for BPA scripts. Set this value to **Accessible from Script Menu** for any script that may be launched as a stand-alone script. Scripts with this configuration may be linked to a navigation option so that they may be invoked from a menu and may be configured by a user as a favorite script. Set this value to **Not Accessible from Script Menu** for any script that cannot be launched on its own. For example, any script that is invoked as a sub-script from another script should have this setting. In addition, any script that is designed to be launched from within a specific portal where certain data is provided to the script should include this setting.

Enter an **Application Service** if the execution of the script should be secured. The application service should include **Execute** as one of its access modes. Refer to [Securing Script Execution](#) for more information.

Algorithm Entity appears only for [plug-in scripts](#). Use this field to define the [algorithm entity](#) into which this script can be plugged in.

Business Object appears only for business object related plug-in scripts. Enter the [Business Object](#) whose elements are to be referenced by the plug-in script.

Script Engine Version defines the version of the XML Path Language (XPath) to be used for the script. Script engine versions 2 and 3 use the XPath 2 engine supplied by the XQuery team. This is the same engine used inside the Oracle database. The current script engine version 3 is a modified version that offers performance improvements without impacting existing version 2 scripts.

Notes regarding Script engine version 1: The XPath library used is Jaxen; for BPA scripts, use IE's built-in MSXML parser; Xpath 1 (and even JavaScript) uses floating point arithmetic, which means that adding a collection of numbers with two decimal places might end up with a value of 10779.079999999998 instead of 10779.08.

Click the **View Script Schema** to view the [script's data areas](#) on the [schema viewer](#) window.

Click the **View XSD** hyperlink to view a service script's schema definition in XSD format.

The **View Script As Text** hyperlink appears for server-based scripts only. Click this link to view the internal scripting commands in a separate window. The presented script syntax is valid within [edit data](#) steps.

The [tree](#) summarizes the script's steps. You can use the hyperlink to transfer you to the **Step** tab with the corresponding step displayed.

Script - Step

Use this page to add or update a script's steps. Open this page using **Admin > System > Script** and then navigate to the **Step** tab.

NOTE: Time saver. You can navigate to a step by clicking on the respective node in the tree on the Main tab.

Description of Page

The **Steps** accordion contains an entry for every step linked to the script. When a script is initially displayed, its steps are collapsed. To see a step's details, simply click on the step's summary bar. You can re-click the bar to collapse the step's details. Please see [accordions](#) for the details of other features you can use to save time.

Select the **Step Type** that corresponds with the step. Refer to [How To Set Up Each Step Type](#) for an overview of the step types.

CAUTION: The Step Type affects what you can enter on other parts of this page. The remainder of this section is devoted to those fields that can be entered regardless of Step Type. The subtopics that follow describe those fields whose entry is contingent on the Step Type.

StepSequence defines the relative position of this step in respect of the other steps. The position is important because it defines the order in which the step is executed. You should only change a Step Sequence if you need to reposition this step. But take care; if you change the Step Sequence and the step is referenced on other steps, you'll have to change all of the referencing steps.

NOTE: Leave gaps in the sequence numbers. Make sure you leave space between sequence numbers so that you can add new steps between existing ones in the future. If you run out of space, you can use the **Renumber** button to renumber all of the steps. This will renumber the script's steps by 10 and change all related references accordingly.

Display Step is only enabled on BPA scripts for step types that typically don't cause information to be displayed in the [script area](#) (i.e., step types like **Conditional Branch**, **Go to a step**, **Height**, etc). If you turn on this switch, information about the step is displayed in the script area to help you debug the script.

CAUTION: Remember to turn this switch off when you're ready to let users use this script.

NOTE: If **Display Step** is turned on and the step has **Text**, this information will be displayed in the script area. If **Display Step** is turned on and the step does NOT have **Text**, a system-generated messages describing what the step does is displayed in the script area.

Display Icon controls the **icon** that prefixes the **Text** that's displayed in the script area. Using an icon on a step is optional. This field is only applicable to BPA scripts.

Text is the information that displays in the **script area** when the step executes. You need only define text for steps that cause something to display in the script area.

FASTPATH: Refer to [How To Substitute Variables In Text](#) for a discussion about how to substitute variables in a text string.

FASTPATH: Refer to [How To Use HTML Tags And Spans In Text](#) for a discussion about how to format (with colors and fonts) the text that appears in the script area.

The other fields on this page are dependent on the **Step Type**. The topics that follow briefly describe each step type's fields and provide additional information about steps.

Click on the **View Script Schema** hyperlink to view the script's data areas. Doing this opens the **schema viewer** window.

The **View Script As Text** hyperlink appears for server-based scripts only. Click this link to view the internal scripting commands in a separate window. The presented script syntax is valid within **edit data** steps.

How To Set Up Each Step Type

The contents of this section describe how to set up each type of step.

Common Step Types To All Script Types

The contents of this section describe common step types applicable to all script types.

How To Set Up Conditional Branch Steps

Conditional branch steps allow you to conditionally jump to a different step based on logical criteria. For example, you could jump to a different step in a script if the customer is residential as opposed to commercial. In addition, several fields are required for **Conditional Branch** steps:

Compare Field Type and **Compare Field Name** define the first operand in the comparison. The **Field Type** defines where the field is located. The **Field Name** defines the name of the field. The following points describe each field type:

- **Current To Do Information.** Use this field type when the field being compared resides on the current To Do entry. Refer to [How To Use To Do Fields](#) for instructions on how to define the appropriate **Field Name**.
- **Data Area.** Use this field type when the field being compared is one that you put into one of the scripts data areas in an earlier step. **Field Name** must reference both a data area structure name as well as the field, for example "parm/charType". Refer to [How To Reference Fields In Data Areas](#) for instructions on how to construct the appropriate **Field Name**.
- **Page Data Model.** Use this field type when the field being compared resides on one of the tab pages in the **object display area**. Refer to [How To Find The Name Of Page Data Model Fields](#) for instructions on how to find the appropriate **Field Name**.
- **Predefined Value.** Use this field type when the field being compared is a **global variable**.
- **Temporary Storage.** Use this field type when the field being compared is one that you put into temporary storage in an earlier step. The **Field Name** must be the same as defined in an earlier step.

- **User Interface Field.** Use this field type when the field being compared resides on the currently displayed tab page. Refer to [How To Find The Name Of User Interface Fields](#) for instructions on how to find the appropriate **Field Name**.

Condition defines the comparison criteria:

- Use **>**, **<**, **=**, **>=**, **<=**, **<>** (not equal) to compare the field using standard logical operators. Enter the comparison value using the following fields.
- Use **IN** to compare the first field to a list of values. Each value is separated by a comma. For example, if a field value must equal **1, 3 or 9**, you would enter a comparison value of **1,3,9**.
- Use **BETWEEN** to compare the field to a range of values. For example, if a field value must be between **1 and 9**, you would enter a comparison value of **1,9**. Note, the comparison is inclusive of the low and high values.

Comparison Field Type, **Comparison Field Name** and **Comparison Value** define what you're comparing the first operand to. The following points describe each field type:

- **Current To Do Information.** Use this field type when the comparison value resides on the current To Do entry. Refer to [How To Use To Do Fields](#) for instructions on how to define the appropriate **Field Name**.
- **Data Area.** Use this field type when the comparison value resides in one of the scripts data areas. **Field Name** must reference both a data area structure name as well as the field, for example "parm/charType". Refer to [How To Reference Fields In Data Areas](#) for instructions on how to construct the appropriate **Field Name**.
- **Page Data Model.** Use this field type when the comparison value resides on one of the tab pages in the [object display area](#). Refer to [How To Find The Name Of Page Data Model Fields](#) for instructions on how to find the appropriate **Field Name**.
- **Predefined Value.** Use this field type when the field being compared is a constant value defined in the script. When this field type is used, use **Comparison Value** to define the constant value. Refer to [How To Use Constants In Scripts](#) for instructions on how to use constants.
- **Temporary Storage.** Use this field type when the comparison value is a field that you put into temporary storage in an earlier step. The **Field Name** must be the same as defined in an earlier step.
- **User Interface Field.** Use this field type when the comparison value resides on the currently displayed tab page. Refer to [How To Find The Name Of User Interface Fields](#) for instructions on how to find the appropriate **Field Name**.

NOTE: Conditional field types. The field types **Current To Do Information**, **Page Data Model** and **User Interface Field** are only applicable to BPA scripts.

The above fields allow you to perform a comparison that results in a value of **TRUE** or **FALSE**. The remaining fields control the step to which control is passed given the value:

- **If TRUE, Go to** defines the step that is executed if the comparison results in a **TRUE** value.
- **If FALSE, Go to** defines the step that is executed if the comparison results in a **FALSE** value.

NOTE: Numeric Comparison. Comparison of two values may be numeric or textual (left-to-right). Numeric comparison takes place only when values on both side of the comparison are recognized as numeric by the system. Otherwise, textual comparison is used. Fields for **Current To Do Information**, **Data Area**, **Page Data Model**, and **User Interface Field** types are explicitly associated with a data type and therefore can be recognized as numeric or not. This is not the case for fields residing in **Temporary Storage** or those set as **Predefined Values**. A **Temporary Storage** field is considered numeric if it either holds a numeric value moved to it from an explicitly defined numeric value (see above) or it is a resultant field of mathematical operation. A **Predefined Value** field is considered numeric if the other field it is compared to is numeric. For example, if a numeric field is compared to a **Predefined Value** the latter is considered numeric as well resulting in numeric value comparison. However, if the two fields are defined as **Predefined Values** the system assumes their values are text strings and therefore applies textual comparison.

How To Set Up Edit Data Steps

Edit data steps provide a free format region where you can specify commands to control your script processing.

In general, the syntax available within edit data mimics the commands available within the explicit step types. However, there are a few commands that are available only within edit data. For example, the two structured commands: **For**, and **If**. For server-based scripts, you may find it useful to create a few explicit step types and then use the **View Script as Text** hyperlink on the [Script - Step](#) page to better understand the edit data syntax.

NOTE: Not all BPA step types are supported using the edit data syntax. **Tips.** Refer to the tips context zone associated with the script maintenance page for more information on edit data commands and examples.

Additional field required for **Edit data** steps:

Enter your scripting commands in the **Edit Data Text** field.

How To Set Up Go To Steps

Go to steps allow you to jump to a step other than the next step. Additional fields required for **Go To** steps:

Next Step defines the step to which the script should jump.

How To Set Up Invoke Business Object Steps

Invoke business object steps allow you to interact with a [business object](#) in order to obtain or maintain its information.

The following additional fields are required for **Invoke business object** steps:

Use **Warning Level** to indicate whether warnings should be suppressed and if not, how they should be presented to the user. By default, warnings are suppressed. If **Warn As Popup** is used, the warning is displayed using the standard popup dialog. If **Warn As Error** is used processing is directed to the **If Error, Go To** step. This field is only applicable to BPA scripts.

Group Name references the [data area](#) to be passed to and from the server when communicating with the **Business Object**. Indicate the **Action** to be performed on the object when invoked. Valid values are **Add, Delete, Fast Add (No Read), Fast Update (No Read), Read, Replace, Update**.

NOTE: Performance note. The actions **Fast Add** and **Fast Update** should be used when the business object's data area does not need to be re-read subsequent to the **Add** or **Update** action. In other words, the **Add** and **Update** actions are equivalent to **Fast Add + Read** and **Fast Update + Read**.

The business object call will either be successful or return an error. The next two fields only appear when the call is issued from a BPA script, to determine the step to which control is passed given the outcome of the call.

If Success, Go To defines the step that is executed if the call is successful. This field is only applicable to BPA scripts.

If Error, Go To defines the step that is executed if the call returns on error. Please note that the error information is held in [global variables](#). This field is only applicable to BPA scripts.

NOTE: Error technique. Let's assume a scenario where a business object is invoked from a BPA script and the call returned an error. If the BPA script is configured to communicate with the user using a UI map, you may find it useful to invoke the map again to present the error to the user. Alternatively, you may invoke a step that transfers control to a script that displays the error message information and stops.

How To Set Up Invoke Business Service Steps

Invoke business service steps allow you to interact with a [business service](#).

The following additional fields are required for **Invoke business service** steps:

Use **Warning Level** to indicate whether warnings should be suppressed and if not, how they should be presented to the user. By default, warnings are suppressed. If **Warn As Popup** is used, the warning is displayed using the standard popup dialog. If **Warn As Error** is used processing is directed to the **If Error, Go To** step. This field is only applicable to BPA scripts.

Group Name references the [data area](#) to be passed to and from the server when the **Business Service** is invoked.

The business service call will either be successful or return an error. The next two fields only appear when the call is issued from a BPA script, to determine the step to which control is passed given the outcome of the call.

If Success, Go To defines the step that is executed if the call is successful. This field is only applicable to BPA scripts.

If Error, Go To defines the step that is executed if the call returns on error. Please note that the error information is held in [global variables](#). This field is only applicable to BPA scripts.

NOTE: Error technique. Let's assume a scenario where a business service is invoked from a BPA script and the call returned an error. If the BPA script is configured to communicate with the user using a UI map, you may find it useful to invoke the map again to present the error to the user. Alternatively, you may invoke a step that transfers control to a script that displays the error message information and stops.

How To Set Up Invoke Service Script Steps

Invoke service script steps allow you to execute a [service script](#).

The following additional fields are required for **Invoke service script** steps:

Use **Warning Level** to indicate whether warnings should be suppressed and if not, how they should be presented to the user. By default, warnings are suppressed. If **Warn As Popup** is used, the warning is displayed using the standard popup dialog. If **Warn As Error** is used processing is directed to the **If Error, Go To** step. This field is only applicable to BPA scripts.

Group Name references the [data area](#) to be passed to and from the server when the **Service Script** is invoked.

The service script call will either be successful or return an error. The next two fields only appear when the call is issued from a BPA script to determine the step to which control is passed given the outcome of the call.

If Success, Go To defines the step that is executed if the call is successful. This field is only applicable to BPA scripts.

If Error, Go To defines the step that is executed if the call returns on error. Please note that the error information is held in [global variables](#). This field is only applicable to BPA scripts.

NOTE: Error technique. Let's assume a scenario where a service script is invoked from a BPA script and the call returned an error. If the BPA script is configured to communicate with the user using a UI map, you may find it useful to invoke the map again to present the error to the user. Alternatively, you may invoke a step that transfers control to a script that displays the error message information and stops.

How To Set Up Label Steps

Label steps allow you to describe what the next step(s) are doing. Steps of this type are helpful to the script administrators when reviewing or modifying the steps in a script, especially when a script has many steps. When designing a script, the label steps enable you to provide a heading for common steps that belong together. The script tree displays steps of this type in a different color (green) so that they stand out from other steps.

There are no additional fields for **Label** steps.

How To Set Up Move Data Steps

Move data steps allow you to move data (from a source to a destination). The following additional fields are required for **Move data** steps:

Source Field Type, **Source Field Name** and **Source Field Value** define what you're moving. The following points describe each field type:

- **Context Variable.** Use this field type in a plug-in or service script if the source value is a variable initiated in a higher level script.
- **Current To Do Information.** Use this field type when the source value resides on the current To Do entry. Refer to [How To Use To Do Fields](#) for instructions on how to define the appropriate **Field Name**.
- **Data Area.** Use this field type when the field being compared is one that you put into one of the script's data areas in an earlier step. **Field Name** must reference both a data area structure name as well as the field, for example "parm/

charType". Refer to [How To Reference Fields In Data Areas](#) for instructions on how to construct the appropriate **Field Name**.

- **Global Context.** Use this field type in a BPA script when the source value is a global variable.
- **Page Data Model.** Use this field type in a BPA script when the source value resides on any of the tab pages in the [object display area](#) (i.e., the source field doesn't have to reside on the currently displayed tab page, it just has to be part of the object that's currently displayed). Refer to [How To Find The Name Of Page Data Model Fields](#) for instructions on how to find the appropriate **Field Name**.
- **Portal Context.** Use this field type when the source value is a variable in the portal context.
- **Predefined Value.** Use this field type when the source value is a constant value defined in the script. When this field type is used, use **Source Field Value** to define the constant value. Refer to [How To Use Constants In Scripts](#) for instructions on how to use constants.

NOTE: Concatenating fields together. You can also use **Predefined Value** if you want to concatenate two fields together. For example, let's say you have a script that merges two persons into a single person. You might want this script to change the name of the person being merged out of existence to include the ID of the person remaining. In this example, you could enter a **Source Field Value** of **%ONAMEmerged into person %PERID** (where **ONAME** is a field in temporary storage that contains the name of the person being merged out of existence and **PERID** contains the ID of the person being kept). Refer to [How To Substitute Variables In Text](#) for a description of how you can substitute field values to compose the field value.

- **Temporary Storage.** Use this field type when the source value is a field that you put into temporary storage in an earlier step. The **Field Name** must be the same as defined in an earlier step.
- **User Interface Field.** Use this field type when the source value resides on the currently displayed tab page. Refer to [How To Find The Name Of User Interface Fields](#) for instructions on how to find the appropriate **Field Name**.

Destination Field Type and **Destination Field Name** define where the source field will be moved. The **Field Type** defines where the field is located. The **Field Name** defines the name of the field. The following points describe each field type:

- **Context Variable.** Use this field type in your plug-in or service script if you use a variable to communicate information to a lower level service script or schema.
- **Data Area.** Use this field type when the destination field resides on one of the scripts data areas. **Field Name** must reference both a data area structure name as well as the field, for example "parm/charType". Refer to [How To Reference Fields In Data Areas](#) for instructions on how to construct the appropriate **Field Name**.
- **Page Data Model.** Use this field type when the destination field resides on any of the tab pages in the [object display area](#) (i.e., the field populated doesn't have to reside on the currently displayed tab page, it just has to be part of the object that's currently displayed). Refer to [How To Find The Name Of Page Data Model Fields](#) for instructions on how to find the appropriate **Field Name**.
- **Portal Context.** Use this field type in a BPA script when the destination to be updated is in the current portal context.
- **Temporary Storage.** Use this field type when the destination field resides in temporary storage. Use **Field Name** to name the field in temporary storage. Use **Field Name** to name the field in temporary storage. Refer to [How To Name Temporary Storage Fields](#) for more information.
- **User Interface Field.** Use this field type when the destination field resides on the currently displayed tab page. Refer to [How To Find The Name Of User Interface Fields](#) for instructions on how to find the appropriate **Field Name**.

NOTE: Conditional field types. The field types **Current To Do Information**, **Page Data Model** and **User Interface Field** are only applicable to BPA scripts.

How To Set Up Terminate Steps

Terminate steps cause a server-based script to end processing successfully or issue an error.

The following additional fields are required for **Terminate** steps:

Error indicates whether an error should be thrown or not. If error, **Error Data Text** must be specified, indicating the error message and any message substitution parameters. Refer to the tips zone associated with the Script page for the actual syntax of initiating an error message.

NOTE: The ability to terminate a step in error is only supported for server-based scripts.

Step Types Applicable to BPA Scripts only

The contents of this section describe step types that are only applicable to BPA scripts.

How To Set Up Display Text Steps

Display text steps cause a text string to be displayed in the script area. Steps of this type can be used to provide the user with guidance when manual actions are necessary. In addition, they can be used to provide confirmation of the completion of tasks.

The information you enter in the **Text** field is displayed in the [script area](#) when the step is executed.

The text string can contain [substitution variables](#) and [HTML formatting commands](#). Also note that for debugging purposes, you can display an entire data area (or a portion thereof) by entering `%+...+%` where `...` is the name of the node whose element(s) should be displayed.

NOTE: Conditional step type. This step type is only applicable to BPA scripts.

How To Set Up Height Steps

Height steps are used to change the height of the script area to be larger or smaller than the standard size.

The following additional fields are required for **Height** steps:

Script Window Height defines the number of **Pixels** or the **Percentage** (according to the **Height Unit**) that the script window height should be adjusted. The percentage indicates the percentage of the visible screen area that the script area uses. For example, a percentage value of **100** means that the script area will use the entire area.

NOTE: Standard Number of Pixels. The default number of pixels used by the script area is **75**.

NOTE: Adjust script height in the first step. If you want to adjust the height of the script area, it is recommendation to define the **height** step type as your first step. Otherwise, the script area will open using the standard height and then readjust, causing the screen to redisplay.

NOTE: Hide script area. You could use this type of step to set the height to **0** to hide the script area altogether. This is useful if the script does not require any prompting to the user. For example, perhaps you define a script to take a user to a page and with certain data pre-populated and that is all.

NOTE: Automatically close script area. If you want the script area to close when a script is completed, you could define the final step type with a height of 0.

NOTE: Conditional step type. This step type is only applicable to BPA scripts.

How To Set Up Input Data Steps

Input data steps cause the user to be prompted to populate an input field in the script area. The input value can be saved in a field on a page or in temporary storage. A **Continue** button always appears adjacent to the input field. You may configure steps of this type to display one or more buttons in addition to the **Continue** button. For example, you may want to provide the ability for the user to return to a previous step to fix incorrect information. The user may click on any of these buttons when ready for the script to continue.

The following additional fields are required for **Input Data** steps:

Destination Field Type and **Destination Field Name** define where the input field will be saved. The **Field Type** defines where the field is located. The **Field Name** defines the name of the field. The following points describe each field type:

- **Page Data Model.** Use this field type to put the input field into a field that resides on any of the tab pages in the [object display area](#) (i.e., the field populated doesn't have to reside on the currently displayed tab page, it just has to be part of the object that's currently displayed). Refer to [How To Find The Name Of Page Data Model Fields](#) for instructions on how to find the appropriate **Field Name**.
- **Temporary Storage.** Use this field type to put the input field into temporary storage. Use **Field Name** to name the field in temporary storage. Refer to [How To Name Temporary Storage Fields](#) for more information.
- **User Interface Field.** Use this field type to put the input field into a field that resides on the currently displayed tab page. Note, if you want to execute underlying default logic, you must populate a **User Interface Field**. Refer to [How To Find The Name Of User Interface Fields](#) for instructions on how to find the appropriate **Field Name**.

The **Prompt Values** grid may be used to define additional buttons. A separate button is displayed in the script area for each entry in this grid.

- **Prompt Text** is the verbiage to appear on the button. Refer to [How To Substitute Variables In Text](#) for a description of how you can substitute field values into the prompts.
- **Sequence** controls the order of the buttons.
- **Next Script Step** defines the step to execute if the user clicks the button.

NOTE: Conditional step type. This step type is only applicable to BPA scripts.

How To Set Up Invoke Function Steps

NOTE: Functions were implemented prior to the introduction of business services (BS), service scripts (SS) and business objects (BO). The functionality is still supported, but the recommendation for implementations going forward is to use a step that invokes one of the above configuration tool objects in a script rather than defining a function.

Invoke function steps may be used to retrieve or update data independent of the page currently being displayed. For example, if you design a script that takes different paths based on the customer's customer class, you could invoke a function to retrieve the customer's customer class.

FASTPATH: You must set up a function before it can be referenced in a script. Refer to [Maintaining Functions](#) for the details.

The following additional fields are required for **Invoke Function** steps:

Function defines the name of the function. The function's **Long Description** is displayed below.

When a function is invoked, it will either be successful or return an error. The next two fields control the step to which control is passed given the outcome of the function call:

- **If Success, Go to** defines the step that is executed if the function is successful.
- **If Error, Go to** defines the step that is executed if the function returns on error. Refer to [How To Use Constants In Scripts](#) for a list of the global variables that are populated when a function returns an error.

NOTE: Error technique. If a function returns an error, we recommend that you invoke a step that transfers control to a script that displays the error message information and stops (note, the error information is held in [global variables](#)). You would invoke this script via a **Transfer Control**.

The **Send Fields** grid defines the fields whose values are sent to the function and whose field value source is not **Defined On The Function**. For example, if the function receives an account ID, you must define the name of the field in the script that holds the account ID.

- **Field** contains a brief description of the field sent to the function.

- **Source Field Type** and **Mapped Field / Value** define the field sent to the function. Refer to the description of Source Field under [How To Set Up Move Data Steps](#) for a description of each field type.
- **Comments** contain information about the field (this is defined on the function).

The **Receive Fields** grid defines the fields that hold the values returned from the function. For example, if the function returns an account's customer class and credit rating, you must set up two fields in this grid.

- **Field** contains a brief description of the field returned from the function.
- **Destination Field Type** and **Mapped Field** define the field returned from the function. Refer to the description of Destination Field under [How To Set Up Move Data Steps](#) for a description of each field type.
- **Comments** contain information about how the field (this is defined on the function).

NOTE: Conditional step type. This step type is only applicable to BPA scripts.

How To Set Up Invoke Map Steps

Invoke map steps are used to invoke a [UI Map](#) to display, capture and update data using an HTML form. You may configure steps of this type to display one or more buttons in addition to the **Continue** button. For example, you may want to provide the ability for the user to return to a previous step to fix incorrect information. The user may click on any of these buttons when ready for the script to continue.

The following additional fields are required for **Invoke map** steps:

Group Name references the [data area](#) to be passed to and from the server when rendering the HTML form associated with the **Map**.

Use **Target Area** to designate where the map will be presented.

- Select **BPA Zone** if the map should be presented within the [script area](#).
- Select **Page Area** if the map should be presented in the [object display area](#), i.e. the frame typically used to house a maintenance page.
- Select **Pop-up Window** if the map should be launched in a separate window.

The **Returned Values** grid contains a row for every button defined on the map.

- **Returned Value** is the value returned when the user clicks the button.
- **Use as Default** can only be turned on for one entry in the grid. If this is turned on, this value's Next Script Step will be executed if the returned value does not match any other entry in the grid. For example, if the user closes a pop-up (rather than clicking a button), the default value will be used.
- **Next Script Step** defines the step to execute if the user clicks the button.

NOTE: Conditional step type. This step type is only applicable to BPA scripts.

How To Set Up Mathematical Operation Steps

Mathematical operation steps allow you to perform arithmetic on fields. You can also use this type of step to add and subtract days from dates. For example, you could calculate a date 7 days in the future and then use this value as the customer's next credit review date. The following additional fields are required for **Mathematical Operation** steps:

Base Field Type and **Base Field Name** define the field on which the mathematical operation will be performed. The **Field Type** defines where the field is located. The **Field Name** defines the name of the field. The following points describe each field type:

- **Page Data Model.** Use this field type when the field resides on any of the tab pages in the [object display area](#). Refer to [How To Find The Name Of Page Data Model Fields](#) for instructions on how to find the appropriate **Field Name**.
- **Temporary Storage.** Use this field type when the field resides in temporary storage. You must initialize the temporary storage field with a Move Data step before performing mathematical operations on the field. Refer to [How To Set Up Move Data Steps](#) for more information.

- **User Interface Field.** Use this field type when the field resides on the currently displayed tab page. Refer to [How To Find The Name Of User Interface Fields](#) for instructions on how to find the appropriate **Field Name**.

Math Operation controls the math function to be applied to the **Base Field**. You can specify +, -, /, and *. Note, if the base field is a date, you can only use + or -.

Math Field Type, **Math Field Name** and **Math Field Value** define the field that contains the value to be added, subtracted, divided, or multiplied. The following points describe each field type:

- **Current To Do Information.** Use this field type when the value resides on the current To Do entry. Refer to [How To Use To Do Fields](#) for instructions on how to define the appropriate **Field Name**.
- **Page Data Model.** Use this field type when the value resides on any of the tab pages in the [object display area](#). Refer to [How To Find The Name Of Page Data Model Fields](#) for instructions on how to find the appropriate **Field Name**.
- **Predefined Value.** Use this field type when the value is a constant. When this field type is used, use **Source Field Value** to define the constant value. Refer to [How To Use Constants In Scripts](#) for more information. Note, if you are performing arithmetic on a date, the field value must contain the number and type of **days/ months/ years**. For example, if you want to add 2 years to a date, the source field value would be **2 years**.
- **Temporary Storage.** Use this field type when the value is a field that you put into temporary storage in an earlier step. The **Field Name** must be the same as defined in an earlier step.
- **User Interface Field.** Use this field type when the value resides in a field on the current tab page. Refer to [How To Find The Name Of User Interface Fields](#) for instructions on how to find the appropriate **Field Name**.

NOTE: Conditional step type. This step type is only applicable to BPA scripts.

How To Set Up Navigate To A Page Steps

Navigate to a page steps cause a new page (or tab within the existing page) to be displayed in the object display area. Steps of this type are a precursor to doing anything on the page. The following additional field is required for **Navigate to a page** steps:

Navigation Option defines the transaction, tab, access mode (add or change) and any context fields that are passed to the transaction in change mode. For example, if you want a script to navigate to Person - Characteristics for the current person being displayed in the dashboard, you must set up an appropriate navigation option. Refer to [Defining Navigation Options](#) for more information.

NOTE: Navigating to a page in update mode. Before you can navigate to a page in change mode, the page data model must contain the values to use for the navigation option's context fields. If necessary, you can move values into the page data model using a [Move Data step](#) first. For example, before you can navigate to a page in change mode with an account ID in context, you may need to move the desired account ID into the ACCT_ID field in the page data model. The actual field name(s) to use are listed as context fields on the [navigation option](#).

NOTE: Conditional step type. This step type is only applicable to BPA scripts.

How To Set Up Perform Script Steps

Perform script steps cause another BPA script to be performed. After the performed script completes, control is returned to the next step in the original script. You might want to think of the scripts referred to on steps of this type as "subroutines". This functionality allows you to encapsulate common logic in reusable BPA scripts that can be called from other BPA scripts. This simplifies maintenance over the long term.

The following additional field is required for **Perform script** steps:

Subscript is the name of the script that is performed.

NOTE: Conditional step type. This step type is only applicable to BPA scripts.

How To Set Up Press A Button Steps

Press a button steps cause a button or link text to be ‘pressed’ in the [object display area](#), the [application toolbar](#) or the [page title area](#). For example, you could use this type of step to add a new row to a person's characteristic (and then you could use a **Move Data** step to populate the newly added row with a given char type and value). The following additional fields are required for **Press a button** steps:

Button Name is the name of the button to be pressed. This button must reside on the currently displayed tab page (or in the application toolbar or page actions toolbar). Refer to [How To Find The Name Of A Button](#) for more information.

NOTE: Conditional step type. This step type is only applicable to BPA scripts.

How To Set Up Prompt User Steps

Prompt user steps cause the user to be presented with a menu of options. The options can be presented using either buttons or in the contents of a drop down. You can also use steps of this type to pause a script while the user checks something out (and when the user is ready to continue with the script, they are instructed to click a prompt button). The following additional fields are required for **Prompt User** steps:

Prompt Type controls if the prompt shown in the script area is in the form of **Button(s)** or a **Dropdown**. Note, if you use a **Dropdown**, a Continue button appears adjacent to the dropdown in the script area when the step executes. The user clicks the Continue button when they are ready for the script to continue.

The **Prompt Values** grid contains a row for every value that can be selected by a user. Note, if you use a **Prompt Type** of **Button(s)**, a separate button is displayed in the script area for each entry in this grid.

- **Prompt Text** is the verbiage to appear on the button or in the dropdown entry. Refer to [How To Substitute Variables In Text](#) for a description of how you can substitute field values into the prompts.
- **Sequence** controls the order of the buttons or dropdown entries.
- **Use As Default** can only be turned on for one entry in the grid. If this is turned on for a dropdown entry, this value is defaulted in the grid. If this is turned on for a button, this button becomes the default (and the user should just have to press **Enter** (or **space**) rather than click on it).
- **Next Script Step** defines the step to execute if the user clicks the button or selects the dropdown value.

NOTE: Conditional step type. This step type is only applicable to BPA scripts.

How To Set Up Set Focus To A Field Steps

Set focus to a field steps cause the cursor to be placed in a specific field on a page. A **Continue** button always appears in the script area when this type of step executes. The user may click the **Continue** button when they are ready for the script to continue. You may configure steps of this type to display one or more buttons in addition to the **Continue** button. For example, you may want to provide the ability for the user to return to a previous step to fix incorrect information. The user may click on any of these buttons when ready for the script to continue.

The following additional fields are required for **Set focus to a field** steps:

Destination Field Name defines the field on which focus should be placed. This field must reside on the currently displayed tab page. Refer to [How To Find The Name Of User Interface Fields](#) for instructions on how to find the appropriate **Field Name**.

The **Prompt Values** grid may be used to define additional buttons. A separate button is displayed in the script area for each entry in this grid.

- **Prompt Text** is the verbiage to appear on the button. Refer to [How To Substitute Variables In Text](#) for a description of how you can substitute field values into the prompts.
- **Sequence** controls the order of the buttons.
- **Next Script Step** defines the step to execute if the user clicks the button.

NOTE: Conditional step type. This step type is only applicable to BPA scripts.

How To Set Up Transfer Control Steps

Transfer control steps cause the current BPA script to terminate and the control to pass to another BPA script. You might want to construct a BPA script with steps of this type when the script has several potential logic paths and you want to segregate each logic path into a separate BPA script (for ease of maintenance).

The following additional fields are required for **Transfer control** steps:

Subscript is the name of the script to which control is transferred.

NOTE: Conditional step type. This step type is only applicable to BPA scripts.

Additional Topics

The contents of this section provide additional information about steps.

How To Find The Name Of User Interface Fields

Follow these steps to find the name of a field that resides on a page:

- Navigate to the page in question.
- Right click in the body of the page (but not while the pointer is in an input field). Note, if the field in question resides in a grid, you must right click while the pointer is in the section that contains the grid (but not while the pointer is in an input field in the grid) - this is because there's a separate HTML document for each grid on a page.
- Select **View Source** from the pop-up menu to display the source HTML.
- Scroll to the Widget Info section (towards the top of the HTML document). It contains a list of all of the objects on a page. For example, the following is an example from the Account - Main page:

```
widget info:
widget_ID , Element Type - label info - label
ENTITY_NAME, IL - $ENTITY_NAME - Name
ACCT_ID, IT - $ACCT_ID - Account ID
ACCT_CHECK_DIGIT, IL - $ACCT_CHECK_DIGIT - Account Check Digit
IM_ACCT_ID, IM - $SEARCH_FOR_ACC_LBL - Search for Account
COLL_CL_CD, HD - $COLL_CL_CD - Collection Class
SETUP_DT, IT - $SETUP_DT - Set Up Date
CURRENCY_CD, IS - $CURRENCY_CD - Currency Code
CIS_DIVISION, IS - $CIS_DIVISION - CIS Division
PROTECT_DIV_SW, CB - $CI_ACCT$PROTECT_DIV_SW - Protect CIS Division
CUST_CL_CD, IS - $CUST_CL_CD - Customer Class
ACCESS_GRP_CD, IT - $ACCESS_GRP_CD - Access Group
IM_ACCESS_GRP_CD, IM - $SRCH_ACC_GRP_CD - Search for Access Group
ACCESS_DESCR, IL - $DESCR - Description
ACCT_MGMT_GRP_CD, IT - $ACCT_MGMT_GRP_CD - Account Management Group
IM_ACCT_MGMT_GRP_CD, IM - $SEARCH_FOR_TDR_LBL - Search for To Do Role
MGMT_DESCR, IL
ALERT_INFO, IA - $ALERT_INFO - Alert Information
BILL_CYC_CD, IS - $BILL_CYC_CD - Bill Cycle
BILL_AFTER_DT, IT - $BILL_AFTER_DT - Bill After
PROTECT_CYC_SW, CB - $PROTECT_CYC_SW - Protect Bill Cycle
BILL_PRT_INTERCEPT, IT - $BILL_PRT_INTERCEPT - Bill Print Intercept
IM_BILL_PRT_INTERCEPT, IM - $FOR_BILL_PRINT_LBL - Search for User
MAILING_PREM_ID, IT - $MAILING_PREM_ID - Mailing Premise
IM_MAILING_PREM_ID, IM - $SEARCH_FOR_MAI_LBL - Search for Mailing Premise
PREM_INFO, IL
PROTECT_PREM_SW, CB - $PROTECT_PREM_SW - Protect Mailing Premise
dataframe, GD
```

The field names that you'll reference in your scripts are defined on the left side of the HTML (e.g., ENTITY_NAME, ACCT_ID, CUST_CL_CD, etc.).

The names of fields that reside in scrolls are in a slightly different format. The following is an example of the HTML for the persons scroll that appears on Account - Person. Notice that the fields in the scroll are prefixed with the name of the scroll plus a \$ sign. For example, the person's ID is called **ACCT_PER\$PER_ID**.

```

Widget Info:
widget_ID , Element Type - label info - label
ENTITY_NAME, IL - $ENTITY_NAME - Name
PREM_INFO, HD
ACCT_ID, IT - $ACCT_ID - Account ID
ACCT_CHECK_DIGIT, IL - $ACCT_CHECK_DIGIT - Account Check Digit
IM_ACCT_ID, IM - $SEARCH_FOR_ACC_LBL - Search for Account
ACCT_PER$recordCount, SH - $OF_LBL - of
ACCT_PER$PER_ID, IT - $PER_ID - Person ID
IM_ACCT_PER$PER_ID, IM - $FOR_PERSON_LBL - Search for Person
ACCT_PER$ENTITY_NAME, IL - $ENTITY_NAME - Name
ACCT_PER$MAIN_CUST_SW, CB - $MAIN_CUST_SW - Main Customer
ACCT_PER$FIN_RESP_SW, CB - $FIN_RESP_SW - Financially Responsible
ACCT_PER$THRD_PTY_SW, CB - $THRD_PTY_SW - Third Party Guarantor
ACCT_PER$ACCT_REL_TYPE_CD, IS - CI_ACCT_PER$ACCT_REL_TYPE_CD - Relationship Type
ACCT_PER$WEB_ACCESS_FLG, IS - $WEB_ACCESS_FLG - Web Self Service Access Flag
ACCT_PER$PFX_SFX_FLG, IS - $PFX_SFX_FLG - Prefix/Suffix
ACCT_PER$NAME_PFX_SFX, IT - $NAME_PFX_SFX - Pfx/Sfx Name
ACCT_PER$RECEIVE_COPY_SW, CB - $RECEIVE_COPY_SW - Receives Copy of Bill
ACCT_PER$BILL_RTE_TYPE_CD, IS - $BILL_RTE_TYPE_CD - Bill Route Type
ACCT_PER$BILL_RTE_TYPE_INFO, IL
ACCT_PER$BILL_RTG METH_FLG, HD
ACCT_PER$BILL_FORMAT_FLG, IS - $BILL_FORMAT_FLG - Bill Format

```

The names of fields that reside in grids are in a slightly different format. The following is an example of the HTML for the names grid that appears on Person - Main. Notice that the fields in the grid are prefixed with the name of the grid plus a :x\$. For example, the person's name is called **PER_NAME:x\$ENTITY_NAME**. When you reference such a field in your script, you have the following choices:

- Substitute **x** with the row in the grid (and keep in mind, the first row in a grid is row **0** (zero); this means the second row is row **1**).
- If you want to reference the "current row" (e.g., the row in which the cursor will be placed), you can keep the **x** notation (**x** means the "current row").

```

Widget Info:
widget_ID , Element Type - label info - label
PER_NAME:x$NAME_TYPE_FLG, IS - $NAME_TYPE_FLG - Name Type
PER_NAME:x$ENTITY_NAME, IT - CI_PER_NAME$ENTITY_NAME - Person Name

```

How To Find The Name Of Page Data Model Fields

You find the name of a **Page Data Model** field in the same way described under [How To Find The Name Of User Interface Fields](#). The only restriction is that you cannot refer to hidden / derived fields. However, you can refer to ANY of the object's fields regardless of the tab page on which they appear. For example, if you position the object display area to the Main tab of the Account transaction, you can reference fields that reside on all of the tab pages.

CAUTION: If you populate a **Page Data Model** field, none of the underlying default logic takes place. For example, if you populate a customer contact's contact type, none of the characteristics associated with the customer contact type are defaulted onto the customer contact. If you want the underlying defaulting to take place, you must populate a **User Interface Field**.

How To Find The Name Of A Button

If you want a **Press a button** step to press a button or click a link in the application toolbar, use one of the following names:

- IM_GOBACK
- IM_HISTORY
- IM_GOFORWARD
- IM_menuButton
- IM_USER_HOME
- IM_MY_PREF
- IM_helpButton
- IM_aboutButton

If you want a **Press a button** step to press a button in the page actions toolbar, use one of the following names:

- IM_SAVE
- IM_REFRESH
- IM_CLEAR
- IM_COPY
- IM_DELETE
- IM_ScrollBack
- IM_ScrollForward

The following buttons are also supported:

- IM_TO_DO. This brings you to the [To Do Summary](#) page.
- IM_PrevTo Do. This simulates clicking the **Previous To Do** button in the [Current To Do Zone](#).
- IM_NextTo Do. This simulates clicking the **Next To Do** button in the [Current To Do Zone](#).
- IM_CurrentTo Do. This navigates to the [To Do Entry](#) page for the user's current To Do. Refer to [A User's Current To Do](#) for more information.
- IM_MINIMIZE_DASHBOARD. Pressing this will collapse the dashboard.
- IM_MAXIMIZE_DASHBOARD. Pressing this will expand the dashboard.

Follow these steps to find the name of other buttons that reside in the object display area:

- Navigate to the page in question.
- Right click in the body of the page (but not while the pointer is in an input field). Note, if the field in question resides in a grid, you must right click while the pointer is in the section that contains the grid (but not while the pointer is in an input field in the grid) - this is because there's a separate HTML document for each grid on a page.
- The option to select may differ based on the browser you are using. For example, for some browsers, the option may be **View Source**. For others, the option may be **This Frame** and then **Frame Source**
- Scroll to the Widget Info section (towards the top of the HTML document). It contains a list of all of the objects on a page, including buttons.
- Iconized buttons (e.g., search buttons) are represented as HTML images and their field names are prefixed with **IM**. The following is an example of the HTML on the To Do Entry - Main page (notice the **IM** fields for the iconized buttons).

```
* Widget Info:
*   Widget_ID , Element Type - label info - label
*   TD_ENTRY_INFO, IL - $TD_ENTRY_INFO - To Do Info
*   TD_ENTRY_ID, IT - CI_TD_ENTRY$TD_ENTRY_ID - To Do ID
*   IM_TD_ENTRY_ID, IM - $TD_ENTRY_ID_SRCH - Search for Entry Id
*   TD_TYPE_CD, IL - $TD_TYPE_CD - To Do Type
*   TYPE_DESCR, IL
*   ROLE_ID, IL
*   ROLE_DESCR2, IL - $DESCR - Description
*   FULL_MSG, PL - $GOTO_TD_ACTION_LBL - Work on To Do
*   IM_EXP_MSG_LONG, IM - $DISPLAY_MESSAG_LBL - Display Message Explanation
```

- Transaction-specific actions buttons (e.g., the buttons use to complete or forward a To Do) are represented as switches. The following is an example of the HTML on the To Do Entry - Main page (notice the **SW** fields for the buttons). Note, if you want to **Set focus** to such a field, you would move a **Predefined Value** of **TRUE** to the switch.

```
*   COMPLETE_SW, BU - $COMPLETE_SW - Complete
*   FORWARD_SW, BU - $FORWARD_SW - Forward
*   SEND_BACK_SW, BU - $SEND_BACK_SW - Send Back
```

How To Substitute Variables In Text

You can substitute field values into a step's text string. You do this by prefixing the field name whose value should be substituted in the string with a %. For example, the message, "On %COMPLETION_DTTM this bill was completed, it's ending balance was %ENDING_BALANCE" contains two substitution variables (the bill's completion date / time and the bill's ending balance).

To substitute the value of an element from a data area you need to reference its XPath location as follows: %=XPath=%. If you want to substitute the whole XML node, not just the value, you need to reference it as follows %+XPath+%.

Only fields linked to the [current To Do](#) and fields that reside in [temporary storage](#) and [global variables](#) can be substituted into a text string.

NOTE: You can substitute fields that reside in the User Interface or Page Data Model by first moving them into temporary storage (using a **Move data** step).

You can also substitute field values into the verbiage displayed in [prompts](#) using the same technique.

How To Use HTML Tags And Spans In Text Strings and Prompts

You can use HTML tags in a step's text string. For example, the word "Continue" will be italicized in the following text string "Press<i>Continue</i> after you've selected the customer" (the <i> and </i> are the HTML tags used to indicate that the surrounded text should be italicized).

The following are other useful HTML tags:

- **
** causes a line break in a text string. If you use **

** a blank line will appear.
- **** text **** causes the surrounded text to be colored as specified (in this case, red). You can also use hex codes rather than the color name.

Please refer to an HTML reference manual or website for more examples.

You can also use "spans" to customize the look of the contents of a text string. For example, your text string could be "Press Continue after you've selected the customer". This would make the word "Continue" appear as large, bold, Courier text. Please refer to a Cascading Style Sheets (CSS) reference manual or website for more examples.

How To Use Constants In Scripts

Some steps can reference fields called **Predefined Values**. For example, if you want to compare an input value to the letter "Y", the letter **Y** would be defined as a Predefined Value's field value.

Special constants are used for fields defined as switches. When you move **TRUE** to a switch, it turns it on. When you move **FALSE** to a switch, it turns it off.

You can use a [global variable](#) as a Predefined Value. For example, if you wanted to move the current date to a field, you'd indicate you wanted to move a Predefined Value named %CURRENT_DATE.

How To Use Global Variables

As described above, some steps can reference fields called **Predefined Values**. In addition to referencing an ad hoc constant value (e.g., the letter **Y**), you can also reference a global variable in such a field value. A global variable is used when you want to reference system data. The following global variables exist:

- **%PARAM-<name>** is the value of a parameter of that name passed in to the application when launched via the standard system URL. Refer to [Launching A Script When Starting the System](#) for more information on these parameters.

- **%PARM-NOT-SET** is to be used to compare against **%PARM-<name>** parameters to check if the parameter has been set or not when the application was launched. A parameter that has not been set would test as equal to this global variable. It is recommended to test parameters against this global variable before using them for the first time.
- **%CONTEXT-PERSONID** is a constant that contains the ID of the current person
- **%CONTEXT-ACCOUNTID** is a constant that contains the ID of the current account
- **%CONTEXT-PREMISEID** is a constant that contains the ID of the current premise
- **%BLANK** is a constant that contains a blank value, i.e. no value
- **%SPACE** is a constant that contains a single space value
- **%CURRENT-DATE** is the current date (as known by the browser, not the server)
- **%SAVE-REQUIRED** is a flag that contains an indication of whether the data on a page has been changed (and this requires saving). You may want to interrogate this flag to force a user to save their work before executing subsequent steps. This flag will have a value of **TRUE** or **FALSE**.
- **%NEWLINE** is a constant that contains a new line character (carriage return). Upon substitution, a line break is inserted in the resultant text.

NOTE: The constant **%NEWLINE** does not have the desired effect when the resultant text is HTML. For example, a step's text and prompt strings. This is because HTML ignores special characters such as new lines. Refer to [How To Use HTML Tags And Spans In Text](#) to learn how to cause a line break in an HTML text.

In addition, if an **Invoke Function** step returns an error, the following global variables contain information about the error:

- **%ERRMSG-CATEGORY** and **%ERRMSG-NUMBER** contain the unique identifier of the error message number.
- **%ERRMSG-TEXT** contains the brief description of the error.
- **%ERRMSG-LONG** contains the complete description of the error.

How To Name Temporary Storage Fields

Input Data and **Move Data** steps can create fields in temporary storage. You specify the name of the temporary storage field in the step's **Field Name**. The name of the field must NOT begin with % and must not be named the same as the [global variables](#). Besides this restriction, you can use any **Field Name** that's acceptable to JavaScript (i.e., you can name a field in temporary storage almost anything). Keep in mind that field names are case-sensitive.

How To Work With Dates

Before we discuss how to work with dates in your scripts, we need to point out that there are two types of date fields: date-only and date-time. Date-only fields only contain a date. Date-time fields contain both a date and a time. The following topics describe how to work with dates on the various step types.

NOTE: If you're working with a field that resides on the database (as opposed to a temporary storage field), the database field name will tell you what type of date it is: date-only fields are suffixed with **DT**, and date-time fields are suffixed with **DTTM**.

Move Data

If you intend to use a **Move data** step to populate a *date-time* field, please be aware of the following:

- If the destination field resides in the *page data model*, the source field value must be in the format YYYY-MM-DD-HH.MM.SS or YYYY-MM-DD. If the field is in the format YYYY-MM-DD, the time of 12:00 am will be defaulted.
- If the destination field resides in the *user interface*, you must use two steps if you want to populate both date and time. To explain this, we'll assume the field you want to populate is called **EXPIRE_DTTM**:

- First, you populate the date portion of the field. To do this, you'd move a date (this value can be in any valid date format that a user is allowed to enter) to a field called EXPIRE_DTTM_FWDDTM_P1. In other words, you suffix **_FWDDTM_P1** to the field name.
- If you want to populate the time, you'd move the time (again, the field value can be in any format that a user could use to enter a time) to a field called EXPIRE_DTTM_FWDTTM_P2. In other words, you suffix **_FWDDTM_P2** to the field name.

If you intend to use a **Move data** step to populate a *date-only* field, please be aware of the following:

- If the destination field resides in the *page data model*, the source field value must be in the format YYYY-MM-DD.
- If the destination field resides in the *user interface*, the source field can be in any valid date format that a user is allowed to enter.

NOTE: %CURRENT-DATE. Keep in mind that the [global variable%CURRENT-DATE](#) contains the current date and you can move this to either a page data model, user interface, or temporary storage field. If you move %CURRENT-DATE to a temporary storage fields, it is held in the format YYYY-MM-DD.

Mathematical Operation

If you intend to use a **Mathematical operation** step to calculate a date, you can reference both date-only and date-time fields. This is because mathematical operations are only performed against the date portion of date-time fields.

Mathematical operations are limited to adding or subtracting days, months and years to / from a date.

NOTE: A useful technique to perform date arithmetic using the current date is to move the [global variable%CURRENT-DATE](#) to a temporary storage field and then perform the math on this field.

Input Data

If you intend to use an **Input data** step on a *date-time* field, please be aware of the following:

- If the field resides in the *page data model*, the user must enter a value in the format YYYY-MM-DD-HH.MM.SS (and therefore we do not recommend doing this).
- If the field resides in the *user interface*, you must use two steps if you want to populate both date and time. To explain this, we'll assume the field you want to populate is called EXPIRE_DTTM:
 - First, you populate the date portion of the field. To do this, you'd input the date (this value can be in any valid date format that a user is allowed to enter) in a field called EXPIRE_DTTM_FWDDTM_P1. In other words, you suffix **_FWDDTM_P1** to the field name.
 - If you want to populate the time, you'd input the time (again, the field value can be in any format that a user could use to enter a time) in a field called EXPIRE_DTTM_FWDTTM_P2. In other words, you suffix **_FWDDTM_P2** to the field name.

If you intend to use an **Input data** step to populate a *date-only* field, please be aware of the following:

- If the field resides in the *page data model*, the user must enter a value in the format YYYY-MM-DD (and therefore we do not recommend doing this).
- If the field resides in the *user interface*, the user can enter any valid date format.

How To Use To Do Fields

As described under [Executing A Script When A To Do Entry Is Selected](#), you can set up the system to automatically launch a script when a user selects a To Do entry. These types of scripts invariably need to access data that resides on the selected To Do entry. The following points describe the type of information that resides on To Do entries:

- **Sort keys.** These values define the various ways a To Do list's entries may be sorted. For example, when you look at the bill segment error To Do List, you have the option of sorting the entries in error number order, account name order, or in customer class order. There is a sort key value for each of these options.
- **Message parameters.** These values are used when the system finds %n notation within the message text. The %n notation causes field values to be substituted into a message before it's displayed. For example, the message text **The %1 non-cash deposit for %2 expires on %3** will have the values of three fields merged into it before it is displayed to the user (%1 is the type of non-cash deposit, %2 is the name of the customer, and %3 is the expiration date of the non-cash deposit). Each of these three values is stored as a separate message parameter on the To Do entry.
- **Drill keys.** These values are the keys passed to the page if a user drilled down on the entry (and the system wasn't set up to launch a script). For example, a To Do entry that has been set up to display an account on the account maintenance page has a drill key of the respective account ID.
- **To Do ID.** Every To Do entry has a unique identifier referred to as its To Do ID.

You can access this information in the following types of steps:

- **Move Data** steps can move any of the above to any data area. For example, you might want to move a To Do entry's drill key to the page data model so it can be used to navigate to a specific page.
- **Conditional Branch** steps can perform conditional logic based on any of the above. For example, you can perform conditional logic based on a To Do entry's message number (note, message numbers are frequently held in sort keys).
- **Mathematical Operation** steps can use the above in mathematical operations.

A To Do entry's sort key values are accessed by using a **Field Type** of **Current To Do Information** and a **Field Name** of **SORTKEY[index]**. Note, you can find an entry's potential sort keys by displaying the entry's To Do type and navigating to the [Sort Keys](#) tab. If you want to reference the first sort key, use an index value of **1**. If you want to use the second sort key, use an index value of **2** (and so on).

A To Do entry's drill key values are accessed by using a **Field Type** of **Current To Do Information** and a **Field Name** of **DRILLKEY[index]**. Note, you can find an entry's potential drill keys by displaying the entry's To Do type and navigating to the [Drill Keys](#) tab. If you want to use the first drill key, use an index value of **1**. If you want to use the second drill key, use an index value of **2** (and so on).

A To Do entry's message parameters are accessed by using a **Field Type** of **Current To Do Information** and a **Field Value** of **MSGPARAM[index]**. Note, because a To Do type can have an unlimited number of messages and each message can have different parameters, finding an entry's message parameters requires some digging. The easiest way to determine these values is to display the To Do entry on [To Do maintenance](#). On this page, you will find the entry's message category/number adjacent to the description. Once you know these values, display the message category/number on [Message Maintenance](#). You'll find the message typically contains one or more %n notations (one for each message parameter). For example, the message text **The %1 non-cash deposit for %2 expires on %3** has three message parameters. You then need to deduce what each of the message parameters are. You do this by comparing the message on the To Do entry with the base message (it should be fairly intuitive as to what each message parameter is). If we continue using our example, %1 is the non-cash deposit type, %2 is the account name, and %3 is the expiration date. You can access these in your scripts by using appropriate index value in **MSGPARAM[index]**.

A To Do entry's unique ID is accessed by using a **Field Type** of **Current To Do Information** and a **Field Value** of **TD_ENTRY_ID**.

In addition, any of the above fields can be [substituted into a text string or prompt](#). Simply prefix the To Do field name with a % as you would fields in temporary storage. For example, assume you want your script to display the following text in the script area: "ABC Supply does not have a bill cycle" (where ABC Supply is the account's name). If the first sort key linked to the To Do entry contains the account's name, you'd enter a text string of **%SORTKEY[1] does not have a bill cycle**.

How To Reference Fields In Data Areas

Various step types involve referencing field elements residing in the [script's data areas](#). To reference an element in a data area you need to provide its absolute XPath notation starting from the data area name. For example, use "CaseLogAdd/caseID" to reference a top-level "caseID" element in a script data area called "CaseLogAdd".

You don't have to type in long XPath notions. Use the **View Script Schema** hyperlink provided on the [Script - Step](#) tab page to launch the script's data areas schema.

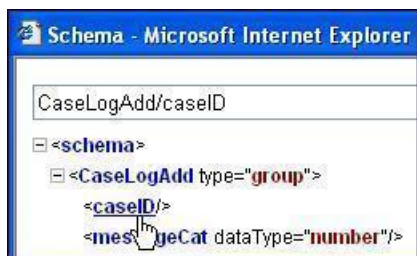


Figure 9: Schema Viewer

Doing this opens the [schema viewer](#) window where you can:

- Click on the field element you want to reference in your script step. The system automatically populates the text box on the top with the element's absolute XPath notation.
- Copy the element's XPath notation from the text box to your script.

You can also use the **View Data Area**, **View Service Script Data Area**, or **View Plug-In Script Data Area** links on [Script - Data Area](#) to the same effect. These open up the schema viewer for a specific data area respectively.

Script - Data Area

Use this page to define the data areas used to pass information to and from the server or any other data area describing your temporary storage. Open this page using **Admin > System > Script** and then navigate to the **Data Area** tab.

Description of Page

The grid contains the script's data areas declaration. For steps that invoke an object that is associated with a schema, you must declare the associated schema as a data area for your script. In addition, if you have defined one or more data areas to describe the script's temporary storage, you need to declare them too. The following bullets provide a brief description of each field on a script data area:

- **Schema Type** defines the type of schema describing the data area's element structure.
- The data area's schema is the one associated with the referenced **Object**. Only objects of the specified Schema Type may be selected.
- **Data Area Name** uniquely identifies the data area for referencing purposes. By default, the system assigns a data area with the associated object name.
- Click on the **View Data Area** link to view the data area's schema in the [schema viewer](#) window.

The **View Service Script Data Area** link appears for service scripts only. Use this link to view the script's parameters data area schema in the [schema viewer](#) window.

The **View Plug-In Script Data Area** link appears for plug-in scripts only. Use this link to view the script's parameters data area schema in the [schema viewer](#) window.

FASTPATH: Refer to [A Script May Declare Data Areas](#) for more information on data areas.

Script - Schema

Use this page to define the data elements passed to and from a service script. Open this page using **Admin > System > Script** and then navigate to the **Schema** tab.

NOTE: Conditional tab page. This tab page only appears for [service scripts](#).

Description of Page

The contents of this section describe the zones that are available on this portal.

The **General Information** zone displays the script name and description.

The [Schema Designer](#) zone allows you to edit the service script's parameters schema. The purpose of the schema is to describe the input and output parameters used when invoking the script.

NOTE: Script Definition Tips. A context sensitive "Script Tips" zone is associated with this page. The zone provides a complete list of the XML nodes and attributes available to you when you construct a schema as well as other useful information assisting with setting up scripts.

The **Schema Usage Tree** zone summarizes all cross-references to this schema. For each type of referencing entity, the [tree](#) displays a summary node showing a total count of referencing items. The summary node appears if at least one referencing item exists. Expand the node to list the referencing items and use their description to navigate to their corresponding pages.

Script - Eligibility

Use this page to define a script's eligibility rules. Open this page using **Admin > System > Script** and then navigate to the **Eligibility** tab.

NOTE: Conditional tab page. This tab page only appears for [BPA scripts](#).

Description of Page

Use the **Eligibility Option** to indicate whether the script is **Always Eligible**, **Never Eligible** or to **Apply Eligibility Criteria**. The remaining fields on the page are only visible if the option is **Apply Eligibility Criteria**.

CAUTION: The following information is not intuitive; we strongly recommend that you follow the guidelines under [The Big Picture Of Script Eligibility](#) before attempting to define this information.

The **Eligibility Criteria Group** scroll contains one entry for each group of eligibility criteria. The following fields may be defined for each group:

- Use **Sort Sequence** to control the relative order in which the group is executed when the system determines if the script should appear in the [script search](#).
- Use **Description** and **Long Description** to describe the criteria group.
- Use **If Group is True** to define what should happen if the eligibility criteria (defined in the following grid) return a value of **True**.
 - Choose **Eligible** if this script should appear.
 - Choose **Ineligible** if this script should not appear.
 - Choose **Check Next Group** if the next criteria group should be checked.
- Use **If Group is False** to define what should happen if the eligibility criteria (defined in the following grid) return a value of **False**.
 - Choose **Eligible** if this script should appear.
 - Choose **Ineligible** if this script should not appear.
 - Choose **Check Next Group** if the next criteria group should be checked.

The grid that follows contains the script's eligibility criteria. Think of each row as an "if statement" that can result in the related eligibility group being true or false. For example, you might have a row that indicates the script is eligible if the current account in context belongs to the residential customer class. The following bullets provide a brief description of

each field on an eligibility criterion. Please refer to [Defining Logical Criteria](#) for several examples of how this information can be used.

- Use **Sort Sequence** to control the order in which the criteria are checked.
- Use **Criteria Field** to define the field to compare:
 - Choose **Algorithm** if you want to compare anything other than a characteristic. Push the adjacent search button to select the algorithm that is responsible for retrieving the comparison value. Click [here](#) to see the algorithm types available for this plug-in spot.
 - Some products may also include an option to choose **Characteristic**. Choosing this option displays adjacent fields to define the object on which the characteristic resides and the characteristic type. The objects whose characteristic values may be available to choose from depend on your product.
- Use **Criteria Comparison** to define the method of comparison:
 - Choose **Algorithm** if you want an algorithm to perform the comparison and return a value of True, False or Insufficient Data. Push the adjacent search button to select the algorithm that is responsible for performing the comparison. Click [here](#) to see the algorithm types available for this plug-in spot.
 - Choose any other option if you want to compare the **Criteria Field** using a logical operator. The following options are available:
 - Use **>**, **<**, **=**, **>=**, **<=**, **<>** (not equal) to compare the **Criteria Field** using standard logical operators. Enter the comparison value in the adjacent field.
 - Use **IN** to compare the **Criteria Field** to a list of values. Each value is separated by a comma. For example, if a field value must equal **1**, **3** or **9**, you would enter a comparison value of **1,3,9**.
 - Use **BETWEEN** to compare the **Criteria Field** to a range of values. For example, if a field value must be between **1** and **9**, you would enter a comparison value of **1,9**. Note, the comparison is inclusive of the low and high values.
- The next three fields control whether the related logical criteria cause the eligibility group to be considered true or false:
 - Use **If True** to control what happens if the related logical criterion returns a value of True. You have the options of **Group is true**, **Group is false**, or **Check next condition**. If you indicate **Group is true** or **Group is false**, the script is judged **Ineligible** or **Eligible** based on the values defined above in **If Group is False** and **If Group is True**.
 - Use **If False** to control what happens if the related logical criterion returns a value of False. You have the options of **Group is true**, **Group is false**, or **Check next condition**. If you indicate **Group is true** or **Group is false**, the script is judged **Ineligible** or **Eligible** based on the values defined above in **If Group is False** and **If Group is True**.
 - Use **If Insufficient Data** to control what happens if the related logical criterion returns a value of "Insufficient Data". You have the options of **Group is true**, **Group is false**, or **Check next condition**. If you indicate **Group is true** or **Group is false**, the script is judged **Ineligible** or **Eligible** based on the values defined above in **If Group is False** and **If Group is True**.

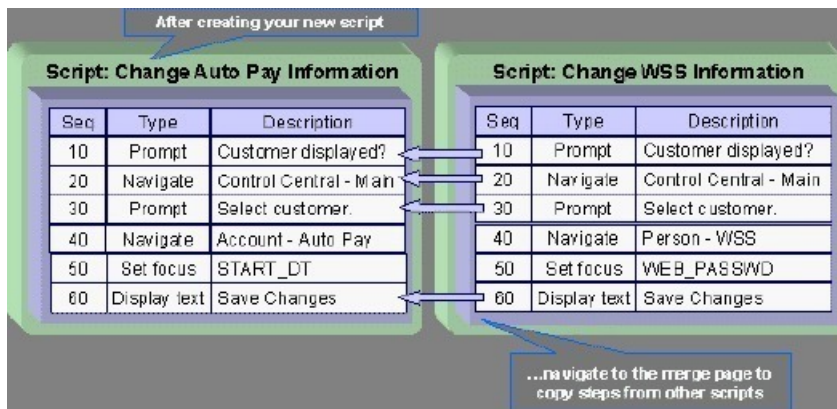
Merging Scripts

Use the Script Merge page to modify an existing script by copying steps from other scripts. The following points summarize the many diverse functions available on the Script Merge transaction:

- You can use this transaction to renumber steps (assign them new sequence numbers).
- You can use this transaction to move a step to a different position within a script. When a step is moved, all references to the step are changed to reflect the new sequence number.
- You can use this transaction to delete a step.
- You can use this transaction to copy steps from other scripts. For example,

- You may want to create a script that is similar to an existing script. Rather than copying all the information from the existing script and then removing the inapplicable steps, this page may be used to selectively copy steps from the existing script to the new script.
- You may have scripts that are very similar, but still unique. You can use this transaction to build large scripts from smaller scripts. In this scenario, you may choose to create special 'mini' scripts, one for each of the various options that may make a script unique. Then, you could use the script merge page to select and merge the mini scripts that are applicable for a main script.

NOTE: The target script must exist prior to using this page. If you are creating a new script, you must first create the [Script](#) and then navigate to the merge page to copy step information.



NOTE: Duplicate versus Merge. The [Script](#) page itself has [duplication](#) capability. You would duplicate a script if you want to a) create a new script AND b) populate it with *all* the steps from an existing script.

Script Merge

Open **Admin > System > Script Merge** to open this page.

Description of Page

For **Original Script**, select the target script for merging steps.

For **Merge From Script**, select the template script from which to copy the steps.

NOTE: You may only copy steps from one Merge From script at a time. If you want to copy steps from more than one script, select the first Merge From script, copy the desired steps, save the original script, and then select the next Merge From script.

The left portion of the page displays any existing steps for the **Original Script**. The right portion of the page displays the existing steps for the **Merge From Script**.

You can use the **Copy All** button to copy all the steps from the **Merge From** script to the **Original** script. If you use **Copy All**, the steps are added to the end of the original script.

Each time you save the changes, the system rennumbers the steps in the original script using the **Start From Sequence Number** and **Increment By**.

Merge Type indicates **Original** for steps that have already been saved in the original script or **Merge** for steps that have been merged, but not yet saved. The **Sequence**, **Step Type** and **Description** for each step are displayed.

The topics that follow describe how to perform common maintenance tasks:

Resequencing Steps


If you need to resequence the steps:

- Use the up and down arrows in the Original Script grid to reorder the steps.
- Make any desired changes to the **Start From Sequence Number** or **Increment By**.
- Click Save.




The steps are given new sequence numbers according to their order in the grid.

FASTPATH: Refer to [Editable Grid](#) in the system wide standards documentation for more information about adding records to a collection by selecting from a list and repositioning rows within a grid.


Removing a Step from Script

If you want to remove a record linked to the Original script, click the delete button, , to the left of the record.

For example, to remove the **Reset existing bundle XML** step, click the  icon.

		Merge Type	Sequence	Step Type	Description
	 	Original	10	Edit data	Edit data - Check that the BO is an Export Bundle
	 	Original	20	Edit data	Edit data - Read the Bundle
	 	Original	30	Edit data	Edit data - Reset existing bundle XML
	 	Original	40	Edit data	Edit data - Create new bundle XML

After removal, the grid displays:

		Merge Type	Sequence	Step Type	Description
	 	Original	10	Edit data	Edit data - Check that the BO is an Export Bundle
	 	Original	20	Edit data	Edit data - Read the Bundle
	 	Original	40	Edit data	Edit data - Create new bundle XML

NOTE: You cannot delete a step that is referenced by other steps unless you also delete the referencing steps, such as **Go to step** or **Prompt** type steps. The system informs you of any missing referenced steps when you attempt to save the original script.

Adding a Step to a Script

You can move any of the steps from the Merge From script to the Original Script by clicking the left arrow adjacent to the desired step. Once a record is moved it disappears from the Merge From information and appears in the Original information with the word **Merge** in the Merge Type column.

For example, to copy the **Navigate to a page** step, click the left arrow.









Merge Type	Sequence	Step Type	Description
	10	Height	Height - 0%
	20	Move data	Move data - Copy 'APP_SVC_ID' to 'APP_SVC_ID'
	30	Move data	Move data - Copy 'USR_GRP_ID' to 'USR_GRP_ID'
	40	Navigate to a page	Navigate to a page - userGroupAppService
	50	Conditional branch	Conditional branch - Compare ACTION with UPDATE
	60	Press a button	Press a button - IM_scrollSec_add
	70	Move data	Move data - Copy 'APP_SVC_ID' to 'UGP\$APP_SVC_ID'
	80	Label	Label - End of Script

The step is moved to the left portion of the page.

Merge Type	Sequence	Step Type	Description
Merge	40	Navigate to a page	Navigate to a page - userGroupAppService
	10	Height	Height - 0%
	20	Move data	Move data - Copy 'APP_SVC_ID' to 'APP_SVC_ID'
	30	Move data	Move data - Copy 'USR_GRP_ID' to 'USR_GRP_ID'
	50	Conditional branch	Conditional branch - Compare ACTION with UPDATE
	60	Press a button	Press a button - IM_scrollSec_add
	70	Move data	Move data - Copy 'APP_SVC_ID' to 'UGP\$APP_SVC_ID'
	80	Label	Label - End of Script

NOTE: If you add a step, such as **Go to step** or **Prompt** type steps, that references other steps, you must also add the referenced steps. The step references are updated to use the new sequence numbers when you save the original script. The system informs you of any referenced steps that haven't been added when you attempt to save the original script.

Removing an Uncommitted Step from a Script

	Merge Type	Sequence	Step Type	Description
	Merge	40	Navigate to a page	Navigate to userGroup
		10	Height	Height - 0%
		20	Move data	Move data - Copy 'APP_SVC_ID' to 'APP_SVC_ID'
		30	Move data	Move data - Copy 'USR_GRP_ID' to 'USR_GRP_ID'
		50	Conditional branch	Conditional branch - Compare ACTION with UPDATE
		60	Press a button	Press a button - IM_scrollSec_add
		70	Move data	Move data - Copy 'APP_SVC_ID' to 'UGP\$APP_SVC_ID'
		80	Label	Label - End of Script

Maintaining Functions

NOTE: Functions were implemented prior to the introduction of business services (BS), service scripts (SS) and business objects (BO). The functionality is still supported, but the recommendation for implementations going forward is to use one of the above configuration tool objects in a script rather than defining a function. The documentation has not been updated throughout this section to highlight where BS, SS or BO could be used to perform the equivalent logic.

Invoke function steps may be used to retrieve or update data independent of the page currently being displayed. For example, if you design a script that takes different paths based on the customer's customer class, you could invoke a function to retrieve the customer's customer class. Doing this is much more efficient than the alternative of transferring to the account page and retrieving the customer class from the Main page.

An **Invoke function** step retrieves or updates the relevant data by executing a service (on the server). These types of steps do not refer to the service directly. Rather, they reference a "function" and the function, in turn, references the service.

NOTE: Functions are abstractions of services. A function is nothing more than meta-data defining the name of a service and how to send data to it and retrieve data from it. Functions allow you to define a scriptwriter's interface to services. They also allow you to simplify a scriptwriter's set up burden as functions can handle the movement of data into and out of the service's XML document.

The topics in this section describe how to set up a function.

NOTE: You can retrieve data from all base-package objects. If you know the name of the base-package "page" service used to inquire upon an object, you can retrieve the value of any of its fields for use in your scripts. To do this, set up a function that sends the unique identifier of the object to the service and retrieves the desired fields from it.

Function - Main

Use this page to define basic information about a function. Open this page using **Admin > System > Function**.

Description of Page

Enter a unique **Function** code and **Description** for the function.

Use the **Long Description** to describe, in detail, what the function does.

Define the **Internal Service** that the function invokes.

NOTE: In this release, only page services can be invoked.

Click the **View XML** hyperlink to view the XML document used to pass data to and from the service. Doing this causes the XML document to be displayed in the Application Viewer.

NOTE: XML document may not be viewable. If you create a new page service and do not regenerate the application viewer, you will not be able to view its XML document.

The tree summarizes the following:

- The fields sent to the service. You can use the hyperlink to transfer to the **Send Fields** tab with the corresponding field displayed.
- The fields received from the service. You can use the hyperlink to transfer to the **Receive Fields** tab with the corresponding field displayed.
- Scripts that reference the function. You can use the hyperlink to transfer to the script page.

Function - Send Fields

Use this page to add or update the fields sent to the service. Open this page using **Admin > Function > System** and then navigate to the **Send Fields** tab.

NOTE: Displaying a specific field. Rather than scrolling through each field, you can navigate to a field by clicking on the respective node in the tree on the Main tab. Also note, you can use the Alt+right arrow and Alt+left arrow accelerator keys to quickly display the next and previous entry in the scroll.

NOTE: You're defining the service's input fields. On this tab, you define which fields are populated in the XML document that is sent to the service. Essentially, these are the service's input fields.

Description of Page

Use **Sequence** to define the order of the **Send Fields**.

Enter a unique **Function Field Name** and **Description** for each field sent to the application service. Feel free to enter **Comments** to describe how the field is used by the service.

Use **Field Value Source** to define the source of the field value in the XML document sent to the service:

- If the field's value is the same every time the function is invoked, select **Defined On The Function**. Fields of this type typically are used to support "hard-coded" input values (so that the scriptwriter doesn't have to populate the field every time they invoke the function). Enter the "hard-coded" **Field Value** in the adjacent field.
- If the field's value is supplied by the script, select **Supplied By The Invoker**. For example, if the function retrieves an account's customer class, the script would need to supply the value of the account ID (because a different account ID is passed each time the function is invoked). Turn on **Required** if the invoker must supply the field's value (it's possible to have optional input fields).

Regardless of the Field Value Source, use **XML Population Logic** to define the XPath expression used to populate the field's value in the XML document sent to the service.

NOTE: Usability suggestion. You populate a field's value in an XML document by specifying the appropriate XPath expression for each field. Rather than referring to an XPath manual, the system can create the XPath expression for you. To do this, click the adjacent **View XML** hyperlink. This will display the XML document used to communicate with the **Service** defined on the Main page. After the XML document is displayed, click the **XPath** hyperlink adjacent to

the desired field to see how the XPath expression looks. You can then cut / paste this XPath expression into the **XML Population Logic Field**.

Function - Receive Fields

Use this page to add or update the fields received from the service. Open this page using **Admin > System > Function** and then navigate to the **Receive Fields** tab.

NOTE: Displaying a specific field. Rather than scrolling through each field, you can navigate to a field by clicking on the respective node in the tree on the Main tab. Also note, you can use the Alt+right arrow and Alt+left arrow accelerator keys to quickly display the next and previous entry in the scroll.

NOTE: You're defining the application service's output fields. On this tab, you define which fields are populated in the XML document that is received from the service. Essentially, these are the service's output fields.

Description of Page

Use **Sequence** to define the order of the **Receive Fields**.

Enter a unique **Function Field Name** and **Description** for each field received from the service. Feel free to enter **Comments** to describe the potential values returned from the service.

Turn on **Required** if the invoker must use the field.

Regardless of the Field Value Source, use **XML Population Logic** to define the XPath expression used to retrieve the field's value from the XML document received from the service.

NOTE: Usability suggestion. You retrieve a field's value in an XML document by specifying the appropriate XPath expression for the field. Rather than referring to an XPath manual, the system can create the XPath expression for you. To do this, click the adjacent **View XML** hyperlink. This will display the XML document used to communicate with the **Service** defined on the Main page. After the XML document is displayed, click the **XPath** hyperlink adjacent to the desired field to see how the XPath expression looks. You can then copy / paste this XPath expression into the **XML Population Logic Field**.

NOTE: Fields in multiple lists. Note that the XPath expression generated in the application viewer refers to lists using a generic "list" reference. If a field within the list is unique across the service, the generic list reference is sufficient for the XML population logic. However, if the field you are trying to reference is in multiple lists, the XPath must include the list name. Adjust the Application Viewer's generated XPath by adding the list name, which can be found in the overview panel in the Service XML viewer. For example, instead of `/pageBody/list/listBody/field[@name='FIELD_NAME']`, the XPath Population Logic must read `/pageBody/list[@name='LIST_NAME']/listBody/field[@name='FIELD_NAME']`.

Attachments

Some implementations may require that attachments be available from the application. These attachments can be stored in the Attachment table and then linked to other objects if applicable.

Attachment Overview

The following topics provide additional information regarding attachment functionality.

Attachment Types

The system supports several different attachment content types, for example:

- PDF Document
- Excel Spreadsheet
- Jpeg Image
- Text Document

The attachment data itself may be text or binary. When storing the data in the application however, it is stored as text information only. As a result, the upload of an attachment that is a binary type requires a conversion prior to storing the data. When viewing the attachment, the data is converted again for display.

Each type of attachment is defined using an attachment business object. The business object includes configuration defining the supported file extensions, whether the data is binary or not and the content type that represents the type of data for the attachment.

NOTE: To view the attachment business objects provided with the base product, navigate using **Admin > Business Object > Search** and search for business objects related to the Maintenance Object for Attachments (**F1-ATCHMT**).

Owned Attachments

Attachments can be either ‘owned’ or ‘common’. An owned attachment is one that is related to a specific record. For example, the specific test results for a given device can be uploaded and linked to that device or to its test records. These types of attachments are typically uploaded and maintained via the object that owns it.

Common Attachments

Common attachments are ones that are uploaded independent of any transaction in the system. They can be used for general system or company information. Or they can be linked to more than one transaction. For example, instructions for performing a certain type of task can be uploaded as an attachment and linked to a task type where those instructions are relevant. These types of attachments are uploaded and maintained in the central Attachment portal. Objects that may refer to the attachments may link the attachments via characteristics or some other appropriate mechanism.

Emailing Attachments

The system supports a business service that may be used by system processing to send an email. The business service **F1-EmailService** supports receiving the IDs of one or more attachments as input parameters.

Refer to [Sending Email](#) for more information.

Configuring Your System for Attachments

In order to link attachments to objects in the system, there may be some configuration or implementation required to support the link. It is possible that one or more objects in your product already support attachments out of the box. Consult the product documentation for the specific object for confirmation. For objects in the system that do not support attachments out of the box, the following sections provide some guidelines for enabling support for attachments. Contact product support for more information.

Supporting Common Attachments

The attachments themselves are created / uploaded using the attachment portal. Refer to [Maintaining Attachments](#) for more information.

If your implementation has a use case where one or more common attachments may be linked to an object (and the object does not already support this functionality), the object may need to be extended to capture the attachments.

- If the object includes a characteristic collection, this is a recommended way to capture attachments. A characteristic type should be defined for each type of attachment. The characteristic type should be a foreign key type and should reference the Attachment FK reference. The characteristic entity collection should include the object that the common attachment will be linked to.
- Most characteristic collections are sequence based characteristics and would support multiple entries for the same characteristic types, if multiple attachments are applicable.
- If the object to support the attachments is governed by a business object, the implementation must extend the business object to define one or more appropriate elements used to capture the attachments. If only one attachment of a certain type is allowed, a single flattened characteristic may be used. If multiple attachments of a certain type are allowed, the BO schema may define a “flattened list” exposing the sequence and the characteristic type.
- If the object is maintained on a “fixed page” with a generic characteristic collection, no additional configuration is needed to allow users to link attachments to that object.

Supporting Owned Attachments

When creating an attachment for a specific record, the attachment itself captures the information about the related record, namely its maintenance object code and its primary key. For these types of attachments, no configuration is needed on the related business object to capture the attachments, as was the case with common attachments.

However, it is recommended to configure the user interface of the related object so that the owned attachments can be viewed and maintained from that page. This typically entails the creation of a special zone that retrieves a list of existing attachment records that reference the current record as its foreign key (owner). The zone would include links or buttons to add, upload or view an attachment.

If your product already has support for viewing and maintaining owned attachments on one of the base portals, that may be used as an example to follow. If your product does not have support for owned attachments, contact customer support for more information.

Defining a New Attachment Type

As mentioned, the product provides support for several content types. If your implementation needs to support attachments for a content type not currently supported, create a new business object copying the configuration of an existing attachment business object.

Configure the following option types for the BO:

- **Binary** indicates whether the attachment data must be converted from binary format. Binary attachments are stored in the database as text, and are then converted back to the original format when retrieved.
- **Content Type** represents the browser's mime type of the attachment.
- **Supported File Extension** specifies the valid file extensions for the content type.

Once the business object is defined, it is ready for use.

Maintaining Attachments

This section describes the functionality supported for viewing and maintaining attachments.

Navigate using **Admin > General > Attachment**. You are brought to a query portal with options for searching for common attachments.

Once an attachment has been selected, you are brought to the maintenance portal to view and maintain the selected record.

NOTE: The base search options for the attachments query only support searching for common attachments. Owned attachments may also be viewed on the attachment maintenance portal, but a user may only drill into the attachment maintenance from the maintenance portal of the “owning” entity.

The Attachment zone provides basic information about an attachment, including the ability to upload the file and to view an uploaded file.

Adding Attachments

Common attachments may be added from the attachments portal (or via the standard menu path). In addition, your product may support attachments associated with specific records (“entity owned attachments”) which may also provide the capability to add attachments.

In both cases, when adding an attachment, you are prompted for the file to upload. Once the file is chose, the system determines the appropriate business object to associate with the attachment based on the file extension. Typically one and only one business object is found at which point you are prompted to provide the Attachment Name. (Your specific product may also require additional information at this time). Fill in the details and save.

Please note the following:

- If no business object is found for the uploaded file’s file type, an error is issued. This type of file is not currently supported as an attachment.
- If multiple business objects are found, the user must choose the appropriate one. This should be rare.

Application Viewer

The Application Viewer allows you to explore meta-data driven relationships and other deliverable files online.

NOTE: Running Stand-Alone. You can also launch the Application Viewer as a stand-alone application (i.e., you do not need to start it from within the system). Refer to [Application Viewer Stand-Alone Operation](#) for more information about running the Application Viewer as a stand-alone application.

To open the application viewer from within your application, navigate to **Admin > Application Viewer**. The application viewer may also be launched from other locations for example when viewing a section of the online help files that contain hypertext for a table name, clicking on that hypertext brings you to the definition of that table in the data dictionary.

Application Viewer Toolbar

The Toolbar provides the main controls for using the Application Viewer. Each button is described below.

Data Dictionary Button



The **Data Dictionary** button switches to the Data Dictionary application.

Physical and Logical Buttons



The **Physical** button changes the display in the List Panel from a logical name view to a physical name view. Note that the Tables are subsequently sorted by the physical name and therefore may not be in the same order as the logical name view. Once clicked, this button toggles to the Logical button.

The **Logical** button changes the display in the List Panel from a physical name view to a logical name view. Note that the Tables are subsequently sorted by the logical name and therefore may not be in the same order as the physical name view. Once clicked, this button toggles to the Physical button.

These buttons are only available in the Data Dictionary.

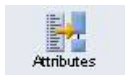
Collapse Button



The **Collapse** button closes any expanded components on the list panel so that the child items are no longer displayed.

This button is only available in the Data Dictionary viewer.

Attributes and Schema Button



The **Attributes** button changes the display in the Detail Panel from a related tables view to an attribute view. Once clicked, this button toggles to the Schema button.



The **Schema** button changes the display in the Detail Panel from an attribute view to a related tables view. Once clicked, this button toggles to the Attributes button. Note that only tables have this view available. Columns are always displayed in an attribute view.

These buttons are only available in the Data Dictionary.

Maintenance Object Button



The **Maintenance Object** button switches to the Maintenance Object viewer application.

Algorithm Button



The **Algorithm** button switches to the Algorithm viewer application.

Batch Control Button



The **Batch Control** button switches to the Batch Control viewer application.

To Do Type Button



The **To Do Type** button switches to the To Do Type viewer application.

Description and Code Buttons



The **Description** button changes the display in the List Panel to Description (Code) from Code (Description). Note that the list is subsequently sorted by the description. Once clicked, this button toggles to the Code button.

The **Code** button changes the display in the List Panel to Code (Description) from Description (Code). Note that the list is subsequently sorted by the Code. Once clicked, this button toggles to the Description button.

These buttons are only available in the Batch Control and To Do Type viewers.

Service XML Button



The **Service XML** button switches to the Service XML viewer. This button is not available when you are already in the Service XML viewer.

You are prompted to enter the name of the service XML file you want to view. The name of the service XML file should be entered without the extension.

A dialog box with a light blue border. It has a label "Service XML Name" followed by a white text input field. Below the input field are two yellow buttons: "Load" on the left and "Cancel" on the right.

Select Service Button



The **Select Service** button loads another service XML file that you specify. This button is only available in the Service XML viewer.

You are prompted to enter the name of the service XML file you want to view. The name of the service XML file should be entered without the extension.

Java Docs Button



The **Java Docs** button switches to the Java Docs viewer.

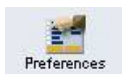
Classic Button



This button is only available in the Java Docs viewer.

The **Classic** button launches the classic Javadocs viewer on a separate window. If you are more comfortable with that look you can use this viewer instead.

Preferences Button



The **Preferences** button allows you to set optional switches used by the Application Viewer. Refer to [Application Viewer Preferences](#) for more information.

Help Button



The **Help** button opens the Application Viewer help system. You used this button to access this information.

About Button



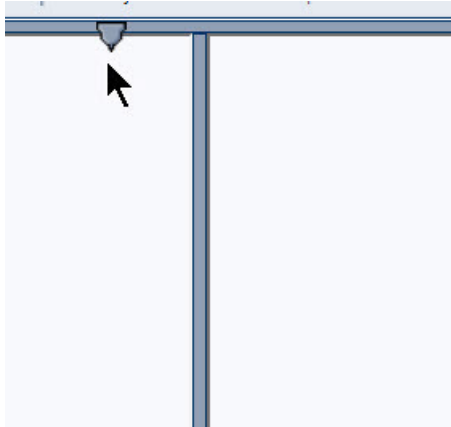
The **About** button opens a window that shows when was each Application Viewer data component recently built.

Data for all application viewer components may be regenerated to incorporate up-to-date implementation-specific information. Refer to [Application Viewer Generation](#) for further details.

Slider Icon



This "slider" icon allows you to resize the list panel and detail panel to your preferred proportions.



Data Dictionary

The data dictionary is an interactive tool that allows you to browse the database schema and to graphically view relationships between tables in the system.

To open the data dictionary, click the [Data Dictionary button](#). You can also open the data dictionary by clicking the name of a table in other parts of the application viewer or in the online help documentation.

NOTE: Data Is Generated. A background process generated the data dictionary information. Refer to [Application Viewer Generation](#) for further details.

Using the Data Dictionary List Panel

The list panel displays a list of tables and their columns. The list panel can list the table names by either their logical names or their physical names. Click the appropriate [button](#) on the toolbar to switch between the two views. The list is displayed in alphabetical order, so the order may not be the same in both views. Both views function in a similar manner.

In the list panel, you can navigate using the following options:

- Click the right arrow icon to expand a table to show its columns.
- Click the down arrow icon to collapse the column list for a table. Optionally, collapse all column lists by using the **Collapse** button.
- Click the column name to display information about the column in the detail panel.
- If the detail panel is in [related table](#) view, click the table name to view its related tables. If the detail panel is in [table detail](#) view, click the table name to display its information.

Primary And Foreign Keys

The columns in the list panel may display key information as well as the column name:

- A yellow key indicates that the column is a primary key for the table.
- A light blue key indicates that the column is a foreign key to another table. If you hover the cursor over the icon, the tool tip indicates the foreign table.
- A dark blue key indicates that the column is a conditional foreign key. A conditional foreign key represents rare relationships between tables where a single field (or set of fields) may reference multiple primary key constraints of other tables within the application as a foreign key.
- A red key indicates that the column is a logical key field. A logical key represents an alternate unique identifier of a record based on a different set of fields than the primary key.

If you hover your cursor over an icon, the tool tip indicates the key type.

Field Descriptions Shown

The language-specific, logical name of each field is shown adjacent to the physical column name in the data dictionary. You can enter an override label for a [table / field's](#) to be used throughout the system as the field's logical name. Here too it is the override label that is shown.

NOTE: Regenerate. You should regenerate the data dictionary after overriding labels. Refer to [Application Viewer Generation](#) for further details.

Using the Data Dictionary Detail Panel

The Data Dictionary detail panel displays the details of the selected item. There are three main displays for the Detail Panel:

- Related tables view
- Table detail view
- Column detail view

Related Tables View

The Related Tables view displays information about the table's parent tables and child tables. Click the [Schema](#) button in the toolbar to switch to related tables view.

In the related tables view, you can navigate using the following options:


- Click the left arrow and right arrow icons to view the related tables for that linked table. The List Panel is automatically positioned to the selected table.
- Click the maintenance object icon () to view the table's maintenance object.
- If you want to position the [List Panel](#) to view the columns for different table click the name of the table for which you want to view the columns.

Table Detail View

The table detail view displays information about the selected table. Click [Attributes](#) (in the toolbar) to switch to the table detail view.

In the table detail view, you can navigate using the following options:

- If user documentation is available for the table, click the View User Documentation link to read the user documentation that describes the table's maintenance object.
- If the table has an associated Language Table, click the link to view the Language Table details.
- If there is an associated Maintenance Program, click the link to view the source code for the maintenance program (you are transferred to the [Java Docs Viewer](#)).
- If there is an associated Key Table, click the link to view the Key Table details.

Column Detail View

Click on a column name in the list panel to switch to the column detail view. The Column Detail view displays information about the selected column.

In the column detail view, you can navigate using the following options:

- If user documentation is available for the column, click the View User Documentation link to read about the column's related maintenance object.
- If the column is a foreign key, click the table name to switch to the Table Detail view for that table.
- If the column has a Value List available (normally only present for a subset of flag and switch fields), click the link to view the source code for the copybook (you are transferred to the [Java Docs Viewer](#)).

Lookup Values

If the selected column is a lookup field its valid values are also listed. Notice that you can enter an override description for [lookup values](#). In this case the override description is shown.

NOTE: Regenerate. You should regenerate the data dictionary after overriding lookup value descriptions. Refer to [Application Viewer Generation](#) for further details.

Maintenance Object Viewer

The maintenance object viewer is an interactive tool that allows you to view a schematic diagram of a maintenance object. A maintenance object is a group of tables that are maintained as a unit.

To open the Maintenance Object Viewer, click the [Maint. Object](#) button in the application viewer or click a [maintenance object icon](#) in the Data Dictionary.

NOTE: Data Is Generated. A background process generated the maintenance object information. Refer to [Application Viewer Generation](#) for further details.


Using the Maintenance Object List Panel

The list panel displays a list of maintenance objects. In the list panel, you can click the maintenance object name to display information about the maintenance object in the detail panel.

Using the Maintenance Object Detail Panel

The Maintenance Object detail panel displays a schematic of the selected maintenance object.

In the detail panel, you can navigate using the following options:

- Click a table name to transfer to the Data Dictionary [table detail view](#) for a table. (Click the [Maint. Object](#) button in the toolbar to return to the maintenance object.)
- Click the service XML icon () to view the XML file of the Service Program used to maintain the displayed object. (Click the [Maintenance Object](#) button in the toolbar to return to the maintenance object.)

Algorithm Viewer

The algorithm viewer is an interactive tool that allows you to view algorithm types (grouped by their plug-in spot) and their related algorithms.



To open the Algorithm Viewer, click the [Algorithm](#) button in the application viewer. The Algorithm viewer may also be opened from certain locations in the online help documentation.

NOTE: Data Is Generated. A background process generates algorithm information. Refer to [Application Viewer Generation](#) for further details.

Using the Algorithm Viewer List Panel

The list panel displays a list of algorithm types and their related algorithms, grouped by their plug-in spot.

In the list panel, you can navigate using the following options:

- Click the algorithm plug-in spot description to display information about the plug-in spot in the detail panel.
- Click the right pointer  icon to expand a plug-in spot and view its algorithm types and their related algorithms.
- Click the down pointer  icon to collapse the list of algorithm types for a plug-in spot.
- Click the algorithm type name to display information about the algorithm type in the detail panel.
- Click the algorithm name to display information about the algorithm in the detail panel.

Using the Algorithm Plug-In Spot Detail Panel

The Algorithm plug-in spot detail panel displays further information about the selected plug-in spot.

Using the Algorithm Type Detail Panel

The Algorithm Type detail panel displays further information about the selected algorithm type.

In the Algorithm Type detail panel, you can navigate using the following options:

- Click on the program name to view its source in the Java docs viewer.

Using the Algorithm Detail Panel

The Algorithm detail panel displays further information about the selected algorithm.

Batch Control Viewer

The batch control viewer is an interactive tool that allows you to view batch controls.

To open the Batch Control Viewer, click the [Batch Control](#) button in the application viewer. The Batch Control viewer may also be opened from certain locations in the online help documentation.

NOTE: Data Is Generated. A background process generates batch control information. Refer to [Application Viewer Generation](#) for further details.

Using the Batch Control Viewer List Panel

The list panel displays a list of batch controls. The list panel can display the list of batch controls sorted by their code or sorted by their description. Click the appropriate [button](#) on the toolbar to switch between sorting by the code and description.

In the list panel, you can click the batch control to display information about the batch control in the detail panel.

NOTE: Not All Batch Controls Included. Note that the insertion and key generation programs for conversion (CIPV*) are not included.

Using the Batch Control Detail Panel

The batch control detail panel displays further information about the selected batch control.

In the batch control detail panel, you can navigate using the following options:

- Click on the program name to view its source in the Java docs viewer.
- If a To Do type references this batch control as its creation or routing process, click on the To Do type to view its detail in the To Do type viewer.

To Do Type Viewer

The to do type viewer is an interactive tool that allows you to view to do types defined in the system.

To open the To Do Type Viewer, click the [To Do Type](#) button in the application viewer. The To Do Type viewer may also be opened from certain locations in the online help documentation.

NOTE: Data Is Generated. A background process generates To Do type information. Refer to [Application Viewer Generation](#) for further details.

Using the To Do Type Viewer List Panel

The list panel displays a list of To Do types. The list panel can display the list of To Do types sorted by their code or sorted by their description. Click the appropriate [button](#) on the toolbar to switch between sorting by the code and description.

In the list panel, you can click the To Do type to display information about the To Do type in the detail panel.

Using the To Do Type Detail Panel

The To Do type detail panel displays further information about the selected To Do type.

In the To Do type detail panel, you can navigate using the following options:

- If the To Do type references a creation process or a routing process, click on the batch process to view its detail in the batch control viewer.
- Click on the table listed in the drill key section to view its detail in the data dictionary.
- Click on the field(s) listed in the drill key section to view its detail in the data dictionary.

Service XML Viewer

The service XML viewer is an interactive tool that allows you to browse the XML files of service programs that execute on the application server.

You can access the service XML viewer as follows:

- The maintenance object viewer allows you to view the XML file of the maintenance object's service program. This feature is implemented by viewing the maintenance object and then clicking on the [Service XML icon](#).
- When viewing a maintenance object on the [Maintenance Object](#) page, clicking the **View XML** hyperlink causes the service's XML document to be displayed in the Service XML Viewer.
- When viewing a business service on the [Business Service](#) page, clicking the **View XML** hyperlink causes the service's XML document to be displayed in the Service XML Viewer.
- When setting up a [Function](#), you may want to view the XML document used to pass data to and from the service. Clicking the **View XML** hyperlink causes the XML document to be displayed in the Service XML Viewer.

Using the Service XML Viewer Overview Panel

The overview panel displays a high level nodes and list names structure of the XML document.

In the overview panel, you can click on any node item to position the detail panel to view that item.

Using the Service XML Viewer Detail Panel

The detail panel displays nodes and attributes of the selected XML file.

Click the **xpath** button to view the XML path that should be used to reference the selected node in the XML document. The box at the top of the overview panel changes to display this information.

NOTE: Fields in multiple lists. Note that the generated XPath expression refers to lists using a generic “list” reference. For example: `/pageBody/list/listBody/field[@name='FIELD_NAME']`. If a service has a field that appears in more than one list, the above XPath may not be sufficient for referencing that field. In this case, references to the XPath should be adjusted to include the list name. The list name is visible in the overview panel. To add the list name, use `[@name='LIST_NAME']`. For example: `/pageBody/list[@name='LIST_NAME']/listBody/field[@name='FIELD_NAME']`.

Java Docs Viewer

The Java Docs viewer is an interactive tool that allows you to browse Java documentation files (Javadocs) for Java classes that execute on the application server.

NOTE: Proprietary Java Classes. A small number of Java classes have been suppressed due to their proprietary nature.

NOTE: Classic view. If you are more comfortable using the classic Javadocs viewer you may use the [Classic](#) button.

To open the Java Docs viewer from within the application viewer, click the [Java Docs button](#). Additionally, the [algorithm viewer](#) and the [batch control viewer](#) allows you to view the Javadocs of a program written in Java.

Using the Java Docs Viewer List Panel

The list panel displays a tree of Java packages where each package may be expanded to list the Java interfaces classes it includes.

In the list panel, you can navigate using the following options:

- Click the right arrow icon to expand a Java package to view the Java interfaces and classes it includes.
- Click the down arrow icon to collapse the list for a Java package. Optionally, collapse all lists by using the **Collapse** button.
- Click the Java interface or class name to display information about it in the detail panel.

The list details panel designates the interfaces and the classes as follows:

- A green dot indicates Java interfaces.
- A blue key indicates Java classes.

If you hover the cursor over the icon, the tool tip indicates whether it's an interface or a class.

Using the Java Package Detail Panel

The package detail panel displays a summary of the various Java classes that are included in the selected Java package.

Click the Java class name to display information about the Java class in the detail panel.

Using the Java Interface / Class Detail Panel

The detail panel displays Java documentation information about the selected Java interface or class.

You can navigate using hyperlinks to other locations in the current detail panel or to view the details of other Java interfaces / classes.

Application Viewer Preferences

This panel displays the Available Languages and allows you to select the language in which the labels and buttons are displayed. Select your desired language and click OK.

Application Viewer Stand-Alone Operation

You can run the Application Viewer as a stand-alone application (i.e., you do not need to launch it from the online application environment). To run it as a stand-alone application, you should copy the Application Viewer files (all files in the appViewer directory) and the online help files (all files in the help directory) to the server on which you want to run the Application Viewer.

NOTE: Online Help. If you do not copy the online help files, online help will not be available for the Application Viewer, nor will you be able to view business descriptions of the tables' maintenance objects.

To start the application viewer in stand-alone mode, launch the appViewer.html file (located in the appViewer directory).

Stand-Alone Configuration Options

You can configure the Application Viewer for stand-alone operation by modifying options in a configuration file. The Application Viewer comes with a default configuration file called `config_default.xml` (located in the `appViewer\config` directory). Create a copy of the default configuration file and rename it to `config.xml`. Modify the options described in the following table to suit the needs of your installation.

NOTE: Default Configuration. If you do not create the `config.xml` file, the Application Viewer launches with its default (internal) configuration.

Option	Description
<code>defaultLanguage</code>	The default language used when the application viewer is started. Available values are those marked as language enabled on the language page.
<code>defaultView</code>	The default view then the application viewer is started. Available values include: - Data Dictionary
<code>dataDictionary</code>	Whether the Data Dictionary is available or not: - Y - N
<code>sourceCode</code>	This property is not being used. Simply enter 'N'.
<code>baseHelpLocation</code>	The location of the stand-alone online help in relation to the application viewer. Specify the directory structure relative to the location of the directory in which the Application Viewer files are located. Note that this is the directory in which the language subdirectories for the online help are located. The default location is: ../help
<code>appViewerHelp</code>	The default help topic that is launched when the Help button is clicked in the Application Viewer. Specify a help file and anchor that is under the appropriate language directory under the <code>baseHelpLocation</code> . The default is: Framework/Admin/91AppViewer.html#SPLINKApplication_Viewer

Example Application Viewer Configuration

The following excerpt shows an example Application Viewer configuration.

```
<?xml version="1.0" encoding="UTF-8" ?>
<configuration>
<option id="defaultLanguage">PTB</option>
<option id="defaultView">Data Dictionary</option>
<option id="dataDictionary">Y</option>
<option id="sourceCode">N</option>
<option id="baseHelpLocation">../help</option>
<option id="appViewerHelp">Framework/Admin/91AppViewer.html#SPLINKApplication_Viewer</option>
</configuration>
```

Application Viewer Generation

The Application Viewer is initially delivered with service XML information only.

The other components of the application viewer are generated on site.

- Use the background process **F1-AVALG** to regenerate algorithm information
- Use the background process **F1-AVBT** to regenerate batch control information.
- Use the background process **F1-AVMO** to regenerate maintenance object information
- Use the background process **F1-AVTBL** to regenerate data dictionary information.
- Use the background process **F1-AVTD** to regenerate To Do type information.

These processes have been introduced so that you can more easily incorporate your implementation-specific information into the application viewer.

To keep the information shown in the application viewer current it is important to execute these background processes after you introduce changes to the corresponding system data.

NOTE: Data Generation Is Not Incremental. Each new execution of these processes first deletes existing data (if any) in the corresponding folder before it generates new data.

NOTE: Other Extensions. Service XML may also be extended to include implementation-specific information. The base package is provided with special scripts that handle this type of extension. Refer to the Software Development Kit User Guide for further information on application viewer extensions.

NOTE: War File. If your application is installed in war file format, each generation of application viewer data rebuilds the corresponding war file. The web application server then needs to be "bounced" in order to make the newly generated data available to the application viewer. Please consult your system administrator for assistance.

NOTE: Certain Web Application Servers Are Special. WebSphere and Oracle Application web application servers require an additional step in order to make the newly generated data available to the application viewer. These web application servers require a rebuild of the application ear file and its redeployment in the web application server. This step is described in the installation document. Please consult your system administrator for further details.

Defining and Designing Reports

This section describes how to configure your third party reporting tool and how to define your reports in the system to enable users to submit reports online.

The Big Picture Of Reports

The topics in this section describe the approach for designing and defining your system reports.

Integration with BI Publisher and Business Objects Enterprise

Your DBMS, your product, and BI Publisher or Business Objects Enterprise / Crystal Reports can work together to produce reports. You may choose to use a different reporting tool, but this may not be a trivial effort. This section provides high-level information about some of the business requirements that are being solved with the reporting solution.

Reports Must Be Multi-Language

The system supports a multi-language implementation and the reporting solution for the system must also support a multi-language implementation.

- If a French-speaking user requests a given report, all labels, headings and messages are in French. If the same report is requested by an English-speaking user, all information is in English
- Different fonts may be necessary for a given report (the font for Chinese is rather different than the font for English)
- Dates and numbers must be formatted as per the user's profile in your product
- Currency must be formatted as per the currency definition in your product

In order to provide the above functionality, the third party reporting tool must do the following:

- Access the system's field and message metadata for labels, headings and messages (as different values can be defined for different languages in system's metadata)
- Retrieve the appropriate font, size, and layout based on the requested report and the user's language
- Use the currency code information in the system to format currency-oriented information
- Use the user's display profile to format date and number fields

Requesting Reports from The System

Although reports are rendered in your reporting tool, users must be able to request ad-hoc reports from within the system (assuming users have the appropriate security access).

- The prompts for the input parameters must be shown in the user's language
- Users should be able to use the standard search facilities to find parameter values
- Plug-ins can be optionally used to cross-validate input parameters
- Application security must authorize ad-hoc report requests

Overview of the Data - BI Publisher

The following diagram provides an overview of where data is stored for your reports for integration with BI Publisher.

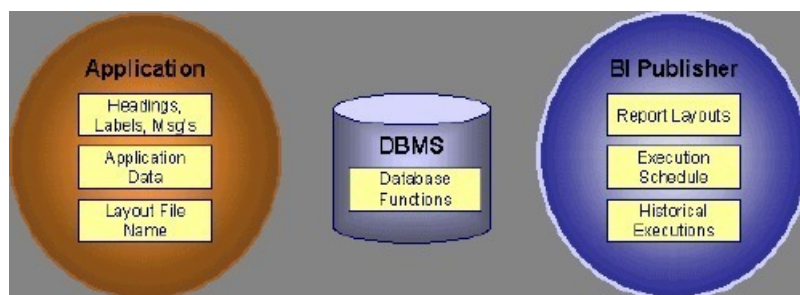


Figure 10: Application and BI Publisher

The application contains:

- The application data that appears on your reports.
- The language-specific headings, labels and messages on your reports.
- The layout file name to be used for the report.

BI Publisher contains:

- How your reports look.
- Information about scheduled reports and reports that have already run.

The DBMS contains the SQL used to retrieve the data on your reports (residing in database functions).

NOTE: BI Publisher can be configured to retrieve data via a service call. Because every business object can be read via a service call, elements that reside in an XML based column can be displayed on reports produced using BI Publisher. See your product's *Optional Products Installation Guide* for information on this configuration.

Overview of the Data - Business Objects Enterprise

The following diagram provides an overview of where data is stored for your reports for integration with Business Objects Enterprise.

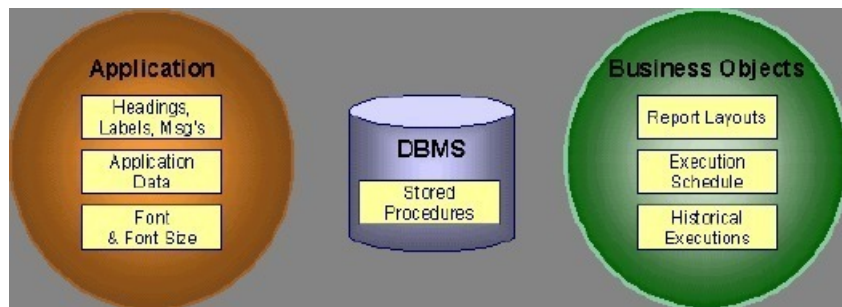


Figure 11: Application and Business Objects Enterprise

The application contains:

- The application data that appears on your reports.
- The language-specific headings, labels and messages on your reports.
- For Business Objects Enterprise, the font and size in which each report is rendered.

Business Objects Enterprise contains:

- How your reports look.
- When your reports are produced (the batch scheduler).
- Historical images of reports.

The DBMS contains the SQL used to retrieve the data on your reports (residing in stored procedures).

How To Request Reports

A user may request an ad hoc report from within your product:

- A [report submission](#) page enables a user to choose the desired report and enter the parameter values for the report
- The user must be granted security access to the report

- The request is passed to the reporting tool real time. Refer to [Configure The System to Invoke BI Publisher](#) or [Configure The System to Invoke Business Objects Enterprise](#) for more information.
- The reporting tool generates the report and displays it in a new browser window

The reporting tools' scheduler creates reports (as per your schedule)

- This function is entirely within the reporting tool. No scheduling functions reside within your product.

A user can request an ad-hoc report from within the reporting tool

- Note, the user's ID must be supplied as a parameter to the report in order for the user's profile to be used to format dates and numbers

Viewing Reports

As described above, ad-hoc reports requested from within your product are displayed immediately after they are generated in a new browser window

Crystal's report repository can be used to retrieve historical versions of a report. The [Report History](#) page allows users to open the Crystal's report execution history page and request a view of this report.

NOTE: The [Report History](#) page currently does not display historical reports for BI Publisher.

Configuring The System To Enable Reports

Configuring BI Publisher Reports

This section contains topics specific about configuring the product to interoperate with BI Publisher.

Configure the System to Invoke BI Publisher Real-time

The base product provides an [installation algorithm](#) plug-in spot called Reporting Tool. This plug-in spot should contain an algorithm that invokes the third party reporting tool real-time.

For BI Publisher, the system provides an algorithm type called [F1-BIPR-INV](#), which invokes BI Publisher.

These algorithms rely on information defined in the [Reporting Options](#) table: the reporting server, reporting folder and the user name and password for accessing the reporting tool. The values in the reporting options should have been set up when the system was installed. Contact your system administrator if there are any problems with the values defined on the reporting options.

To use the algorithm types to invoke BI Publisher, perform the following steps:

- Create an [algorithm](#) for the appropriate algorithm type.
- On the [installation options](#), add an entry to the algorithm collection with an algorithm entity of **Reporting Tool** and indicate the algorithm created in the previous step.

Batch Scheduling in BI Publisher

For many of your reports, you probably want the report to be produced on a regular basis according to a scheduler. The reporting solution relies on the BI Publisher software to provide the batch scheduler functionality. Refer to BI Publisher documentation for details about configuring the batch scheduler.

NOTE: The [report history](#) page currently does not display historical reports for BI Publisher.

Configuring Business Objects Enterprise Reports

This section contains topics specific about configuring the product to interoperate with Business Objects Enterprise.

Configure the System to Invoke Business Objects Enterprise Real-time

The base product provides an [installation algorithm](#) plug-in spot called Reporting Tool. This plug-in spot should contain an algorithm that invokes the third party reporting tool real-time.

For Business Objects Enterprise, the system provides an algorithm type called [RPTE-INV](#), which invokes Business Objects Enterprise.

These algorithms rely on information defined in the [Reporting Options](#) table: the reporting server, reporting folder and the user name and password for accessing the reporting tool. The values in the reporting options should have been set up when the system was installed. Contact your system administrator if there are any problems with the values defined on the reporting options.

To use the algorithm types to invoke one of the reporting tools, perform the following steps:

- Create an [algorithm](#) for the appropriate algorithm type.
- On the [installation options](#), add an entry to the algorithm collection with an algorithm entity of **Reporting Tool** and indicate the algorithm created in the previous step.

Batch Scheduling in Business Objects Enterprise

For many of your reports, you probably want the report to be produced on a regular basis according to a scheduler. The reporting solution relies on the Business Objects Enterprise software to provide the batch scheduler functionality. Refer to Business Objects Enterprise documentation for details about configuring the batch scheduler.

The product provides a [report history](#) page to display report instances that were produced via the batch scheduler and are stored in a repository. The report history page relies on the [reporting tool algorithm](#) to invoke Business Objects Enterprise and display the historic instances for the selected report.

Defining Reporting Options

The reporting options are provided as a mechanism for defining information needed by your reporting solution. The base product uses the reporting options to define information needed to access the reporting tool from within the system using the algorithm defined on the installation option.

Navigate to this page using **Admin > Reporting > Reporting Options**.

Description of page

The following information must be defined to interface with BI Publisher real-time. Contact your system administrator to report any problems with the settings defined here.

Reporting Folder defines the shared folder where reports are stored.

For Business Objects Enterprise, defines the name of the virtual directory on the server where Java Service pages (JSP) are located. The reporting tool algorithm uses this information to construct the URL to launch the reporting tool. The reporting tool algorithm assumes that a JSP named "logon.jsp" is located there.

Reporting Server defines the URL of the web application where the reporting tool is installed. For example, using BI Publisher, the format is: `http://<BI Publisher Server>:<port>`.

Reporting Tool User ID is not applicable when integrating with BI Publisher.

For Business Objects Enterprise, defines the user id to use when logging in.

Reporting Tool Password is not applicable when integrating with BI Publisher.

For Business Objects Enterprise, defines the password to use when logging in.

NOTE: Customize Options. The reporting options are customizable using the Lookup table. This field name is **RPT_OPT_FLG**. The reporting options provided with the system are needed to invoke the reporting tool. If your implementation requires other information to be defined as reporting options, use the lookup table to define additional values for the reporting option flag.

Where Used

This information is used by the reporting tool algorithm on the [installation option](#) to invoke the reporting tool software.

Implementations may use reporting options to record other information needed for their reporting tool.

Defining Report Definitions

For each report supplied by your installation, use the report definition page to define various attributes of the report.

Report Definition - Main

Navigate to this page using **Admin > Reporting > Report Definition**.

CAUTION: Important! If you introduce new report definitions, you must prefix the report code with **CM**. If you do not do this, there is a slight possibility that a future release of the application could introduce a new system report with the name you allocated.

Description of page

Enter an easily recognizable **Report Code** and **Description** for each report. Use the **External Reference ID** to define the identifier for this report in your external reporting tool.

Define an [application service](#) to enable users to request submission of this report online or to view report history for this report. Once you define an application service for each report, use [application security](#) to define which users may access this report.

NOTE: Access Mode. The access mode for application services related to reports must be set to **Submit/View Report**.

If you have more than one parameter defined for your report and you wish to perform cross-validation for more than one parameter, provide an appropriate **Validation Algorithm**. Click [here](#) to see the algorithm types available for this system event.

Enter a **Long Description** to more fully describe the functionality of this report. This information is displayed to the user when attempting to submit the report online or when viewing history for this report.

For BI Publisher, if you want to use one of the sample reports provided by the system, but with a different layout, indicate the layout to use for the report in the **Customer Specific Font/ Layout** field and BI Publisher uses this information instead. The name for base report layout is <report code>_Base. For example, a base layout for CI_VACANT is named CI_VACANT_Base.

For Business Objects Enterprise, the **Report Font** and **Report Font Size** are used to control the display of the report information. If you wish to use one of the sample reports provided by the system, but wish to use a different font and font size, indicate your **Customer Specific Font** in the **Customer Specific Font/Layout** field and Business Objects Enterprise uses this information instead.

Report Definition - Labels

Navigate to this page using **Admin > Reporting > Report Definition** and go to the **Labels** tab.

NOTE: Company name and logo. Note the company name used as a title in the sample reports is defined as a message on the [installation options](#). For information about installing the company logo, refer to the *Reports Configuration* chapter of the *Installation Information*.

Description of Page

In order to provide multi-language capability for each report, the labels used for the report must support multiple language definitions. For each label used by your report, indicate a unique **Sequence** and the **Field** used to define the **Label**. The label defined here should be the same label that is defined in your report layout defined in the external reporting tool.

When rendering an image of the report, the external reporting tool retrieves the appropriate label based on the language used for the report.

Report Definition - Parameters

Navigate to this page using **Admin > Reporting > Report Definition** and go to the **Parameters** tab .

Description of Page

The **Parameters** scroll contains one entry for every parameter defined for the report. To modify a parameter, simply move to a field and change its value. To add another parameter, click + to insert a row and then fill in the information for each field. The following fields display:

Parameter Code The identifier of the parameter. This must correspond to the parameter definition in the reporting tool.

Required Turn on this switch if the user must supply a value for the parameter when submitting the report.

Sort Sequence Indicate the sort sequence for this parameter. This sequence must match the parameter order defined in the reporting tool's report. It is also used when displaying the list of parameters on the [report submission](#) page.

Characteristic Type Indicate the characteristic type used to define this parameter.

Default Value Use this field to define a default value for this parameter. Default values are displayed to the user when the report is chosen on the [report submission](#) page.

Description Define a brief description of the parameter. This description is used when displaying the parameter on the [report submission](#) page.

Long Description Define a detailed description of the parameter. This description is used on the [report submission](#) page when the user requests more information for a given parameter.

Sample Reports Supplied with the Product

Depending on your specific product, there may be sample reports provided that your organization may use as they are or as a starting point for creating a [new report](#). The following sections provide an overview of the sample reports along with instructions on how to use one of the sample reports in your implementation environment.

How to Use a Sample Report Provided with the System

If you would like to use any of the sample reports, you need to perform some steps to be able to execute them in an implementation environment. This section walks you through the steps needed.

Steps Performed at Installation Time

The *Installation Guide* provides instructions for setting up and configuring your product and reporting tool to use the sample reports provided with the system. The following steps are described there.

- Setting up the stored procedures used by the sample reports.
- Defining the company title and logo used by the sample reports. Note the company name used as a title in the sample reports is defined as a message on the [installation options](#). For information about installing the company logo, refer to the *Reports Configuration* chapter of the *Installation Information*.
- Defining a user for integration with your product.
- Publishing the sample reports in BI Publisher or Business Objects Enterprise.

Contact your system administrator to verify that the above steps have occurred.

Subreports Used with Crystal Reports

The sample reports supplied with the system use several common subreports. Subreports are used in Crystal Reports to retrieve common data such as, labels and your company title. They are shared for all reports and may be reused for customer reports. Implementers may also use these subreports when designing new reports.

NOTE: Specific Subreports. This section only includes common subreports that may be reused by new reports. You may notice that other subreports are supplied with the system. These subreports provide functionality for a specific sample reports and are not meant for reuse.

Display Company Logo and Title

The subreport **CIZCOMP** receives the user id as a parameter and calls the stored procedure **CIZCOMP**. It retrieves the company's title in the user's language from the appropriate [installation message](#) record.

FASTPATH: Refer to the **Reports Configuration** chapter of the installation guide for more information about defining the location for the company logo.

Format Report Information

The subreport **CIZINST** defines shared variables that are used for formatting fields in the main report. It calls the stored procedure **CIZINST**. This subreport receives the user id and report code as parameters. It retrieves the font and font size from the [report definition](#). It retrieves the format date/time and number format from the user's [display profile](#). Finally, it retrieves the currency from the [installation](#) record and retrieves the currency symbol and position from the currency's record.

NOTE: Multi-currency. All reports support multiple currencies. Currency information is returned for each row by the appropriate stored procedure. This subreport retrieves the currency code from the Installation Options and should only be used in a report if there is no other currency information available.

Labels

The subreport **CIZLABEL** keeps all labels used in the main report. It calls the stored procedure **CIZLBALL** with the user ID as a parameter. This stored procedure returns all labels defined for all reports. The subreport selects labels specified for the current report and sets shared variables L001...L100 to store the labels. If more than 100 labels are required for a

new report, the version of the CIZLABEL subreport used for the new report should be changed to add additional shared variables.

How To Define A New Report

Use a Sample Report as a Starting Point

- Make a copy of the report and save it in an appropriate directory. Prefix the new report name with **CM**.
- Review the stored procedure(s) used for this report. Refer to the installation guide for information about where the stored procedures should be defined. If you want to change the data that is being accessed, copy the stored procedure, prefixing the new stored procedure with **CM**. Make the appropriate changes in the new version of the stored procedure. Contact your database administrator to find out the procedure for creating a new stored procedure.

NOTE: Performance considerations. When designing a stored procedure, you must consider the performance of the report when executed. Consult your database administrator when designing your database access to ensure that all issues are considered.

NOTE: Defining Messages. The stored procedures provided with the system use messages defined in message category 30. If your new stored procedures require new messages, use message category 90000 or greater, which are reserved for implementations.

- Review the parameters used by the report. Make appropriate changes to the parameters required by the report. This affects how you define your report. Refer to [Designing Parameters](#) for more information.
- Determine whether or not you require cross validation for your report parameters. If any cross validation is necessary, you should design an appropriate validation algorithm to be executed when requesting a report in your product. Refer to [Designing Validation Algorithms](#) for more information.

NOTE: Cross Validation for On-line Submission Only. The cross validation algorithm is only executed for ad-hoc report submissions via your product. If you submit this report through your reporting tool, this algorithm is not executed.

- Review the labels used by the report. Labels and other verbiage are implemented in the sample reports using a reference to the field table in the system. This enables the report to be rendered in the appropriate language for the user. For any new report label you require, you must define a new field entry. Refer to [Designing Labels](#) for more information.
- Review the layout of the report and make any desired changes based on your business needs.

When you have finished designing and coding your new report in your reporting tool, you must do the following in order for it to be usable:

- Publish the report in BI Publisher or Business Objects Enterprise. Refer to the documentation for these products for details about publishing a report. Refer to [Publishing Reports in BI Publisher](#) and [Publishing Reports in Business Objects Enterprise](#) for configuration information specific to publishing a report for integration with your product.
- Define the report. Refer to [Designing Your Report Definition](#) for more information.

Publishing Reports in BI Publisher

Please refer to the documentation for BI Publisher for more information about publishing a report in this system. The remaining topics in this section provide information about settings needed to ensure that the report is accessible using BI Publisher.

BI Publisher Database Access

When publishing a report in BI Publisher, you are asked for database logon information. The logon user name and password must be the user name and password that has access to the database functions related to this report in your database.

Verify BI Publisher User Access Rights

To verify the user's access rights to folders in BI Publisher:

- Open the BI Publisher Enterprise Security Center.
- Check that the role for the user has access to the appropriate report folders.

For more information, refer to the "Understanding Users and Roles" section in the Oracle Business Intelligence Publisher User's Guide.

Publishing Reports in Business Objects Enterprise

Please refer to the documentation for Business Objects Enterprise for more information about publishing a report in this system. The remaining topics in this section provide information about settings needed to ensure that the report is accessible using Business Objects Enterprise.

Business Objects Enterprise Database Access

When publishing a report in Business Objects Enterprise, you are asked for database logon information. The logon user name and password must be the user name and password that has access to the stored procedures related to this report in your database.

Verify Parameter Definition

This section describes how to verify parameter definitions in the Crystal Management Console (CMC).

- Once your report has been published, navigate to the CMC. This is the web-based administration component for Business Objects Enterprise and provides access to all administrative functions.
- To verify/change the settings of a report in the CMC go to the Objects management area and select the desired report by clicking its link located in the Object Title column.
- Once you have selected your report, click the **Parameters** tab to change the settings.
- If your report requires parameters to be provided by the user, you must configure the parameter settings in the CMC to ensure that parameter values are passed from the system when submitting the report via the [Report Submission](#) page. If you plan to submit reports from within your product, the **Prompt the user for new value(s) when viewing** check box should be checked.
- Click **OK** on this window and then click **Update**.

NOTE: Submitting Reports Through Business Objects Enterprise. If you plan to submit reports from Business Objects Enterprise, you must also define an appropriate initial value for each parameter, if applicable.

NOTE: User ID. The user id is defined as the first parameter in every sample report. This parameter is hidden when the report is submitted from within your product, but it must be defined in the Crystal report.

Verify Business Objects Enterprise User Access Rights

To verify the access rights for a user in CMC:

- Navigate to the **Rights** tab in the Objects management area of the CMC and check that the user has correct security level for the report.
- Integration with your product requires an access level of **View On Demand** for the user.

Designing Your Report Definition

When adding a new report, you must define it in the system to allow users to request ad-hoc reports from on-line and to take advantage of the multi-language provisions in the system. The following topics illustrate the steps to take to correctly configure your report definition.

Designing Main Report Definition Values

Refer to field description section of the [report definition](#) main page for information about defining general information about the report.

For the validation algorithm, preliminary steps are required. Refer to [Designing Validation Algorithms](#) for more information.

For the application service, preliminary steps are required. Refer to [Designing Application Services](#) for more information.

Designing Characteristic Types

The parameter tab on the report definition page uses [characteristic types](#) to define the report parameters. For each report parameter that you plan to use, you must define a characteristic type.

You do not need a unique characteristic type for each report parameter. For example, if Start Date and End Date are parameters your report, only one **Report Date** characteristic type needs to be defined. This characteristic type would be used on both date parameters.

Each characteristic type to be used as a report parameter must indicate a characteristic entity of **Report**.

To illustrate the characteristic type definitions, let's look at the sample report Tax Payables Analysis. It needs the following parameters: From Date, To Date, GL Account Type Characteristic Type and Account Type value.

NOTE: Account Type Parameters. The tax payables report must find general ledger entries that have posted to a certain distribution code. In order to find the appropriate distribution code, the report expects each distribution code to be defined with a characteristic indicating its GL account type (for example, **Revenue**, **Asset**, etc.) The report needs to know the characteristic type used to define this entry.

To support the required parameters, the following characteristic types are needed.

Char Type	Description	Type	Valid Values	Char Entities
CI_DATE	Date Parameter	Ad-hoc	(Uses validation algorithm to validate proper date entry)	Report
CI_CHTYP	Characteristic Type	FK Reference	CHAR_TYP	Report
CI_GLTY	GL Account Type	Pre-defined	A- Asset, E- Expense, LM- Liability/miscellaneous, LT- Liability/taxes, R-Revenue	Distribution Code, Report

Highlights for some of the above settings:

- We have defined a characteristic type for defining a characteristic type. This is to allow the user to indicate which Char Type on the Distribution Code is used for the GL account type. This is an FK reference type of characteristic.
- The GL account type characteristic type is referenced on both the Distribution Code entity and the report entity.

Designing Parameters

Your report definition parameters collection must define a unique parameter entry for each parameter sent to the reporting tool. The sequence of your parameters must match the sequence defined in your reporting tool.

Continuing with the Tax Payables Analysis report as an example, let's look at the parameter definitions.

Parameter Code	Description	Char Type	Default Value
P_FROM_DT	From Date	CI_DATE	N/a
P_TO_DT	To Date	CI_DATE	N/a
P_CHAR_TYPE	Account Type Characteristic	CI_CHTYP	CI_GLTYP
P_TAX_ACCTY_CHAR	Account Type Char Value for Tax Related GL Account	CI_GLTYP	LT-Liability/taxes

Highlights for some of the above settings:

- The from date and to date parameters use the same characteristic type.
- The characteristic type parameter is defined with a default value pointing to the GL account type characteristic type.
- The GL account type parameter defines the liability/taxes account type as its default value.

NOTE: User Id. The sample reports provided by the system pass the user id as the first parameter passed to the reporting tool. It does not need to be defined in the parameter collection for the report.

Designing Validation Algorithms

When designing your report definition, determine if cross validation should occur for your collection of parameters. In the Tax Payables Analysis report, there are two date parameters. Each date parameter uses the characteristic type validation algorithm to ensure that a valid date is entered. However, perhaps additional validation is needed to ensure that the start date is prior to the end date. To do this, a validation algorithm must be designed and defined on the report definition.

The system provides a sample algorithm [RPTV-DT](#) that validates that two separate date parameters do not overlap. This algorithm should be used by the Tax Payables Analysis report.

If you identify additional validation algorithm, create a new [algorithm type](#). Create an [algorithm](#) for that algorithm type with the appropriate parameter values. Plug in the new validation algorithm to the appropriate report definition.

Designing Application Services

[Application services](#) are required in order to allow a user to submit a report on-line or to view history for a report. Define an application service for each report and define the user groups that should have submit/view access to this report.

Update [report definition](#) to reference this application service.

Designing Labels

The system supports the rendering of a report in the language of the user. In order to support a report in multiple languages, the verbiage used in the report must be defined in a table that supports multiple languages. Some examples of verbiage in a report include the title, the labels and column headings and text such as "End of Report".

The system uses the [field](#) table to define its labels.

NOTE: Report Definition. This section assumes that your new report in the reporting tool has followed the standard followed in the sample reports and uses references to field names for all verbiage rather than hard-coding text in a single language.

For each label or other type of verbiage used by your report, define a field to store the text used for the verbiage.

- Navigate to the field page using **Admin > System > Field**.
- Enter a unique **Field Name**. This must correspond to the field name used in your report definition in the reporting tool and it must be prefixed with **CM**.
- Define the **Owner** as **Customer Modification**.
- Define the **Data Type** as **Character**.
- **Precision** is a required field, but is not applicable for your report fields. Enter any value here.
- Use the **Description** to define the text that should appear on the report.
- Check the **Work Field** switch. This indicates to the system that the field does not represent a field in the database.

Update the [report definition](#) to define the fields applicable for this report in the **Labels** tab.

If your installation supports multiple languages, you must define the description applicable for each supported language.

External Messages

This section describes mechanisms provided in the product that enable an implementation to configure the system to communicate with an external application.

Incoming Messages

This section provides information about support for incoming messages.

Inbound Web Services

Inbound web service functionality is provided to support receiving web service requests from an external system.

Overview

The following topics provide overview information about inbound web services (IWS).

Multiple Operations

An inbound web service supports the configuration of one or more operation per web service. Each operation defines the schema-based object to invoke to perform the desired function. An operation may refer to a Business Service, a Business Object, or a Service Script. If the IWS supports multiple operations, each operation can refer to the same or a completely different schema-based object from other operations within the IWS. In addition, each operation may define a transaction type, which is essentially an action that is supported by the schema-based object.

By default, Inbound Web Services uses the Schema Name to dictate the Request and Response for the service. The API can be overridden with custom formats by specifying Request and Response Schemas with the appropriate Request and Response XSL to transform into the relevant schema formats.

Annotations Used for Security

When preparing to deploy inbound web services, the security aspects of the service must be decided. There are three options available:

- Attach a security policy to the IWS via a Web Service Annotation. The product supplies the security policy **Username Token Policy** that may be used. This attaches a policy file (UsernameToken.xml) to the Inbound Web Service at deployment time.
- Attach security policies to the Inbound Web Service via the J2EE Web Application Server. This allows for multiple policies to be attached as support by the J2EE Web Application Server.
- A combination of the internal and external security policies.

Inbound Web Service Deployment

An Inbound Web Service must be deployed to the J2EE Web Application Server in order for it to be available to the Web Service Clients to access the system. Refer to [Deploying Web Services](#) for more information.

Deploying XAI Inbound Service via IWS

For implementations using XAI inbound services for external messages, the product recommends moving to the inbound web service mechanism, which uses the J2EE Web Application Server to communicate with the product rather than the XAI servlet.

For XAI inbound services that use the **Business Adapter**, it is straight forward to move to IWS because the configuration is similar. In both cases, the service is configured to reference a business object, business service or service script. The associated WSDL for each record is similar. Changing the interface for the incoming message to use IWS instead of XAI inbound services is similar.

However, for XAI inbound services that use the **Core Adapter**, these services reference an underlying “page service” in the product. For these services, the Request and Response schemas for the XAI inbound service were created using the Schema Editor. In order to support calling an underlying “page service” in IWS, first a [business service](#) must be created to reference the page service (if one doesn’t already exist). However, the resulting schema for the business service is different from the Request and Response schemas related to the XAI inbound service. Moving this functionality to IWS using business services requires changes to the format of the incoming messages.

Moving all incoming messages over to use IWS instead of XAI is the product recommendation. However, to aid in implementations that have many integrations in place using the XAI inbound services that use the **Core Adapter** (or any adapter whose message class is **BASEADA**), the product provides the ability to deploy these types of XAI inbound services to the J2EE Web Application Server along with the Inbound Web Services.

To take advantage of this capability, you must define a feature configuration option. Under the **External Messages** feature configuration type, the **Support XAI Services via IWS** is used to indicate if this feature is supported. Setting the value to **true** turns on the feature. If no option is defined for that option type, it is equivalent to setting the value to **false**.

When the system is configured to support XAI services via IWS, the [Inbound Web Service deployment](#) includes XAI inbound services (that are configured with an Adapter that references the **BASEADA** message class). The deployment portal will also include a zone showing the deployment status of these XAI Inbound Services.

Configuring Inbound Web Service Options

This topic describes the configuration needed for using inbound web services.

Technical Configuration

In order to use inbound web services, there are tasks a system administrator must perform.

Refer to the Server Administration Guide for technical details of each of these processes.

Maintaining Web Service Annotation Types

The product provides the annotation type **Policy Annotation** that defines the WS-Policy annotation. If your implementation wishes to define additional annotation types, use the Web Service Annotation Type portal. Open this page using **Admin > External Message > Web Service Annotation Type**.

You are taken to the query portal where you can search for an existing web service annotation type. Once an annotation type is selected, you are brought to the maintenance portal to view and maintain the selected record.

NOTE: Use of custom policies should only be considered if the policies supplied by the J2EE Web Application Server are not sufficient for your implementation's needs.

The **Web Service Annotation Type** zone provides basic information about the web service annotation type.

Please see the zone's help text for information about this zone's fields.

Maintaining Web Service Annotations

The product provides the annotation **Username Token Policy**, which references the annotation type **Policy Annotation** and enables WS-security. If your implementation wishes to define additional annotations, use the Web Service Annotation portal. Open this page using **Admin > External Message > Web Service Annotation**.

This is a standard [All-in-One portal](#).

Maintaining Inbound Web Services

[Inbound Web Services](#) are used to define a specific message that your implementation will receive from an external system and provides configuration needed to process the inbound message.

The product provides several inbound web services out of the box. By default, no annotations are defined for the base inbound web services. You may modify the message options or the annotations for any base IWS record. In addition, you may define additional IWS records for other incoming messages supported by your implementation.

To view an inbound web service, navigate using **Admin > External Message > Inbound Web Service**. You are brought to a query portal with options for searching for inbound web services.

Once an inbound web service has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The Inbound Web Service zone displays the configuration information for the record, including its annotations, if applicable and its operations.

Deploying Web Services

This topic describes the configuration needed for using inbound web services.

Once an Inbound Web Service is defined it is not automatically available to the Web Service Clients to access the system. The Deployment Status and the Active flag (set to true) indicate whether a Web Service is available or not. The last step is to deploy the Inbound Web Services to the J2EE Web Application Server. This deployment phase has a number of steps that are automatically performed when a deployment is initiated:

- The Web Service files are generated and policies are attached.
- The WSDL is generated with appropriate annotations and enumerations.
- The necessary java stub code to implement the Web Service in the J2EE Web Application Server is generated and compiled.

- The Web Services are built into a valid Web Application Archive (WAR) file.
- Optionally, the newly created Web Services WAR file is deployed to the J2EE Web Application Server. This can also be done manually for clustered deployments, if desired.

There are two methods available for deploying inbound web services:

- Deployment at the command line using the **iwdeploy[.sh]** command as outlined in the Server Administration Guide. This method is recommended for native installations and production implementations.
- Deployment using the Inbound Web Service Deployment portal. This method is recommended for development and non-production environments.

Inbound Web Service Deployment Portal

To use the online Inbound Web Service Deployment portal, navigate using **Admin > External Message > Inbound Web Service Deployment**.

The following sections describe the base zones that are provided on the portal.

Deploy Inbound Web Services

The Deploy Inbound Web Services zone provides information about the last deployment. Use the **Deploy** button to deploy or re-deploy inbound web services. All inbound web services whose Active switch is Yes will be deployed. All whose active switch is No will be undeployed.

NOTE: When an Inbound Web Service is deployed, the value of its service revision field is captured. Certain changes to configuration will require re-deployment to take effect. When any of the following changes occur, the IWS service revision value is incremented. This will cause the deployment status to show **Needs Deploy**.

- Active switch is changed
 - An Annotation is added or removed
 - An Operation is added or removed.
 - The Operation Name, Schema Type / Schema Name, Request or Response Schema, Request or Response XSL for an Operation is changed.
-

NOTE: In addition, if the implementation supports XAI services deployed through IWS, the appropriate XAI inbound services will also be deployed or undeployed as required.

Deployment Status

The Deployment Status zone displays a list of inbound web services in the product, including the deployment status.

The deployment status is determined by comparing the internal Service Revision field on each IWS against the value captured at the time of deployment.

- **Deployed.** Indicates that the IWS has been deployed and no changes have been detected to the configuration.
- **Needs Deploy.** Indicates that the IWS has never been deployed or has been deployed but in the meantime, changes have been detected to the configuration that require redeployment.
- **Undeployed.** Indicates that the IWS is marked as inactive and the IWS is not found to be deployed at this time.
- **Needs Undeploy.** Indicates that the IWS is marked as inactive but the IWS is found to be deployed at this time.

Use the broadcast button adjacent to any of the inbound web services listed in the zone to view the details of the IWS record. This causes the **Inbound Web Service** zone to appear. It is the same zone that appears on the [Inbound Web Service](#) maintenance portal.

XAI Inbound Service Deployment Status

The XAI Inbound Service Deployment Status zone is only visible if the feature configuration option **Support XAI Services via IWS** is configured on the **External Messages** feature type or if the system detects that there are XAI inbound services that have been deployed. (The latter condition is checked for the case where an implementation has XAI inbound services deployed and then chooses to discontinue using this functionality. After changing the feature configuration option to false, one more deployment is required to “undeploy” the XAI services.) The zone displays a list of XAI inbound services in the product that are related to page services. Refer to [Deploying XAI Inbound Service via IWS](#) for more information.

The deployment status is determined by comparing the record’s Version field against the value captured at the time of deployment.

- **Deployed.** Indicates that the XAI inbound service has been deployed and no changes have been detected to the configuration.
- **Needs Deploy.** Indicates that the XAI inbound service has not been deployed or has been deployed but in the meantime, changes have been detected to the configuration.
- **Undeployed.** Indicates that the XAI inbound service is marked as inactive or the **Support XAI Services via IWS** is not set to **true** and the XAI inbound service is not found to be deployed at this time.
- **Needs Undeploy.** Indicates that the XAI inbound service is marked as inactive or the **Support XAI Services via IWS** is not set to **true** but the XAI inbound service is found to be deployed at this time.

XAI inbound service does not have the equivalent of a Service Revision field that inbound web service has, which is only incremented when changes are made to the record that impact deployment. For XAI inbound service, the version number on the record is used. This field is incremented when any changes are made, even ones that may not impact deployment. As a result, some XAI Inbound Services may indicate “Needs Deploy” in cases where a redeployment may not be necessary. The recommendation when this occurs is to simply Deploy again to be safe.

Guaranteed Delivery

There are alternatives for sending messages to the system besides using inbound web services. An external system may be able to send messages to the system in a generic manner where a new web service does not need to be defined for every new type of message. These types of messages may provide a payload (the message) and the service script or business service to invoke. An example of this type of communication is a message sent from a mobile application using RESTful operations.

The external system may have no mechanism for retrying failed messages. For this situation, the product provides an algorithm that may be used to capture incoming messages that should ‘guarantee delivery’. A servlet processing this type of message may invoke the [installation algorithm](#) - Guaranteed Delivery, passing the details of the message and an indication if a response should be returned. The algorithm is responsible for storing the message information in a table so that it can be subsequently processed.

NOTE: The framework does not provide a sample algorithm for this plug-in spot. Your specific product may provide an algorithm and additional support for guaranteeing messages. Refer to your product documentation for more information.

Outgoing Messages

"Outgoing messages" is the term used to describe messages that are initiated by our system and sent to an external system. Messages may be sent real time or near real time. The system provides the following mechanisms for communicating messages to external systems.

- **Outbound Messages.** This method allows implementers to use configurable business objects to define the message format and to use scripts to build the message. If sent near real-time the message is posted to the outbound message table waiting for Oracle Service Bus to poll the records, apply the XSL and route the message. If sent real time, the message dispatcher routes the message immediately.
- **Web Service Adapters.** Using a web service adapter, an implementation can consume a WSDL from an external system and create an “adapter” record that references the URL of the external system and creates appropriate request and

response data areas to expose the payload information in a format understood by configuration tools. A script may then be written to build the request information and initiate a real-time web service call from within the system.

- **Send Email.** The system supplies a dedicated business service that may be used to send an email real-time from within the application.

All these methods are described in more detail in the following sections.

Outbound Messages

Outbound messages provide functionality for routing XML messages to an external system real-time or in near real time. In addition the functionality supports batching related messages to then be sent to an external system as a consolidate XML message.

For each outbound message that your implementation must initiate you define a [business object](#) for the outbound message maintenance object. Using the business object's schema definition, you define the fields that make up the XML source field. These are the fields that make up the basis for the XML message (prior to any XSL transformation).

Each outbound message requires the definition of its schema by creating a business object whose schema describes the information that is provided to the external system. An XSL transformation may then be performed when routing the message to an external system.

For each external system that may receive this message, you configure the appropriate message XSL and routing information.

Because the outbound message type is associated with a business object, your implementation can easily create outbound message records from a script using the **Invoke business object** script step type. Such a script would

- Determine the appropriate [outbound message type](#) and [external system](#) based on business rules
- Access the data needed to populate the message detail
- Populate the fields in the schema and use the **Invoke business object** script step type for the outbound message type's business object to store the outbound message.
- The resulting outbound message ID is returned to the script and the scriptwriter may choose to design a subsequent step to store that ID as an audit on the record that initiated this message.

The following topics provide more information about functionality supported by outbound messages.

NOTE: For implementations that continue to use MPL and XAI, there is additional functionality related to outbound messages. Refer to [Outgoing Messages](#) for more information.

Polling Outbound Messages Using OSB

If the outbound message that needs to be sent to an external system can be sent as an asynchronous message (in 'near real time'), the process initiating the outbound message should create a record in the outbound message staging table. Oracle Service Bus (OSB), is the recommended tool to use to process outbound messages in near real-time.

Outbound messages that should be processed by OSB should be configured with a processing method defined as **SOA** for the external system / outbound message type. No other information is required for defining outbound message types that are processed by OSB.

For the OSB part of the processing, the product provides a custom transport: OUAF Outbound Message that may be used by an implementation to define messages to process and how to process them.

This section provides an overview of steps required to develop OSB integrations for outbound messages created by your product.

Before developing OSB integrations, a developer should be familiar with OSB development such as creating proxy services, business services, and message flow/routing. These terms are defined as follows:

Proxy Service: In OSB, a Proxy Service is the entity that processes a given type of message and routes it to a Business Service. A separate proxy service should be defined for each type of outbound message. If a given outbound message type may be routed to different external systems, it is the responsibility of the proxy service to query the external system defined on the outbound message and invoke the appropriate business service (see below). If any transformation is required prior to routing a message to a business service, it is the proxy service's responsibility to perform the transformation.

Business Service: In OSB, a Business Service is an entity that receives a message from OSB and routes it to the appropriate destination. This should not be confused with the Business Service object provided in the product in the configuration tools.

FASTPATH: Refer to the whitepaper OSB Integration for more information.

Batch Message Processing

Your implementation may be required to send messages to the same destination as a single XML file with multiple messages include. The following points describe this logic:

- The individual messages that should be grouped together must have a processing method of **batch** on the external system / outbound message type record. The appropriate batch code that is responsible for grouping the messages must also be provided.
- A separate "consolidated message" outbound message type should be configured for the external system with a processing method of **SOA**.
- When outbound message records are created for the individual messages, the batch code and current batch run number are stamped on the record.
- When the batch process runs it is responsible for building the XML file that is a collection of the individual messages. This batch process should include the following steps:
 - Format appropriate header information for the collection of messages
 - Apply the individual message XSL to each message before including the message
 - Insert a new outbound message for the external system with the "consolidated message" outbound message type.
- The consolidated message is ready to be processed by Oracle Service Bus.

NOTE: No process provided. The system does not supply any sample batch job that does the above logic.

Real Time Messages

The system supports the ability to make web service calls, i.e. sending real time messages, to an external system. The configuration of real time messages is similar to the configuration of near real-time ones, with the following exceptions:

- The processing method defined for the outbound message type and an external system must be **Real-time**.
- The [message sender](#) defined for the outbound message type and an external system has to be set up with **Real-time** invocation type.

Just like near real-time messages, you can easily create outbound message records from a script. When a real time message is added, the system immediately routes it to the external system. If the external system provided a response message back, the system captures the response on the outbound message. If the outbound message type for the external system is associated with a response XSL it is applied to transform the response. In this case the system captures the raw response as well on the outbound message.

Any error (that can be trapped) causes the outbound message to be in a state of **Error**. It is the responsibility of the calling process to check upon the state of the outbound message and take a programmatic action. When the outbound message state is changed back to **Pending** the message will be retried.

The base package provides a business service called "Outbound Message Dispatcher (**F1-OutmsgDispatcher**)" that further facilitates making web service calls, allowing the calling script to configure the following behavior:

- Whether or not the sent message is captured as an actual outbound message record.
- Whether or not exceptions encountered while sending the message are trapped. Trapping errors allows the calling script to interrogate any errors encountered and take some other programmatic action.

Refer to the description of the business service for a better understanding on how it works.

NOTE: HTTP Sender. In the current release of the product, only senders that communicate via HTTP are supported.

Configuring the System for Outbound Messages

The following sections describe the setup required when using [outbound messages](#) to communicate with an external system. The configuration walks you through the steps to configure a single external system and all its messages.

Define the Outbound Message Type

For each outbound message that must be sent to an external system, create a [business object](#) for the outbound message maintenance object. Using the business object's schema definition, your implementation defines the fields that make up the XML source field. These are the fields that are the basis for the XML message. XSL transformations may be applied to this XML source to produce the XML message.

Once you have your business object and schema, define an [outbound message type](#) for each unique outbound message.

Define the Message Sender

When messages are routed to an external system real-time using the outbound message dispatcher or using the real-time send email business service, there must be a [Message Sender](#), which tells the system how to send the message. The two different mechanisms differ in how the system knows which sender to use:

- For sending emails, the default sender may be defined on the [message option](#) table. Alternatively, the message sender can be provided to the business service as input. Refer to [Sending Email](#) for more information. Senders of this type should be configured with the **RTEMAILSNDR** class.
- For other outbound messages that are routed using the real-time outbound message dispatcher, the sender to use is defined on the [external system](#) / outbound message type collection. Refer to [Real Time Messages](#) for more information. Senders of this type should be configured with the **RTHTTPSNDNR** class.

Define the External System and Configure the Messages

Define an [external system](#) and configure the valid outgoing messages and their method of communication (processing method). Refer to [Outbound Messages](#) for more information.

Message Sender

The topics in this section describe the maintenance of a message sender

Message Sender - Main

To define a new sender, open **Admin > External Message > Message Sender**.

Description of Page

Enter a unique **Message Sender** and **Description**.

Use **Invocation Type** to define whether the sender is a **Real-time** sender or called by **MPL** to route near real-time messages.

NOTE: The MPL invocation type remains in the product for upgrade purposes but is not recommended.

Indicate the **Message Class** for this sender. The class should be one that is defined for a sender. The real-time sender classes are **EMAILSENDER**- Email sender and **HTTPSNDR**- HTTP sender.

The following sender classes are related to MPL processing and remain in the product for upgrade purposes: **DWNSTGSNDR**- Download Staging sender, **FLATFILESNDR**- Flat file sender, **JMSENDER**- JMS Queue sender, **STGSENDER**- Staging Upload sender, **TPCSNDR**- JMS Topic sender and **UPLDERRHANDLER**- Upload Error Handler.

Indicate whether or not this sender is currently **Active**.

Indicate whether the **MSG Encoding** is **ANSI message encoding** or **UTF-8 message encoding**.

If the Message Class is **JMSENDER** or **TPCSNDR** indicate the appropriate **XAI JMS Connection**

FASTPATH: Refer to [XAI JMS Connection](#) for more information.

If the Message Class is **JMSENDER**, use the **XAI JMS Queue** to define where the response is to be sent.

FASTPATH: Refer to [XAI JMS Queue](#) for more information.

If the Message Class is **TPCSNDR**, use the **XAI JMS Topic** to define where the response is to be sent.

FASTPATH: Refer to [XAI JMS Topic](#) for more information.

If the Message Class for this sender is **STGSENDER** indicate the **XAI JDBC Connection**.

FASTPATH: Refer to [XAI JDBC Connection](#) for more information.

Message Sender - Context

The sender may require context information to define additional information needed by the system to successfully send outgoing messages. Open **Admin > External Message > Message Sender** and navigate to the **Context** page.

Description of Page

Define the **Context Type** and **Context Value**, which contain parameters for senders when more information is required. See below for some examples of context for different types of senders.

NOTE: The values for the Context Type field are customizable using the Lookup table. This field name is **SENDER_CTXFLG**.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_XAI_SENDER](#).

Email Sender Context

The email sender is used by the business service that [sends email messages real-time](#).

An email sender must point to the Message Class **EMAILSENDER**. In addition, the following context records should be defined for senders of this type.

Context Type	Description
SMTP Host name	The SMTP server host name.

Context Type	Description
SMTP Username	The user ID used to access the SMTP server.
SMTP Password	The password used to access the SMTP server.
Response Time Out	The amount of time the system should wait for a real time response.

HTTP Sender

An HTTP sender is one that sends messages to an HTTP server using the HTTP protocol. HTTP senders should reference a Message Class of **HTTSPNDR**.

Various parameters are required to establish a session with the target HTTP server. You specify these parameters by defining a collection of context values for the sender. A set of context types related to HTTP variables is provided with the product. The following section describes the context types and where appropriate, indicates valid values.

Before defining the HTTP sender, you need to find out how the HTTP server on the other side expects to receive the request, and in particular, to answer the following questions:

- What is the address of the HTTP server?
- Is the HTTP server using a POST or GET HTTP method?
- If the server is using POST, how are message contents passed? Does it use an HTTP FORM or does it pass the data in the body of an XML message?

Context Type	Description	Values
HTTP URL1 - URL9	Used to construct the URL of the target HTTP server. Since the URL may be long and complex, you can break it into smaller parts, each defined by a separate context record. The full URL is built by concatenating the values in URL1 through URL9. You may use substitution variables when entering values for URL parts.	
HTTP Method	The HTTP method used to send the message.	POST or GET
HTTP Proxy Host	If connecting to the remote system requires using an HTTP Proxy, then this context field can be used to configure the HTTP Proxy Host. If the Proxy Host is set, the Sender class must use the value specified to connect to the remote system via a proxy.	
HTTP Proxy Port	If connecting to the remote system requires using an HTTP Proxy, then this context field can be used to configure the HTTP Proxy Port. If the Proxy Port is set, the Sender class must use the value specified to connect to the remote system via a proxy. If the HTTP Proxy Host is not set, HTTP Proxy Port is ignored and the connection will be made directly to the remote system.	

Context Type	Description	Values
HTTP Transport Method	Specifies the type of the message. You can either send the message or send and wait for a response.	Send or sendReceive
HTTP Form Data	<p>Used when the message is in the format of an HTML Form (<code>Content-Type: application/x-www-form-urlencoded</code>).</p> <p>This context specifies the form parameters (data) that should be passed in the HTTP message. Since a form may have multiple parameters, you should add a context record for each form parameter.</p> <p>The value of a form parameter takes the format of <code>x=y</code> where <code>x</code> is the form parameter name and <code>y</code> is its value.</p> <p>If <code>y</code> contains the string <code>@XMLMSG@</code> (case sensitive) then this string is replaced by the content of the service response XML message.</p> <p>If a context record of this type is defined for a sender, the sender uses the HTML Form message format to send the message even if <code>@XMLMSG@</code> is not specified in one of the context records.</p> <p>If a context record of this type is not defined for a sender, then the XML is sent with <code>Content-Type: text/plain</code>. When using POST it is put in the HTTP message body.</p> <p>Always required when using the GET method. If you are using the GET method and do not specify a Form Data context record, no message is transferred to the HTTP server.</p> <p>The MPL server builds <code>formData</code> by concatenating the individual parts.</p> <p>You may use substitution variables when entering values for Form Data.</p>	
HTTP Login User	The HTTP server may require authentication. Add a context record of this type to specify the login user to use.	
HTTP Login Password	The HTTP server may require authentication. Add a context record of this type to specify the login password to use.	
HTTP Header	<p>Sometimes the HTTP server on the other side may require the addition of HTTP headers to the message.</p> <p>For each HTTP header that has to be specified you should add a context record with a value having the following format <code>x:y</code> where</p>	

Context Type	Description	Values
	x is the header name and y is the value for the header	
HTTP Time Out	Indicates the amount of time to wait for a connection to be established with the remote system.	
Character Encoding	Indicates if the message should be encoded. The sender will add to the HTTP's content type header the string ; charset=x where x is the value of this context and when sending the message it will encode the data in that encoding.	UTF-8 or UTF-16
Response Time Out	The amount of time the system should wait for a real time response.	

Defining Outbound Message Types

Use this page to define basic information about an outbound message type. Open this page using **Admin > External Message > Outbound Message Type**.

NOTE: This page is not available if the **External Message** module is [turned off](#).

Description of Page

Enter a unique **Outbound Message Type** and **Description**. Use the **Detailed Description** to describe the outbound message type in detail.

Indicate the Business Object that defines business rules and the schema for outbound messages of this type.

Indicate the relative **Priority** for processing outbound message records of this type with respect to other types.

This bottom of this page contains a [tree](#) that shows the various objects linked to the outbound message type. You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [FI_OUTMSG_TYPE](#).

External Systems

Use this page to define an external system and define the configuration for communication between your system and the external system.

External System - Main

Open this page using **Admin > External Message > External System**.

NOTE: This page is not available if both the **External Message** and the **Open Market Interchange** modules are [turned off](#).

Description of Page

Enter a unique **External System** and **Description**.

Use the field **Our Name In Their System** to specify the identity of your organization (i.e., your external system identifier) in the external system.

NOTE: The workflow process profile and notification download profile are only applicable to products that support workflow and notification. They are not visible in the product if the **Open Market Interchange** module is [turned off](#).

If this external system sends inbound communications through notification upload staging, the type of workflow process that is created is controlled by the sender's **W/F (Workflow) Process Profile**.

If you send notifications to this external system, select a **Notification DL (download) Profile** that is used to define the configuration of the outgoing messages.

NOTE: The remaining fields are not visible if the **External Message** module is [turned off](#).

Set **Usage** to **Template External System** for external systems whose outbound message type configuration is inherited by other external systems.

If the outbound message type configuration should be inherited from a template external system, define the **Template External System**. If this field is specified, the outbound message type collection displays the data defined for the template system as display-only.

The **Outbound Message Type**[accordion](#) contains an entry for every type of outbound message defined for this external system. For each type of outbound message identify its **Outbound Message Type**.

Define the **Processing Method** for messages of this type. If the value is **XAI** or **Real-time**, indicate the appropriate **Message Sender**. If the value is **Batch**, indicate the appropriate **Batch Control**.

The **Message XSL** is the schema used to transform information from the format produced by the system to a format understood by the sender, who receives a message of this type. This is not applicable for Processing Method of **SOA**.

Enter the file name of the appropriate **W3C Schema** if you want to validate the message built for outbound messages for this external system / outbound message type prior to being routed to their destination. Refer to [Outbound Message Schema Validation](#) for more information. This is not applicable for Processing Method of **SOA**.

Response XSL will have the same search service as is used for the existing Message XSL field. This field will only be displayed when the processing method is **Real-time**. Refer to [Real Time Messages](#) for more information on how it is used.

External System - Template Use

If you are viewing an external system whose usage is a **Template External System**, use this page to view the other external systems that reference this one. Open this page using **Admin > External Message > External System** and then navigate to the **Template Use** tab.

Description of Page

The tree shows every external system that references this external system as its template.

Managing Outbound Messages

Use this page to view information about outbound messages.

Outbound Message - Main

Open this page using **Menu > External Message > Outbound Message**.

Description of Page

Outbound Message ID is the system-assigned unique identifier of the outbound message. These values only appear after the outbound message is added to the database.

The **Processing Method** indicates whether this record will be processed by a **Batch** extract process, through the **XAI** tool or **Real-time**. The value defined on the external system / outbound message type collection populates this value.

When records are created with a processing method of **Batch**, the system sets Extract to **Can Be Extracted**. Change the value to **Not to be extracted** if there is some reason that processing should be held for this record.

For records with a processing method of **Batch**, **Batch Control** indicates the process that will extract this record. This value is populated based on the on the external system / outbound message type's value. **Batch Number** indicates in which batch run this record was extracted or will be extracted.

The **Retry Count** is used by the XAI tool to keep track of how many times the tool tried to process this record and could not process the record, resulting in an error.

The **Creation Date** indicates the date that this record was created.

If the processing method is **XAI**, **Status** defines the state of the outbound message record. Refer to [Lifecycle of Outbound Message](#) for more information.

For messages in **error** status, click **Pending** to change the status back to pending for reprocessing.

For messages in **pending**, **error**, or **in progress** status, click **Cancel** to cancel the message and prevent further processing.

Outbound Message - Message

Use this page to view the XML source used to build an outbound message. Open this page using **Menu > External Message > Outbound Message** and then navigate to the **Message** tab.

Description of Page

The **XML Source** is displayed.

If a message XSL is defined on the external system / outbound message type record linked to this outbound message, the **Show XML** button is enabled. Click this button to view the XML that is a result of applying the Message XSL to the XML source.

Outbound Message - Response

Use this page to display the XML response. Open this page using **Menu > External Message > Outbound Message** and then navigate to the **Response** tab.

Description of Page

The **XML Response** and optionally the **XML Raw Response** is displayed.

XML Response displays the response data from the system called by the real-time message. If a response XSL is defined on the external system / outbound message type record linked to this outbound message, a transform is performed and the XML Raw Response displays the original, unchanged response.

Message Option

The Message Option page defines various system settings used by the system when processing external messages.

Note that some of the options are only applicable to implementations still using the XAI and MPL servers. The settings here may be overridden by the [AdHoc Parameters](#) section of the XAIParameterInfo.xml or MPLParameterInfo.xml.

To define options for your environment, open **Admin > External Message > Message Option**.

Description of Page

The following options are supported.

Option	Description	MPL / XAI Option Name
Automatically Attempt Resend to Unavailable Sender (Y/N)	Set to Y if you wish to enable Automatic Resend . Set to N if you wish to log errors when the system fails to send an outgoing message.	shouldAutoResend
Default Email Sender	This is the default Message Sender used for sending e-mails when no explicit Message Sender is specified.	defaultEmailSender
Default Response Character Encoding	Determines the character encoding to be used when a response is sent. For example, you may specify UTF-8 or UTF-16 . If no value is specified then the default is UTF-8 . If no special encoding should be done, then enter the value none .	defaultResponseEncoding
Default User	The default user is used by XAI to access your product when no other user is explicitly specified. Refer to Server Security for more information. Additionally, the Default User is used for MPL transactions where there is no facility to provide a User ID. For example, no facility exists to provide a user id when reading messages from a JMS Queue. In these messaging scenarios, the system will use the Default User for authorization purposes.	defaultUser
Email Attachment File Location	This is the default location of e-mail attachment files. If not specified, the e-mail service provided with the product assumes a full path is provided with each attachment file.	emailAttachmentFileLocation
Email XSL File Location	This is the default location of e-mail XSL files. If not specified, the e-mail service provided with the product assumes a full path is provided to an XSL file as part of an e-mail request.	emailXSLFileLocation
JDBC Connection Pool Max size	The MPL uses a pool of JDBC connections to connect to the database. This option determines the maximum number of JDBC connections that can be opened in the pool. The default value is 100.	JDBCConnPoolMaxSize
Maximum Errors for a Sender	This value is required if you have enabled Automatic Resend . It defines how many errors you receive from a sender when attempting to send an outgoing message before you mark the sender unavailable .	maxSendingErrors
Messages JDBC Connection	Specifies the JDBC connection that XAI uses to read the text for its messages.	messagesJDBCConnection
Messages Language	The default language to use for the messages.	language
MPL Administrator Port	The port number to be used for receiving MPL operator commands.	adminPort
MPL HTTP Server Authentication Method	This setting, along with the MPL HTTP Server User and Password are used to secure commands received by your MPL (such as those issued via XAI Command) through HTTP. Currently only BASIC authentication is supported.	MPLHTTPAuthMethod
MPL HTTP Server Password	This setting, along with the MPL HTTP Server Authentication Method and User are used to secure commands received by your MPL (such as those issued via XAI Command) through HTTP. The password should be in encrypted form, using the same encryption that is used for the database password. .	MPLHTTPAuthPassword
MPL HTTP Server User	This setting, along with the MPL HTTP Server Authentication Method and Password are	MPLHTTPAuthUser

used to secure commands received by your MPL (such as those issued via [XAI Command](#)) through HTTP.

MPL Log File	The MPL Log File setting is used to specify the name of the file where MPL log information is to be written. The log contains technical information about the operation of the MPL.	MPLLogFile
MPL Trace File	The MPL Trace File setting is used to specify the name of the file where MPL trace information is to be written.	MPLTraceFile
MPL Trace Type	The MPL Trace Type is used to enable or disable tracing of the MPL. The possible values are FULL - All trace messages are written to the log file and NOLOG - No information is written to the log file.	MPLTraceType
Outbound Message Schema Location	Enter the full path of the virtual directory where valid W3C schemas are stored if your implementation wants to validate outbound message schemas . For example: http://localhost/cisxai/schemas.	xaiOuboundSchemaLoc
Privileged Users	Comma separated list of users that are allowed to specify an effective User or effective User Id via framework custom SOAP Headers.	superUsers
Records MPL Receiver Will Process At a Time	If your implementation has configured multiple MPL servers , indicate the number of records that each MPL receiver should process.	Not currently used
Schema Directory	The full path of the virtual directory where XML schemas are stored. For example: http://localhost/cisxai/schemas. If this option is not specified, the XAI uses the current directory, from where it is being run, to locate schemas.	schemaDir
Schema Validation Flag	Enter Y to turn on schema validation for outbound messages . Enter N to turn this off.	xaiSchemaValidationCheck
Send SOAP Fault as HTTP 500	Enter Y to ensure that a SOAP error is reported as an HTTP 500 "internal server error".	sendErrorAsHttp500
Sender Retry Seconds	This value is required if you have enabled Automatic Resend . It defines how many seconds to wait after marking a sender unavailable before you mark the sender available again (and retry sending messages to it).	senderWaitTime
System Error JDBC Connection	When a request fails to execute due to a system error, the MPL retries its execution several times. The MPL registers the system error in a table and uses this table for the retries. This setting specifies the JDBC connection required to access this table. Only enter a value in this field if it is different from the database environment used to read the XAI registry.	systemErrorTableJDBCConnection
System Error Max Retry	When a request fails to execute due to a system error, the MPL retries its execution several times until a maximum number of retries is reached. This option specifies the maximum number of retries.	systemErrorMaxRetries
System Error Retry Interval	When a request fails to execute due to a system error, the MPL retries its execution several times. This option specifies the number of seconds the MPL server waits between retries.	systemErrorRetryInterval

Thread Pool Initial Size	The MPL uses a thread of pools to enhance performance. The MPL starts with a minimum number of threads and grows/shrinks the pool based on the MPL system load. This option specifies the initial number of threads in the thread pool. The minimum number of threads is 12.	threadPoolInitialSize
Thread Pool Max Size	This option specifies the maximum number of threads in the thread pool.	threadPoolMaxSize
Thread Pool Non Activity Time	This option specifies how long a thread in the pool may be inactive before it is timed out and released from the pool.	poolNoneActivityTime
To Do Type for Inbound JMS Message Errors	To Do type for inbound JMS message errors. The inbound message processor uses this To Do type when creating To Do entries for inbound JMS messages that cannot be successfully processed. The system provides the To Do type F1-INJMS that may be used here.	toDoTypeforInboundJMSMessageErrors
To Do Type for Outbound Message Errors	To Do type for outbound message errors. The outbound message receiver uses this To Do type when creating To Do entries for outbound messages that cannot be successfully processed. The system provides the To Do type F1-OUTMS that may be used here.	outboundErrorTodo
WSDL Service Address Location	Specifies the SOAP address location that XAI uses in creating a WSDL. If no value is present, the XAI's URL is used.	wsdlAddressLocation
XAI Authentication Password	The multi-purpose listener uses this field in combination with the XAI Authentication User when attempting to communicate with the XAI server over HTTP, which is running on a secured servlet and requires authentication.	HTTPBasicAuthPassword
XAI Authentication User	The multi-purpose listener uses this field in combination with the XAI Authentication Password when attempting to communicate with the XAI server over HTTP, which is running on a secured servlet and requires authentication.	HTTPBasicAuthUser
XAI Trace File	The full path name for the file, where the XML messages should be written. For example: c:\inetpub\wwwroot\cisxai\xai.log.	traceFile
XAI Trace Type	Use this option to specify the level of logging. The possible values are FULL - All XML messages are written to the log file and NOLOG - No information is written to the log file. FASTPATH: Refer to Server Trace for more information about tracing.	traceType
XSL Directory	The full path of the virtual directory where XSL transformation scripts are located. XSL transformation scripts can be defined for each service. By default, this is the same directory as the schemas directory.	XSLDir

Where Used

Used by the XAI tool to obtain various required settings and locations.

Web Service Adapters

The base product provides a configuration object called Web Service Adapter that is used to help build configuration objects to allow for functionality in the system to initiate a web service call from within the system. A Web Service Adapter provides the following functionality:

- WSDL (web service description language) import. An implementer can use the WSDL import functionality to read the details of a WSDL into the system
- Internal API generation. The system generates internal data areas that have two main purposes: they provide the API for custom code to define the appropriate input and they provide output data for the web service call using Oracle Utilities Application Framework schema language. In addition, the web service dispatcher uses element mapping defined in the data areas to transform the internal XML into the structure expected by the external system as described in the WSDL.
- Defines the URL needed to perform the web service call at runtime.

Understanding Web Service Adapters

The following topics describe the system functionality in more detail.

Importing a WSDL

Configuring a Web Service Adapter starts by identifying the WSDL (the web service description language document used to define the interface) that will be provided by the external system. The following steps describe the base product functionality provided to allow a user to import a WSDL.

- Navigate to the **Web Service Adapter** page in add mode and select the appropriate base business object.
- Enter a meaningful Web Service Name and appropriate descriptions.
- Provide the URL of the given WSDL.
- Click **Import** to retrieve the details of the WSDL. The system then parses the WSDL details and populates the WSDL Service Name, WSDL Source, WSDL Port, URL and a list of Operations (methods) defined in the WSDL.
- Determine which Operations should be **active** based on the business requirements for invoking this web service. **Active** operations are those that the implementation is planning to invoke from the system. These require appropriate request and response data areas generated for them. The following section provides more information about that.
- Specify the appropriate Security Type to configure the type of security to use when invoking this web service.
- Click Save.

At this point, a web service adapter record is created in pending status. The next step is to generate the request and response data areas for the operations configured as active.

Generating Request and Response Data Areas

Each **active** operation for the web service adapter requires a pair of data areas, request and response, that represent the request and response XML messages for the operation.

The base product provides steps to generate the data areas as follows:

- As described in the Importing a WSDL section above, the operations listed in the WSDL are generated for the web service adapter and the implementer should indicate which operation to activate.
- After saving the **pending** web service adapter, the display lists all the active operations and for each one includes a **Generate** button.
- After clicking **Generate** for an operation, a window appears where the names of the new Request and Response Data Areas may be defined. Click **Save** to generate the data areas.

The generated data areas provide the API for the implementer to use when implementing the web service call in an appropriate algorithm or service in the system. The data areas contain the appropriate mapping from the elements that the implementer works within the code that invokes the web services and the WSDL definitions.

To facilitate generating the request and response data areas, the base product invokes a special business service used to create the appropriate mapping. The business service is defined as a BO option on the Web Service Adapter business object. This allows an implementation to provide a custom business service to further enhance the request and response mapping where appropriate.

NOTE:

Generated data areas. It is possible to edit and modify the generated data areas after they are created. An implementer can change element names or remove unneeded elements if desired. Manually changing the generated data areas must be done only when absolutely necessary. This is because the system is not able to validate manual changes and issues with the data areas would only be detected at run time.

Activating Web Service Adapters

The business objects provided by the base package for web service adapters include a simple lifecycle of **Pending** and **Active**. Configure the web service adapter and its data areas while in **Pending** status and activate it when it is ready to be implemented in the appropriate system functionality.

Invoking Web Services

To make a call to a web service using a web service adapter, the system has provided a Web Service Dispatcher business service (**F1-InvokeWebService**) to submit a web service call. The calling program is responsible for retrieving all the information to correctly populate the request data required by the web service call before invoking the business service.

NOTE:

Refer to the detailed description of the business service for more information.

Limitations

The following points highlight limitations associated with the types of web services that the system supports:

- It is possible for one WSDL document to contain definitions for several web services. The system currently supports only one port or service per WSDL document.
- It is possible for a WSDL to support multiple message patterns. The system currently supports only request / response.

Setting Up Web Service Adapters

Use the Web Service Adapter portal to define the configuration needed to communicate with an external system using a web service call. Open this page using **Admin > External Message > Web Service Adapter**. You are brought to a query portal with options for searching for web service adapters.

Once a web service adapter has been selected, you are brought to the maintenance portal to view and maintain the selected record.

The **Web Service Adapter** zone provides basic information about the web service annotation type.

Please see the zone's help text for information about this zone's fields.

FASTPATH: Refer to [Understanding Web Service Adapters](#) for information about common web service adapter functionality.

Sending Email

The framework provides the ability to initiate an email from within the system. The following topics highlight the functionality available.

- Sending email “real time”. The framework provides a business service **F1–EmailService** that supports sending an email. The schema supports elements for all the information required to create an email. It also supports sending attachments. The SMTP information (host, user name and password) may be provided or may be defined on a message sender, that may be provided as input. Review the business service schema for information about the input elements.

NOTE: Validating attachments. If a Validate Email Attachment algorithm is plugged into the [installation record](#), it is called to validate the attachments supplied, if applicable.

- Staging records to be emailed in “near real time”. This option may be used by options that are creating emails in bulk. This technique uses the outbound message table to capture the details of the email message.

XML Application Integration

This section describes the XML Application Integration (XAI) utility, which enables you to configure your system to receive information from and to send information to external applications using Extensible Markup Language (XML) documents.

NOTE: The XAI functionality is legacy functionality and not recommended for new implementations. The topics for the functionality outlined in the previous sections describe the recommended features for supporting sending and receiving external messages. The XAI information remains in the documentation for upgrade purposes.

The Big Picture of XAI

The XML Application Integration (XAI) module provides the tools and infrastructure that businesses require for integrating their applications with your product. The integration your product with other systems across organizational boundaries or business is made possible, regardless of the platforms or operating system used. XAI provides an integration platform that enables the following:

- Integrate with Customer Relationship Management (CRM) systems
- Provide information feeds for web based customer portals
- Fit seamlessly with web based applications
- Facilitate fast implementation of batch interfaces
- Integrate with other XML compliant enterprise applications

XAI exposes system business objects as a set of XML based web services. The service can be invoked via different methods, e.g., Hypertext Transfer Protocol (HTTP) or Java Message Service (JMS). Consequently, any application or tool that can send and receive XML documents can now access the rich set of system business objects. Business-to-Business (B2B) or Business-to-Consumer (B2C) integration with other enterprise applications as well as the setup of web portals is made very simple and straightforward.

XAI Architecture

The XAI architecture contains 3 major components:

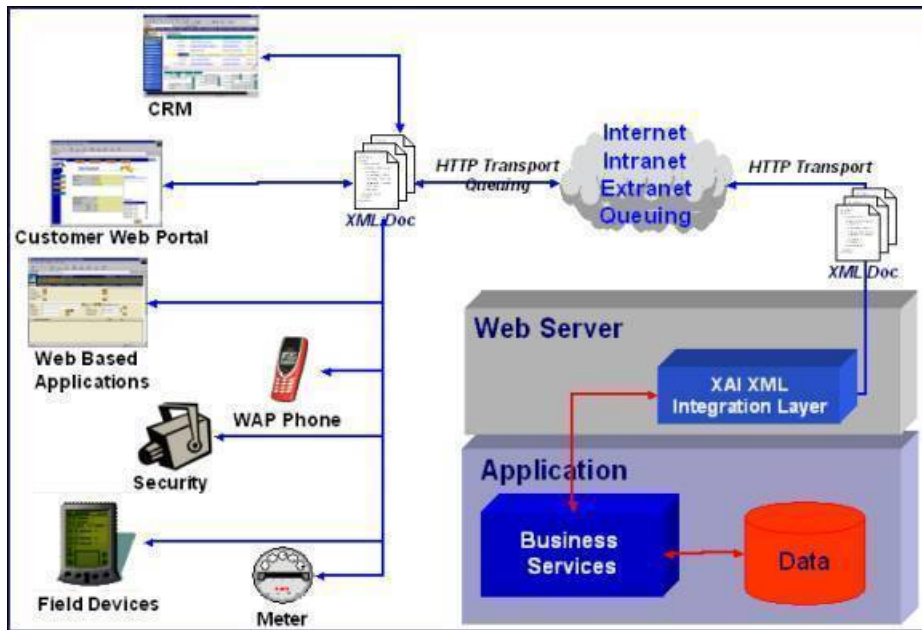
- The Core Server Component

- The Multi Purpose Listener (MPL)
- The Client Component

The Core Server Component

The core server component is responsible for receiving XML requests, executing the service and returning the response to the requester.

The following diagram shows the XAI tool operating on a web server and providing integration between the system business objects and various external applications.



The core is built in Java, using a layered, scalable architecture. The basic transport protocol supported by the core is SOAP/HTTP.

FASTPATH: Refer to [SOAP](#) for more information.

The XAI core server provides a Java servlet to process requests through HTTP. You may also use other messaging mechanisms such as message queuing, publish/subscribe or a staging table. The multi-purpose listener processes the messages.

The Multi Purpose Listener (MPL)

NOTE: Multi Purpose Listener functionality is legacy functionality that is not recommended going forward. The Oracle Service Bus (OSB) is the recommended tool. This section remains in place for upgrade purposes.

The Multi Purpose Listener (MPL) is a multi-threaded Java server that constantly reads XML requests from various external and internal data sources, such as a Java Message Service (JMS) message queue, a JMS topic or system staging tables.

The MPL can be used to process inbound messages (those sent by an external application to invoke a system service), or outgoing messages (those sent by your product to external applications). The MPL uses different receivers to process messages from different data sources.

A receiver is implemented using 3 distinct layers:

- The Receiving Layer

- The Execution Layer
- The Response Layer

The Receiving Layer

This layer deals with polling various locations to determine if new records, files or incoming requests exist. The various locations include:

- Staging tables, including [XAI staging control](#), [XAI upload staging](#), notification download staging (certain products only), and [outbound message](#).
- An external directory that contains a file, for example a comma delimited file or an XML file.
- A JMS queue/topic.

A separate receiver is defined to read requests from each of these locations. When the MPL server starts, it looks for all defined active receivers in the [XAI Receiver](#) table, and for each receiver it starts a thread that constantly fetches messages from the message source.

Once a request message is read, it is passed to an execution thread that implements the execution layer. Each receiver references an [Executer](#) that is responsible for executing the request.

Configuring Multiple MPL Servers

A single MPL server may only run one of each of the above staging table receivers for a given JDBC connection. To enhance the performance of the processing of the staging tables, you may define multiple MPL servers where each one runs the active receivers defined in the receiver table.

To ensure that each staging table receiver processes its own set of records in the staging table, the receiver selects a set number of records (specified as [Message option](#) **Number of Records an MPL Receiver Will Process At a Time**) and marks those records with the IP address and port number of the MPL.

The Execution Layer

The execution layer sends the XML request to the [XAI core server](#) and waits for a response.

NOTE: Currently the only type of [executer](#) supported is the XAI servlet running either on an XAI server or locally under the MPL. However the architecture allows for executing a request on other execution environments.

The executer is invoked and is passed in an [XAI Inbound Service](#) that specifies an XML request schema and an [adapter](#). Adapters tell XAI what to do with a request. The adapters point to a specific Java class that renders a service.

For example you can configure an Adapter to invoke any published application object (by pointing it to the appropriate java class). This adapter accesses system objects through the page service. You can think of an adapter as a plug-in component.

Once the executer processes the request and a response is received, it is transferred to the next layer, the [response](#) layer.

The Response Layer

The response layer is responsible for "responding" to the execution. The responses are handled by invoking an appropriate [sender](#) defined on the receiver's response information. Each sender defined in the system knows how to process its response. For example:

- The JMS queue sender and the JMS topic sender post responses to the appropriate queue / topic.
- The [staging control sender](#) handles errors received during the execution of the staging control request.
- The [upload staging sender](#) updates the status of the upload staging record based on the success or failure of the staging upload request.
- The download staging sender is a bit unusual because it is helping to build the message being sent (Oracle Customer Care and Billing only).

NOTE: There are some cases where a response is not applicable. For example, the file scan receiver creates a staging control record to process a file that exists in a directory. There is no "response" applicable for executing this request.

The XAI Client Component

The XAI Client component is the set of online control tables and tools used to manage your XAI environment.

The [Schema Editor](#) is a tool used to create and maintain XAI schemas.

FASTPATH: Refer to [XAI Schema](#) for more information.

The **Registry** is a term used to refer to all the tables required to "register" a service in the system. It includes the [XAI Inbound Service](#) and a set of control tables defining various options.

The [Trace Viewer](#) is installed with your XAI client tools and is used to view traces created on the XAI server.

XML Background Topics

The following section introduces some background information related to XML.

XAI Schemas

NOTE: Business Adapter. The **BusinessAdapter** adapter supports communication to configuration tool objects: [business objects](#), [business services](#) and [service scripts](#) through their own schema API. When communicating to these objects, it is not necessary to create XAI schemas for the schemas associated with the objects. As a result, there is no need to use the XAI schema editor when defining [XAI Inbound Services](#) for this adapter.

At the core of XAI are XML web services based on XML schemas. XML schemas define the structure of XML documents and are the infrastructure of e-business solutions. They are used to bridge business applications and enable transaction automation for e-commerce applications. Industry standard schemas document common vocabularies and grammars, enhancing collaboration and standardization. Validating XML processors utilize XML schemas to ensure that the right information is made available to the right user or application.

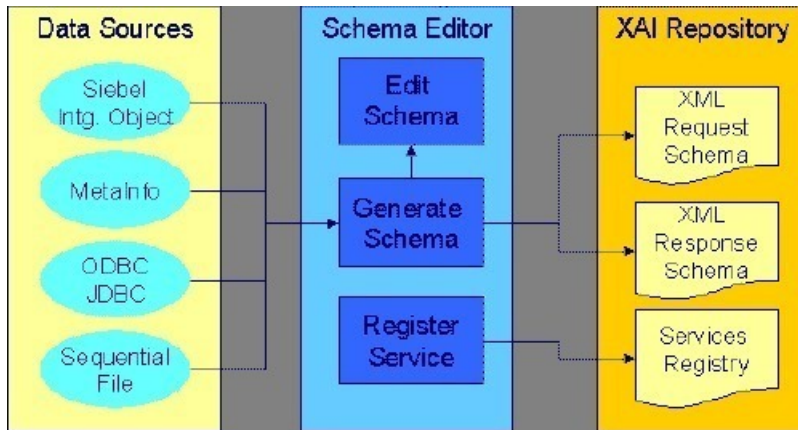
The system exposes its application objects as XML schemas that define the interface to system services. Every service (e.g., CreatePerson or AccountFinancialHistory) is defined using a pair of schemas: the Request Schema and the Response Schema. The request and response schema can be identical.

The Request Schema defines the XML document structure describing the "inputs" for a service.

The Response Schema defines the XML document structure describing the "outputs" of a service.

The Schema Editor

To facilitate the process of exposing business objects as XML schemas, we use the Schema Editor, a graphical tool to create, import and maintain schemas. The Schema Editor provides automated wizards to import schemas residing in existing data structures and documents. The Schema Editor can import schemas from the following sources: system business objects, ODBC data sources, sequential files.



Before the XAI tool can use a service, it must be registered or published.

FASTPATH: Refer to [Schema Editor](#) for more information.

XSL Transformations

XSL Transformations (XSLT) is a language used to transform an XML document into another XML document or another document format such as HTML. It is part of the Extensible Stylesheet Language (XSL) family of recommendations for defining XML document transformation and presentation. It is defined by the World Wide Web Consortium (W3C) and is widely accepted as the standard for XML transformations. Several tools are available on the market to generate XSLT scripts to transform an XML document defined by a source schema to an XML document defined by a target schema.

In XAI you can use XSL to:

- Transform an inbound message into the structure required by the XAI request schema for that service.
- Transform the response to an inbound message into a format defined by a schema provided by the requesting application.

FASTPATH: Refer to [XAI Inbound Service](#) for more information.

- Transform an outgoing message before it is sent out.

FASTPATH: Refer to [XAI Route Type](#) for more information.

- Transform data from an external source before it is loaded into the staging upload table.

FASTPATH: Refer to [XAI Inbound Service Staging](#) for more information.

SOAP

SOAP stands for Simple Object Access Protocol. The SOAP "Envelope" is the wrapping element of the whole request document that may be used by messages going through the XAI tool.

The following diagram shows a simple XML request using the SOAP standard.


```

<SOAP-ENV:Envelope>
  <SOAP-ENV:Header>
    <CorrelationID>1234</CorrelationID>
    <SOAPActionVersion>1.2</SOAPActionVersion>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <CISAccount transactionType='Read'>
      <CISAccountService>
        <CISAccountHeader
          AccountID='1234'
        />
      </CISAccountService>
    </CISAccount>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

XPATH

The XML Path Language (XPath) is an expression language used by XSLT to access or refer to parts of an XML document. It is part of the XSL recommendations maintained by the W3C. XPath provides the syntax for performing basic queries upon your XML document data. It works by utilizing the ability to work with XML documents as hierarchically structured data sets.

In the following XML document, some examples of XPath references are:

- authors/author/name
- authors/*/name
- authors/author[nationality]/name
- authors/author[nationality='Russian']/name
- authors/author[@period="classical"]

```

<?xml version='1.0'?>
<authors>
  <author period="classical">
    <name>Sophocles</name>
    <nationality>Greek</nationality>
  </author>
  <author>
    <name>Leo Tolstoy</name>
    <nationality>Russian</nationality>
  </author>
  <author period="classical">
    <name>Plato</name>
    <nationality>Greek</nationality>
  </author>
</authors>

```

In the XAI tool, XPath is used to construct outgoing messages.

Server Security

XAI server security supports the basic HTTP authentication mechanism as well as web service security (WS-Security) to authenticate the user requesting service. When authenticating using WS-Security, the SOAP header contains the authenticating information.

The base package provides two XAI server URLs, one that uses basic HTTP authentication ('/classicxai') and another that supports both methods ('/xaiserver'). Regardless of which authentication method is practiced, it is the latter you should

expose as your main XAI server. The main XAI servlet gathers authentication information from the incoming request (HTTP or SOAP header) and further calls the internal ("classic") servlet for processing.

The "classic" XAI server security uses the basic HTTP authentication mechanism to authenticate the user requesting service. It assumes the requester has been authenticated on the Web server running the XAI servlet using the standard HTTP (or HTTPS) basic authentication mechanism. The authenticated user-id is passed to the application server, which is responsible for enforcing application security. This requires the system administrator to enable basic authentication for the Web server running the XAI servlet. To enable HTTP basic authentication, the XAI server '/classicxai' should be defined as a url-pattern in the web resource collection in the web.xml file. When the XAI server is not enabled for basic authentication, it transfers the user-id specified on the **Default User** [Message Option](#) to the application server.

By default, the system would always attempt to further authenticate using SOAP header information. This is true even if the request has already been authenticated via the Web server. Use the **Enforce SOAP Authentication** [Message Option](#) to override this behavior so that a request that has been authenticated already by the Web server does not require further authentication by the system.

If SOAP authentication information is not provided, the system attempts to authenticate this time using information on the HTTP header. You can force the system to solely use SOAP authentication using the **Attempt Classic Authentication** [Message Option](#).

Currently the system only supports the standard *Username Token Profile* SOAP authentication method where "Username", "Password" and "Encoding" information is used to authenticate the sender's credentials. The following is an example of a *Username Token Profile* in a SOAP header:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV = "urn:schemas-xmlsoap-org:envelope">
<SOAP-ENV:Header xmlns:wsse="http://www.w3.org/2001/XMLSchema-instance">
<wsse:Security>
<wsse:UsernameToken>
<wsse:Username>MYUSERID</wsse:Username>
<wsse:Password Type="PasswordText">MYPASSWORD</wsse:Password>
</wsse:UsernameToken>
</wsse:Security>
<SOAPActionVersion>2.0.0</SOAPActionVersion>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
...
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

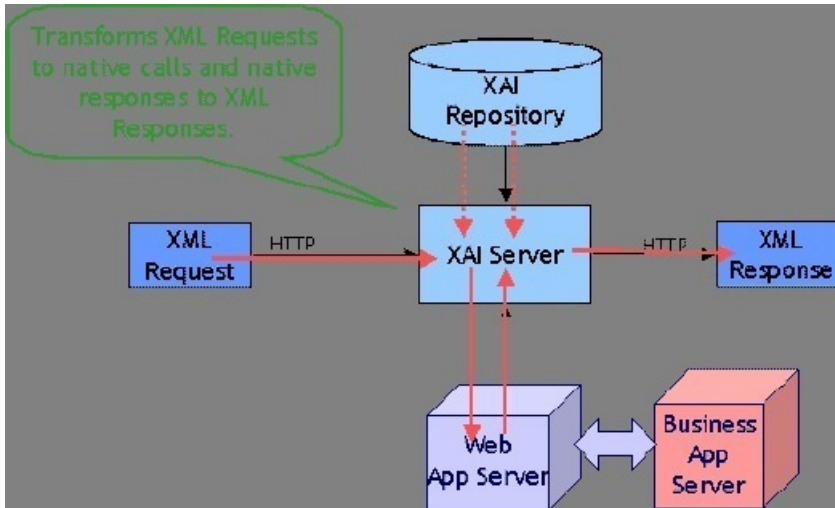
By default both user and password are authenticated. You can use the **System Authentication Profile** [Message Option](#) to change this.

NOTE: Custom authentication. You can override the base package user credentials authentication logic using the **System Authentication Class** [Message Option](#).

Inbound Messages

Inbound messages are XML messages sent by an external application to invoke a system service. Inbound messages can be sent one at a time or in batches of several messages in a file. Third party integration points can import web services for inbound service calls. The standard method of doing this is by publishing a WSDL (Web Service Definition Language) definition for each exposed service.

Synchronous Messages



Synchronous service requests are sent using the HTTP protocol to the XAI servlet, which runs under a web application server such as WebLogic.

- The service request XML document must conform to the request schema that defines the service.
- The XAI servlet on the web server receives the service request XML document and based on the service name in the document identifies the appropriate XAI Inbound Service. Once the service is identified, the XAI servlet accesses the XAI Inbound Service record to find out the properties of the service.
- Based on the service properties, the XAI module loads the request and response schemas from the schemas repository (and caches them in memory).
- Based on the Adapter referenced by the service, it calls the appropriate adapter. The adapter performs most of the work to service the request.
 - The adapter connects to the application server, based on the connection information in the registry control tables.
 - It then parses the request document using the request schema.
 - Once the request document has been validated, the adapter converts the XML request document into a call to the application server.
- The response returned by the application server is then converted into a service response XML document, based on the response schema.
- The XML response document is shipped back to the caller over HTTP.

Using HTTP for Synchronous Service Execution

Invoking a service is not much different from sending a regular HTTP request. Here the HTTP request contains the XML as a parameter. The XAI server handling requests via the HTTP protocol is implemented using a Java servlet running on the web server.

Microsoft Visual Basic/C Example

Microsoft provides an easy way to send XML requests over HTTP. To send and receive XML data over HTTP, use the Microsoft XMLHTTP object

```
set xmlhttp = createObject("Microsoft.XMLHTTP")
xmlhttp.Open "POST", http://localhost:6000/xaiserver, false
xmlhttp.Send XMLRequest
```

```
XMLResponse = xmlhttp.ResponseText
```

Here **http://localhost:6000/xaiserver** is the URL of the XAI server. **XMLRequest** contains the XML request to be processed by XAI and **XMLResponse** contains the XML response document created by the XAI runtime.

Java Example

Java provides a very simple way to send a request over HTTP. The following example shows how a request can be sent to XAI from an application running under the same WebLogic server as the one XAI runs. In this example, we use the "dom4j" interface to parse the XML document.

```
import com.splwg.xai.external.*;
import org.dom4j.*;

String xml;
xml = "<XML request>";
XAIHTTPCallForWebLogic httpCall = new XAIHTTPCallForWebLogic();
String httpResponse = httpCall.callXAIServer(xml);
Document document = DocumentHelper.parseText(httpResponse);
```

Asynchronous Messages

Various types of [receivers](#) running under the MPL server (rather than the XAI server) handle asynchronous inbound messages from several sources.

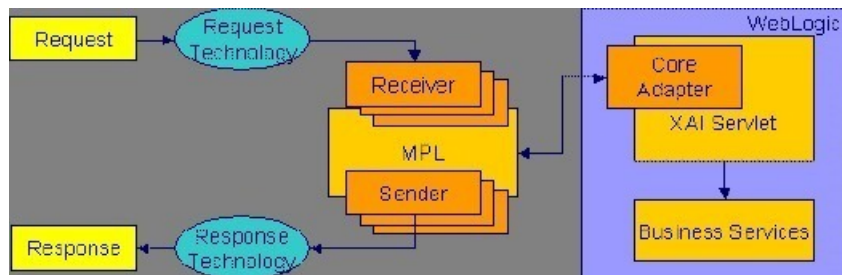
Requests may be received via the following:

- MQSeries, TIBCO or any JMS compatible messaging system
- The XAI Upload Staging table

The response is returned to the JMS queue/topic or to the staging table.

FASTPATH: Refer to [Designing XAI Receivers](#) for more information about the different receivers provided by the product for the different data sources.

The following diagram shows the flow for asynchronous inbound messages.

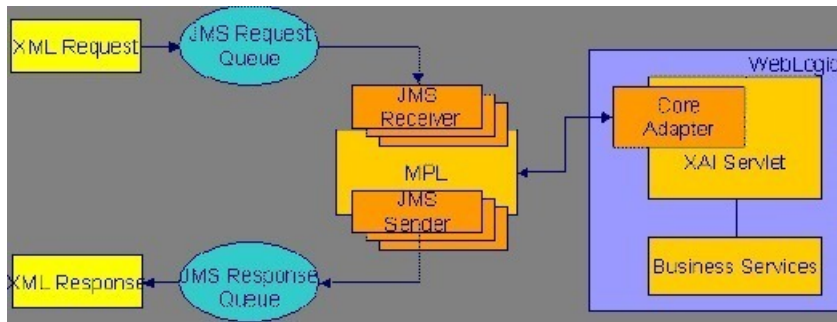


Using JMS

Java Message Services (JMS) is a standard Java 2 Platform, Enterprise Edition (J2EE) protocol to send/receive asynchronous messages using a queue or topic message protocol.

XML messages may be received and sent via JMS using either a JMS Queue or JMS Topic. In order to access a JMS provider such as MQSeries, TIBCO or WebLogic, the MPL must first connect to the appropriate server using a JMS Connection.

The following diagram depicts a message sent and received through a JMS Queue.



Using JMS Queues

JMS Queues are used to receive and send messages using the message queue protocol. Products that support this protocol include MQSeries.

The following describes events that take place when a JMS queue is used:

- The requester places the XML request message on a JMS queue. The request contains both the XML message and the name for the *reply queue*.
- The **JMS Queue Receiver** waits for messages on that queue. When the message arrives it is selected from the queue and executed using the adapter for the requested service.
- The XML response message is placed on the reply queue specified in the request. It is the requester's responsibility to fetch the response message from the queue.

The MPL uses a **JNDI server** to locate the queue resource.

FASTPATH: Refer to [Designing XAI JMS Queues](#) for more information.

Using JMS Topics

JMS Topics are used to send and receive messages using the publish/subscribe messaging model. Products that support this protocol include TIBCO, MQSeries and WebLogic.

The MPL uses a **JNDI server** to locate the topic resource.

- Define a **JMS Topic receiver**, listening to a predefined JMS Topic.
- The other application builds an XML message based on the schema defined for that service.
- It sends the XML request to the predefined topic and specifies the reply topic name.
- The MPL reads the message from the Topic, executes the service and returns the response to the reply topic specified in the inbound message.

FASTPATH: Refer to [Designing XAI JMS Topics](#) for more information.

Staging Upload

The system provides a staging table, where an interface can store XML requests to perform a service in the system.

Some external systems interfacing with the system are not able to produce XML request messages. Or you may have external systems that produce XML messages but the messages are sent in a batch rather than real time. The system provides the capability to read an external data source containing multiple records, map the data to an XML request and store the request on the **XAI upload staging** table. These records may be in XML requests, sequential input files or database tables.

The XAI upload staging table may be populated in one of the following ways:

- When a collection of messages in a file or database table must be uploaded, a **staging control** record should be created for each file/database table. The **Staging Control Receiver** processes each file or database table and creates records in the XAI upload staging table for each message.

- The [XML File Receiver](#) creates records directly in the XAI upload staging table. Note, in addition, the XML File Receiver creates a staging control record to group together these records.
- XAI creates records in the XAI upload staging table for [inbound messages in error](#) that are configured to post the error.
- It is possible that when a response to a notification download staging message is received (Oracle Customer Care and Billing only), the response requires some sort of action. If this is the case, the system creates an XAI upload staging record (and an XAI staging control record) for the response.

Staging Upload Receiver

Once the XML requests are in the staging table, the [Staging Upload Receiver](#) reads the requests from the XAI upload staging table and invokes the XAI server (via the executor) with the appropriate XAI inbound service. Inbound service records typically point to the core [adapter](#) used to invoke system services.

Staging Upload Sender

The staging upload [sender](#) handles "responses" to the execution of the message in XAI upload staging. If the execution is successful, the sender updates the status of the upload staging record to **complete**. If the execution is unsuccessful, the sender updates the status to **error** and creates a record in the [XAI upload exception](#) table.

NOTE: Configuration required. The above explanation assumes that you have correctly configured your upload staging receiver to reference the upload staging sender. Refer to [Designing Responses for a Receiver](#) for more information.

Staging Control

The [staging control](#) table is used to indicate to XAI that there is a file or table with a collection of records to be uploaded. The special [Staging Control Receiver](#) periodically reads the staging control table to process new records.

The XAI staging control table may be populated in one of the following ways:

- To process a specific sequential input file, manually create a staging control record and indicate the location and name of the file and the XAI inbound service to use for processing. Use this mechanism to process ad-hoc uploads of files.
- If a file is received periodically, you may define a [File Scan Receiver](#), which periodically checks a given file directory for new files. The file scan receiver creates a new staging control record to process this file.
- The [XML File Receiver](#) processes a file containing a collection of XML messages to be uploaded. The XML file receiver creates a staging control record and creates records directly into the XML upload table. The staging control record is automatically set to a status of **Complete** and is used to group together the XML upload records. Refer to [Processing Staging Upload Records for a Staging Control](#) for more information.
- To upload records from a database, manually create a staging control record and indicate an appropriate XAI inbound service, which contains the information needed by the system to access the appropriate table. Use this mechanism to process ad-hoc uploads of files.

NOTE: To upload records from a database table, you must create a staging control record. There is no receiver that periodically looks for records in a database table.

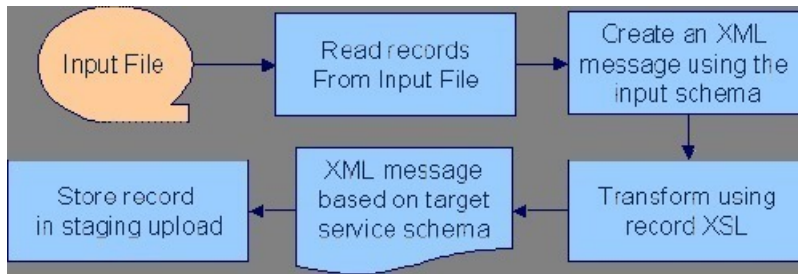
- Whenever an [XAI upload staging](#) record is created, an XAI staging control record is created as well.

FASTPATH: Refer to [Batch scenarios](#) for more information about configuring the system to populate the staging control with requests from external files.

Staging Control Receiver

The Staging Control [Receiver](#) processes staging control records and invokes XAI (via the executor) to execute the request. The executor uses the appropriate adapter to generate records in the [XAI Upload Staging](#) table - one for each record in the file or table.

The diagram below illustrates the information used by the staging control receiver to load data onto the staging table from a sequential input file.



The staging control [adapter](#) does the actual work. It reads the individual records in the input file and applies the XSL transformation script indicated on the XAI inbound service record to the input data to produce an XML request in the XAI upload staging table.

Processing Staging Upload Records for a Staging Control

In some cases, a process may populate records directly into the XML staging upload table. An example of such a process is the [XML File Receiver](#). In this case, a staging control record is also created and used to group together the staging upload records.

The staging control contains information needed to process a group of staging upload records:

- The user related to these records. This user is for application security purposes. The user indicated here must have the proper rights for the application service and transaction type to be executed by XML upload records processed for this staging control.
- An indication of whether or not the records should be processed [sequentially](#).

Processing Staging Records in Sequential Order

In some cases, a collection of messages uploaded together in a file must be processed in the order the messages are received. For example, if messages to add a person and add an account for this person are received together, the message to add the person must be processed before the message to add the account.

If messages received in a file must be processed sequentially, turn on the Sequential Execution switch on the [staging control](#) record. When the staging control receiver creates records in the XML upload table, the identifier of each record is built as a concatenation of the staging control record and a sequential number. If your staging control record indicates that the XML upload records should be processed in sequential order, the records are processed in primary ID order.

NOTE: Non-sequential processing. If your staging control does not indicate that the related XML records should be processed in sequential order, the records are processed by the staging control receiver using a random, multi-threaded mechanism.

If you have defined a [receiver](#) to periodically search for files and populate records in the staging control table, you may turn on the Sequential Execution switch on the receiver. This ensures that records processed as a result of this receiver are executed in sequential order.

Staging Control Parameters

If your staging control accesses the data from a database table, you have the capability of defining the selection criteria. [XAI inbound services](#) that reference the **CISStagingUpload** adapter may contain a collection of fields that are used in an SQL WHERE clause. When adding a new [XAI Staging Control](#) record, you can define the values for the WHERE clause.

For example, imagine that you have a work management system where new premises are defined. Rather than waiting for this system to "push" new data to you, you design the interface to have the system "pull" the new data by looking directly at the "new property" database table.

The Request XML for your XAI service contains SQL statements used to access the data. You could define a Staging Control Parameter of "Add Date". When creating your staging control, you may enter a parameter value of today's date. When this record is processed, it only retrieves new properties from this work management table whose Add Date is today.

The following example shows part of the Request XML schema for an XAI Service that SELECTs premises based on postal code.

```

- <Schema xmlns="urn:schemas-microsoft-com:xml-data" xmlns:dt="urn:schemas-microsoft-com:datatypes" xmlns:d="urn:schemas-microsoft-com:datatypes" xmlns:b="urn:schemas-microsoft-com: BizTalkServer" b:root_reference="TestDuplicatePremise">
- <ElementType name="TestDuplicatePremise" content="eltOnly">
  <b:property name="adapter">CISStagingUpload</b:property>
  <b:property name="stagingUploadType">DB</b:property>
  <element type="Request" />
  <element type="Response" />
</ElementType>
- <ElementType name="Request" content="eltOnly">
  <AttributeType name="Postal" d:type="string" d:maxLength="12" />
  <b:RecordInfo />
  <attribute type="Postal" />
</ElementType>
- <ElementType name="Response" content="eltOnly">
  <b:RecordInfo />
  <element type="Premises" />
</ElementType>
- <ElementType name="Premises">
  <b:RecordInfo />
  <b:property name="SQL">SELECT PREM_TYPE_CD, KEY_SW, OK_TO_ENTER_SW,
TREND_AREA_CD, ADDRESS1, MAIL_ADDR_SW, POSTAL FROM CISADM.CI_PREM WHERE
POSTAL = @TestDuplicatePremise/Request/Postal</b:property>
  <b:property name="tableName">CISADM.CI_PREM</b:property>
  <AttributeType name="PREM_TYPE_CD" d:type="string" d:maxLength="000">
    <b:property name="dataType">string</b:property>
    <b:property name="uniqueId">PREM_TYPE_CD</b:property>
    <b:FieldInfo />
  </AttributeType>
  <AttributeType name="KEY_SW" d:type="string" d:maxLength="001">
    <b:property name="dataType">string</b:property>
    <b:property name="uniqueId">KEY_SW</b:property>

```

The postal code value to substitute into the WHERE clause is defined on the individual XML Staging Control records.

FASTPATH: Refer to [XAI Staging Control](#) for more information.

Staging Control Sender

Before the staging control receiver invokes the executer, it changes its status to **complete**, assuming that there will be no problems. If the executer detects an error condition, the staging control sender updates the status of the staging control record to **error** and removes any XAI upload staging records that may have been created.

NOTE: Configuration required. The above explanation assumes that you have correctly configured your staging control receiver to reference the staging control sender. Refer to [Designing Responses for a Receiver](#) for more information.

Inbound Message Error Handling

For messages that are processed using the [staging upload](#) table, application errors that prevent XAI from successfully processing the message cause the staging record to be marked in error and highlighted via a To Do entry.

For messages that are not processed via staging upload, your implementation should consider what should happen to application errors. If the origin of the message is able to handle an immediate error returned by XAI, then no special configuration is needed. An example of this is an HTTP call to our system where the originator of the message is waiting for a real-time response.

Otherwise, for messages where errors should not be returned to the originator, but should be highlighted in this system for resolution, be sure to mark the **Post Error** switch on the [XAI inbound service](#). When this switch is turned on and an application error is received by XAI when processing the message an [XAI upload staging](#) record is created (along with a staging control record) and marked in **error**.

For example, if a message is received via a JMS queue, application errors that prevent XAI from processing the message should not be returned to the queue because there is no logic to route the error to the sending system.

Integration Scenarios

Integration Using an EAI (or Hub)

It is possible for your various systems to be integrated with each other using a hub. The hub is implemented using an Enterprise Application Integration (EAI) tool provided by a third party vendor. Most hubs support HTTP and/or JMS and can work with XML schemas or document type definitions (DTDs).

- XAI services are presented to the EAI tool as schemas or as DTDs, immediately making the system callable from the hub.
- Integration scripts or workflow processes are defined in the EAI tools.
- At run time the hub uses HTTP or JMS to access the system using inbound messages.
- Outgoing messages are used to notify the hub about events occurring in the system. The messages are sent using HTTP or JMS.

Batch Scenarios

Messages may be sent in batch files, or may be retrieved from a database. In all cases, the system needs to be able to read the file and identify each individual message in order to create an XML request that can be processed by the XAI server. Once each individual message is identified, a request is stored on the XAI Staging Upload table for later execution.

XML Message File

It is possible for you to receive a file containing a collection of XML messages. The system identifies each separate message within the file and creates an entry for each message on the XAI upload staging table. It also creates a staging control record to group together each newly created XAI upload record. This staging control is created in **Complete** status and is not processed by the staging control receiver.

Since external applications may send messages in a format unknown to XAI, the system needs a mechanism for identifying the messages and mapping them to an XAI service.

XAI Groups

First the system associates the entire XML file with an XAI Group. You can think of the XAI Group as a categorization of the collection of messages. For example, you may have a separate XAI Group for each third party who sends you a collection of XML messages.

The system uses an [XPath](#) and XPath value to identify the correct XAI Group for the XML file.

Attachments and Rules

After identifying the appropriate group to which an XML file belongs, the system takes each message in the file and applies the appropriate XSL transformation to the message to produce a record on the upload staging table.

To process the messages in a file, the system needs to know how to identify each message in a file containing multiple messages. A file may use the same root element for each message or different root elements for different types of messages. For each XAI group, you must indicate the root element(s) that identifies a message by defining one or more attachments. Each attachment defines a root element, which tells the system when a new message begins.

Once the system has identified each separate message in the file, it must determine the correct XSL transformation script to apply. Once again the system uses an [XPath](#) and XPath value to identify the correct XSL to apply. For each XAI group, you define one XAI rule for every possible type of message you may receive in the file. Each XAI rule defines an XPath, XPath value and XSL transformation script.

Note you may assign a priority to each of your rules. The rules for more common messages may be assigned a higher priority. This enhances performance by ensuring that rules for more common messages are processed before rules for less common messages.

NOTE: Include parent elements. If the XML message includes parent elements (such as a transmission id or a date/time) that are needed for any of the separate child messages that are posted to the upload staging table, you can configure the appropriate attachment to include **parent** elements.

FASTPATH: Refer to [Designing an XML File Receiver](#) for more information about defining receivers that process XML files.

Sequential Input File

You may receive messages in a sequential input file, such as a comma-delimited file.

The following steps should be performed when configuring the system to enable data to be uploaded from an input file into the staging upload table:

1. Create an XML Schema that describes the records in the sequential input file.
2. Create the XSLT transformation that maps a record in the input file to an XML service request in your product.
3. Create an [XAI service](#) representing the batch process that loads the input file into the staging table.
4. If desired, create a new file scan receiver, which waits for an input file to appear in a particular directory. (If you do not take this step, then you need to create a [staging control](#) when you want a file to be processed.)

NOTE: Character Encoding. If the file is encoded with a specific character encoding, you may indicate the encoding as part of the file name to be uploaded. If the file name ends with **?enc=?x**, where x is the file character encoding, the adapter processes the file accordingly. For example, the file name may be specified as **premiseUpload.csv?enc=?UTF-8**. If the encoding is not specified as part of the file name and the file is in UTF-16 or UTF-8 with byte order mark, then the adapter can recognize the encoding.

Database File

It is possible for you to define an interface where inbound messages are retrieved by reading records in a database table.

The following steps should be performed when configuring the system to enable data to be uploaded from a database table into the staging upload table:

- Create an XML Schema that describes the records in the database table.
- Create the XSLT transformation that maps a record in the database table to an XML service request in your product.
- Create an [XAI service](#) representing the process that loads the records from the database table into the staging table.

WSDL Catalog

Web Service Definition Language (WSDL) is a language for describing how to interface with XML-based services. It acts as a "user's manual" for Web services, defining how service providers and requesters communicate with each other about Web services.

The base package provides the ability to publish a WSDL definition for each service exposed as an [XAI Inbound Service](#). In addition, it is possible to request a catalog of all the XAI Inbound Services and a link to each WSDL. To view the catalog, launch a new browser session and enter the URL **http://\$host:\$port/XAIApp/xaiserver?WSDL** where \$host and \$port are replaced by the appropriate values for the current environment.

Outgoing Messages

This section describes outgoing message functionality related to MPL, which is no longer recommended.

- Outbound messages. As described in [Outbound Messages](#), outbound messages are supported for sending outgoing messages. This functionality includes support that is only related to MPL.
- Notification download staging (NDS) messages. This method is only supported by *Oracle Utilities Customer Care and Billing*. Using this method near real-time, a record is written to the NDS staging table referencing only key fields. MPL then polls the records, invokes a service to build the message, applies the XSL and routes the message. If sent real-time, no record is posted to the staging table but rather the message dispatcher routes the message immediately. Refer to Oracle Utilities Customer Care and Billing help, Workflow and Notifications, Notification and XAI for more information.

The following sections describe the outbound messages topics specific to MPL in more detail.

Outbound Message Receiver

The outbound message [receiver](#) processes records in the outbound message table that have a processing method flag equal to **XAI** and a status of **pending** and changes the status to **in progress**. The receiver then retrieves the message XSL and the message sender defined for the external system / outbound message type.

NOTE: Template external system. If the outbound message's external system references a template external system, the outbound message type configuration for the template external system is used.

It applies the message XSL (if supplied). If the option to [validate outbound message schemas](#) is turned on, the schema validation is performed.

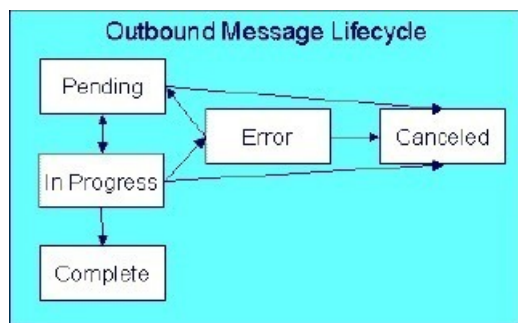
Refer to [Outbound Message Error Handling](#) for information about error handling.

If no errors are received, control is turned over to the [outbound message sender](#) for routing.

NOTE: Initialization of receiver. During the initialization of this receiver (for example if there is a problem with MPL and it is restarted) any records that are found to be **in progress** are changed to **pending** so that those messages are sent properly.

Lifecycle of Outbound Message

The outbound message receiver processes outbound message records based on their status. The following diagram describes the lifecycle of an **XAI** type outbound message.



- Records are created in **pending** status.
- The outbound message receiver processes pending records and changes the status to **in progress**.
- If the message is sent successfully the system changes the status to **complete**.
- If there was a problem sending the message the system changes the status to **error**.
- When the user resolves the error they can change the status back to **pending**.
- A user can change the status of a **pending** or **error** record to **anceled**.
- For the rare cases where there is a problem with MPL and a message is left in the status **in progress**, users may manually change the status to **anceled**. In addition the outbound message receiver includes a step at startup to find **in progress** messages and change them to **pending**.

Outbound Message Sender

The outbound message sender is responsible for routing the message to the [Message sender](#) determined by the receiver. If the routing is successful the outbound message status is marked **complete**. If the routing is unsuccessful, the status is marked in **error**.

Refer to [Outbound Message Error Handling](#) for information about error handling.

NOTE: Automatic Resend. If you have configured the system for [automatic resend](#) and the system detects that the error is due to the sender being unavailable, the message remains in **pending** status.

NOTE: Configuration required. The above explanation assumes that you have correctly configured your outbound message receiver to reference the outbound message sender. Refer to [Designing Responses for a Receiver](#) for more information.

Outbound Message Error Handling

If the outbound message receiver or the outbound message sender detects an error while attempting to process the outbound message, it marks the message in **error**, captures the error message and its parameter values and creates a To Do entry using the To Do type specified in the Message option **To Do Type for Outbound Message Errors**.

A separate background process [F1-DTDOM](#) is responsible for completing To Do entries for outbound messages no longer in **Error**.

Outbound Message Schema Validation

The outbound messages that are generated by the system should be well formed and valid so that they do not cause any issues in the external system. To ensure this validity you may configure the system to validate messages before they are routed to their destination.

- Define a directory where the valid W3C schemas are located using the Message option **Outbound Message Schema Location**
- Each [external system](#) message must indicate the appropriate W3C schema to validate against

You may turn on or off this validation checking using an Message option **Schema Validation Flag**.

Automatic Resend

If a system error is received by the MPL when attempting to route the message to a sender, (using the outbound message method or the NDS message method), the system marks the appropriate table in **error**. This is true even if the reason for the error is that the connection to the sender is unavailable. When the connection is restored, a user must change the status of the appropriate record to **pending** (for outbound messages) or **retry** (for NDS messages) in order for the message to be resent.

Alternatively, you can configure your system to attempt to automatically resend the message. This section describes the logic available for auto resend. To enable automatic resend, you must set the flag **Automatically Attempt Resending to Unavailable Senders** on [Message option](#) appropriately.

If an error is received by the MPL when it attempts to invoke a sender and the auto resend option is on, the system does not mark the record in **error**. It continues to attempt sending messages to the sender until the number of errors has reached a predefined maximum error number (defined as an [Message option](#)). When the maximum is reached, the sender is marked as **unavailable** and an MPL log entry is created. The MPL ignores messages for **unavailable** senders.

The system tries to resend messages to this sender the moment the sender is reset to be **available**. The following points describe how a sender becomes **available**:

- MPL attempts to retry sending messages to unavailable senders every X minutes, where X is defined on [Message option](#).
- On MPL startup all senders are marked as **available**
- A user may navigate to the [XAI Command](#) page and issue a command **MPL Refresh Sender** to refresh the cached information for a particular sender

Designing Your XAI Environment

This section guides you through the steps required to design the tables that control your XAI processing.

Installation

The XAI server is installed with default configuration. This section describes how you may customize the XAI server configuration.

Startup parameters are defined in two parameters files

- The XAIParameterInfo.xml file is used by the XAI HTTP server. This file is found within the XAI directory in the path (... \splapp\xai).
- The MPLParameterInfo.xml file is used by the MPL server. This file is found within the XAI directory in the path (... \splapp\mpl).

Both files store the parameters as XML files with the following elements (sections):

- Source
- ParameterVariables
- AdHocParameters

The XAI Source Section

The XAI tool accesses XAI [registry](#) information through the standard system programs. The <Source> section in the XAIParameterInfo.xml file tells XAI the user ID for accessing the registry information. It contains the following attributes:

Attribute Name	Description
Source Type	This should be set to CorDaptix , which tells XAI to access the registry through the standard access to system programs.
CorDaptixUser	The user ID to use when accessing the registry data.

The MPL Source Section

The <Source> section in the MPLParameterInfo.xml file defines the database connection information used to connect to the database storing the XAI table information. It contains the following attributes:

Attribute Name	Description
Source Type	Defines the source of the data, for example ORACLE or DB2 .
jdbcURL	The URL used to connect to the product database. For example: jdbc:oracle:thin:@//server-name:1234/DBNAME
databaseUser	The Oracle User Id used to connect to the database. For example: sysuser
databaseUserPassword	The Oracle password used to connect to the database. For example: sysuserpassword

The Parameter Variables Section

When defining values for fields in certain control tables in the registry, you may reference substitution variables that point to the <ParameterVariables> section of the installation files. Substitution variables provide for dynamic substitution of values based on parameters provided at server startup.

To specify a substitution parameter in a string value you enter the name of the substitution parameter enclosed with @.

For example if you have a field in the XAI control tables that should contain the URL for the XAI HTTP servlet, you could enter the value in the following way: **http://@HOST@:@PORT@/xaiserver**.

In the parameters section, define the appropriate values for these parameters, for example:

```
<ParameterVariables>
<ParameterVariable name="HOST" value="localhost" />
<ParameterVariable name="PORT" value="8001" />
</ParameterVariables>
```

At run time, the system builds the URL as `http://localhost:8001/xaiserver`.

Every substitution parameter is defined using an `<ParameterVariable>` element with the following attributes:

Attribute Name	Description
Name	The name of the substitution parameter
Value	The value to replace an occurrence of the substitution parameter in an XAI control table field

NOTE: Substitution parameters can only be used for string fields, and not for fields that are foreign keys to other objects.

The AdHoc Parameters Section

The `<AdHocParameters>` section is used to provide registry definitions that override the existing ones. Unlike the `<ParameterVariables>` section, a whole registry object definition can be specified in this section. When the XAI server starts, it first reads the registry definitions from the database and then it reads the `<AdHocParameters>` section. If it finds an object definition in this section, it uses it to replace the one read from the database.

Attribute Name	Description
Object Name	The object name may be one of the following objects: Option Receiver Sender
Object Attributes	The attributes of the object. Each object type has its own set of attributes:
'Option' object attributes	
name	The option flag. Must be defined in the OPTION_FLG table
value	The value for that option
'Receiver' object attributes	
name	The receiver ID
Class	The JMS provider. May be 'MQ'
TargetClient	The client type writing/reading to the JMS queue/topic. May be 'JMS' or 'MQ'. Only relevant for interfacing with MQSeries
JMSProvider	The JMS provider. May be 'MQ'
TargetClient	The client type writing/reading to the JMS queue/topic. May be 'JMS' or 'MQ'. Only relevant for interfacing with MQSeries
Executer	The XAI Executer ID for this receiver
'Sender' object attributes	
Class	The JMS provider. May be 'MQ'
JMSProvider	The JMS provider. May be 'MQ'
TargetClient	The client type writing/reading to the JMS queue/topic. May be 'JMS' or 'MQ'. Only relevant for interfacing with MQSeries

Designing XAI Inbound Services

When designing your XAI environment, you should first identify the services that you would like to perform. Determining your services facilitates your design for the other registry options.

To design your inbound services,

- Determine each service that needs to be performed
- Determine the correct [adapter](#) that is needed by your service.
- Determine the required layout of the request and response messages and specify the request [schema](#) and response [schema](#).
- If a transformation of the data is required, you need to design the appropriate request XSL and response [XSL transformation](#) scripts.
- If the service references the staging upload adapter, determine the staging file type and design the record XSL transformation script. In addition, determine if you want to enter an input file name and interface name. Finally, determine whether or not you need to indicate a special [JDBC connection](#).
- As new releases of the system are installed, it may be necessary to modify your service for the new release. If this is the case, you need to design separate versions of the inbound service.

FASTPATH: Refer to [XAI Inbound Services](#) for more information.

Designing XML Schemas

You need XML schemas for the services you designed in [Designing XAI Inbound Services](#).

For each message, identify what service you need to invoke and what action you need to perform. If you have multiple actions that you may need to perform for the same service, you may choose to create a single generic XML schema or you may choose to create multiple schemas, which are more specific. For generic messages, the transaction type, indicating the action to perform would be passed in on the XML request document to indicate what must be done. For more specific messages, you may be able to indicate the transaction type directly on the schema and it would not need to be overwritten at run time.

You need to create a response schema for each request schema. It is possible for you to use the same schema for both functions.

FASTPATH: Refer to [Schema Editor](#) for more information.

Designing XSL Transformations

You need an XSL transformation script for each service you designed in [Designing XAI Inbound Services](#), where you determined a transformation is necessary. In addition, you need XSLT scripts for your outgoing messages. Each sender, which receives a message, probably requires a transformation of the message into a local format. Refer to [Outgoing Messages](#) for more information.

For each message requiring transformation, determine the format used by the external system. In most cases, it is not the same format recognized by the system. For each case, you must create an XSL transformation, which maps the message format from the external format to one expected by your product or from your product format to one expected by the external system.

When identifying the required XSL transformations, remember to take into consideration the data that is processed by the staging control table. This service reads data stored in a file or database table and uses the Record XSL to map the individual records to an individual service request.

FASTPATH: Refer to [XAI Inbound Service](#) for more information.

Designing Your Registry Options

The XAI registry is a set of control tables that is used to store service definitions as well as various system information required by the XAI and MPL servers. The following sections describe each table in the registry.

Designing XAI JNDI Servers

The XAI tool, including receivers and senders, uses a Java Name Directory Interface (JNDI) server to locate resources on the network. JDBC connections, JMS connections, JMS queues and JMS topics should be defined on the JNDI server. In addition, adapters that need to access your product information reference the JNDI server to determine where your product is running.

Design your JNDI server values as follows:

- Define a JNDI server that indicates where the system is running.
- If you are using JMS resources, such as JMS Queue or JMS Topic, and these are located in a server other than the one where the system is running, define the server where these resources are located.
- If there are any other resources that may be needed by XAI, define a JNDI server to indicate where these are located.

The product is shipped with a JNDI server running under the same Weblogic server used for the system.

When defining the URL, you may use substitution parameters such as those shown in the example below. XAI uses the parameter variables section of your start up parameters to build the appropriate URL. Refer to [Installation](#) for more information

JNDI Server	Description	Provider URL
WLJNDI	Weblogic JNDI Server	t3://@WLHOST@:@WLPORT@

Designing XAI JDBC Connections

If you need to access a database table to process your messages, XAI needs to know the location of the database and how to access it. If the tables are located in the same database used for the system (defined in your [Installation](#)), then you do not need to enter any extra JDBC Connections. If you need to access data that lives in another database, design the additional JDBC Connections and determine the type of connection and connection information.

FASTPATH: Refer to [XAI JDBC Connections](#) for more information about defining XAI JDBC Connections.

Designing XAI JMS Connections

If you are using JMS to send and receive messages, then you must define a JMS Connection to indicate the JNDI server to use to locate these resources. For each JMS connection defined, the [MPL](#) server creates a pool of connections that are later shared by multiple threads.

FASTPATH: Refer to [XAI JMS Connections](#) for more information about defining XAI JMS Connections.

Designing XAI JMS Queues

If your business uses JMS Queues to send and receive messages, then you need to add an entry on the [XAI JMS Queue](#) page, defining the JNDI server and Queue Name.

Designing XAI JMS Topics

If your business uses JMS Topics to send and receive messages, then you need to add an entry on the XAI JMS Topic page, defining the JNDI server and Topic Name.

Designing XAI Formats

The Formats section of the registry is used to define data formats. Data formats can be used in schema definitions to specify data transformations. To determine what data formats you need to define for your XAI environment, you must review the expected format of data that you will be exchanging and determine whether or not data transformation is required.

The following sections describe the four different types of formats and some guidelines in their use.

Date Formats

Date formats may be specified using any valid Java format supported by the `java.text.SimpleDateFormat` class.

To specify the time format use a time pattern string. For patterns, all ASCII letters are reserved. The following usage is defined:

Symbol	Meaning	Presentation	Example
G	era designator	Text	AD
y	year	Number	1996
M	month in year	Text & Number	July & 07
d	day in month	Number	10
h	hour in am/pm (1-12)	Number	12
H	hour in day (0-23)	Number	0
m	minute in hour	Number	30
s	second in minute	Number	55
S	millisecond	Number	978
E	day in week	Text	Tuesday
D	day in year	Number	189
F	day of week in month	Number	2 (2 nd Wed in July)
w	week in year	Number	27
W	week in month	Number	2
a	am/pm marker	Text	PM
k	hour in day (1-24)	Number	24
K	hour in am/pm (0-11)	Number	0
z	time zone	Text	Pacific Standard Time
'	escape for text	Delimiter	
"	single quote	Literal	'

Currency Formats

Currency formats are used to specify formatting for elements representing currencies. They may include the following:

Symbol	Meaning
#	number place holder
,	thousands separator
.	decimal point
\$	currency sign

For example to define the currency format for US dollar, indicate: `$$,#.00`

Phone Formats

Phone formats can be used to specify formats for telephone numbers. The supported format specification is limited to the following format characters:

Symbol	Meaning
0	number place holder
\0	0

Any other character appearing in the formatting expression is a placeholder for that character. To specify the '0' character, use '\0'.

Phone Format Example: (000) 000-0000

Text Formats

Text formats are used to specify formats for character string attributes. The following expressions are supported:

Symbol	Meaning
\cUpperCase	Translate the string to upper case.
\cLowerCase	Translate the string to lower case.
\cProperCase	Translate the string to proper case. The first character of every word is translated to uppercase. The remaining characters are translated to lowercase.

FASTPATH: Refer to [XAI Format](#) to define your XAI Formats.

Designing XAI Adapters

The product provides a set of adapters to process your XML requests. The adapters point to a specific Java class that renders a service. If you find that you need to use a protocol, which is not supported by the adapters provided, you will need to add a new Message Class (which points to a Java class) and a new XAI Adapter. It is recommended that your implementers contact customer support. The following adapter classes are provided.

- **BASEADA:** This is the core adapter class that provides access to any published system service. This adapter accesses system objects through the page server. Services with this adapter need to indicate the object (application service), which should be invoked.
- **BUSINESSADA:** This is the core adapter class that provides access to schema-based objects. This adapter accesses [business objects](#), [business services](#) and [service scripts](#) through their schema API. Services with this adapter need to indicate the schema of the object, which should be invoked. When communicating to these objects, it is not necessary to create XAI schemas for the schemas associated with the objects. XAI is able to directly communicate with these objects using their existing schema definitions. As a result, there is no need to use the XAI schema editor when defining XAI Inbound Services for this new adapter.
- **STGUPADA:** This staging upload adapter class is used when an extra step is required prior to using a service with the core adapter. For example, perhaps you need to read a file, which is not in XML format, and convert it to an XML format recognized by the system. Services with this type of adapter do not need to indicate an application service but must indicate information about the file to be converted. Refer to [XAI Staging Control](#) for more information.
- **XAICMNDADA:** This is an internal adapter. It is used to send commands to the XAI Server.
- **LDAPIMPRTADA:** This adapter is no longer supported .
- **SIEBELADA:** This adapter is no longer supported.

Designing XAI Executors

The executor is responsible for executing messages received through a message receiver. The product provides an executor, which uses the XAI server; however the architecture allows for implementing additional execution classes. If you require a different executor and therefore a different execution class, it is recommended that your implementers contact customer support.

FASTPATH: Refer to [XAI Executer](#) for more information.

Designing Message Senders

This section only describes message sender functionality that is only related to the XAI/MPL processing. Refer to [Define the Message Sender](#) for details about other types of senders that are supported independent of XAI/MPL.

Message senders are responsible for define outgoing message destinations and for "responding" to the XAI executer.

- For NDS messages, the sender to use is defined on the [XAI route type](#) for the notification download profile.
- For outbound messages, the sender to use is defined on the [external system](#) / outbound message type collection.
- For responding to the XAI executer, the sender to use is defined on the [receiver](#).

For each sender, you must reference an appropriate Message Class. The information in this section describes the sender classes that are provided with the system.

You must create senders to "respond" to the various staging table receivers in the system.

- Create a sender to be used for "responses" to messages processed by the staging control receiver. You should create one sender, which points to the Message Class **UPLDERRHDLR**. Refer to [Staging Control Sender](#) for more information about this sender.
- Create a sender to be used for "responses" to messages processed by the upload staging receiver. You should create one sender, which points to the Message Class **STGSENDER**. Refer to [Staging Upload Sender](#) for more information about this sender.
- Create a sender to be used for messages processed by the download staging receiver. You should create one sender, which points to the Message Class **DWNSTGSNDR**. (DWNSTGSNDR is not supported in all products.)
- Create a sender to be used for messages processed by the outbound message receiver. You should create one sender, which points to the Message Class **OUTMSGSNDR**. Refer to [Outbound Message Sender](#) for more information about this sender.

Next, design the senders for "responses" to other receivers, for example the JMS queue receiver or JMS topic receiver. The system provides message classes to use for these senders. Use the class **JMSENDER** for a JMS queue sender and **TPCSNDR** for a JMS topic sender.

Finally, review all your [outgoing messages](#) and determine the mechanism for communicating with the target system for each message.

- For all senders that are used for [real time messages](#), define a context entry with a context type of **Response Time Out** to define the amount of time the system should wait for a real time response.
- An HTTP sender is one that sends messages to an HTTP server using the HTTP protocol. For an HTTP sender, reference a message class of **HTTPSNDR**. In addition, the context described in [Message Sender — Context](#) for the **RTHTTPSNDR** apply to this sender as well.
- An email sender allows for XML messages to be sent as email messages through an SMTP server. It can be used in notification download processes to send a response as an email message. The email sender supports standard email functionality such as "CCs" and attachments. The content of the email message is controlled by the XSL script defined in the [XAI route type](#) of the NDS message. The XSL script has access to all context records of the NDS message as well as the input XAI message that was created by processing the NDS. Reference a message class of **EMAILSENDER**. In addition, the context described in [Message Sender — Context](#) for the **RTHTTPSNDR** apply to this sender as well.
- If you want XML messages to be written to a flat file, use a flat file sender. For example, it can be used in the notification download process to write a response message to a flat file. Flat file senders should reference an Message Class of **FLATFILESNDR**. In addition, the following context records should be defined for senders of this type.

Context Type	Description	Values
Flat file output directory	Directory in the file system where to write the file	
Flat file filename pattern	The name of the output file. The file name may be a literal constant, or generated dynamically. To create a dynamic filename use <file name>\$\$ID, where \$\$ID is replaced at run time by the ID of the NDS message that triggered the response message. If no file name is defined for the sender, the XAI server generates a file name with the following format 'XAI\$\$ID.xai'.	
Append data to file	This parameter controls whether the content of the response message is appended to an existing file, or a new file is created (possibly replacing an existing one).	YES or NO
Character Encoding	Indicates if the message should be sent with character encoding. The sender will write the content of the file with encoding specified in the context value. If no value is specified, the sender uses the default Java system encoding, which is determined according to the operating system locale.	UTF-8 or UTF-16

The topics below describe configuration required for senders that route a message via an HTTP sender, a flat file sender or an email sender.

Designing XAI Groups

XAI groups are used by the system to process an XML file containing multiple messages to be uploaded into the system. One or more groups may be defined for an [XML file receiver](#).

FASTPATH: Refer to [XML Message File](#) for more information about how groups are used to process an XML file.

When setting up your XAI environment, identify the interfaces that require uploading an XML file containing multiple XML messages into the system through XAI.

First you need to categorize the XML files that you may receive. Define an XAI Group for each logical categorization. For example, you may want to define a separate XAI Group for each third party who may send you a collection of XML messages. Or, if all third party service providers send direct access messages in a standard format, you may want to define a single XAI Group for direct access messages.

For each group, you need to identify the root elements that indicate when a new message is starting. This collection of unique root elements for a group is called the attachments.

For each group, you must identify every possible message that may be sent. For every message, define an XAI Rule. The rule indicates the XSL transformation script to be executed along with the XPath and XPath value that the system uses to identify each message.

FASTPATH: Refer to [XAI Group](#) to define groups, their attachments and their rules.

Designing XAI Receivers

Receivers define small pieces of code that wait for requests to be received through various sources. Each receiver references an Message Class where the small piece of code is defined. The following receiver classes are provided:

- **STGRCVR** The receiver that references this class polls the XAI upload staging table for new inbound requests.
- **STGCTLR**CVR: The receiver that references this class polls the XAI staging control table for new upload processes.
- **DWNSTGRCVR**: The receiver that references this class polls the notification download staging table (NDS) for new messages. (Not available in all products).
- **OUTMSGRCVR**: The receiver that references this class polls the outbound message table for new messages.
- **JMSRCVR**: Receivers that reference this class receive requests through a message queue that supports the JMS Queue interface, such as IBM MQSeries.
- **TPCRCVR**: Receivers that reference this class receive requests through a publish/subscribe model, such as TIBCO, or any system supporting the JMS Topic interface.
- **FILESCANRCVR**: Receivers that reference this class poll a given directory for files with a given file name pattern.
- **XMLFILERCVR**: Receivers that reference this class poll a given directory for XML files with a given file name pattern.

Multiple receivers may be defined for these receiver classes. For example, the XML file receiver defines the scan directory. If you have multiple directories that contain files to be uploaded, define a receiver for each directory.

All types of receivers reference an Message Class and XAI Executer. If you require a new Message Class or Executer because you use a protocol that is not currently supported, it is recommended that your implementers contact customer support.

Designing Responses for a Receiver

Once a request has been sent for execution to the XAI server (via the executer), [the response layer](#) processes the response. For some receivers, a response may not be applicable. For example, a file scan receiver reads flat files in a given directory and posts records to the XAI staging control table. Responses are not applicable for this type of receiver.

The response may be conditional on the outcome of the request and may be sent to more than one destination ([sender](#)). To design your receiver responses, determine the conditions under which a response should be sent for each request processed by each receiver:

- Never send a response
- Send a response if the request was successful
- Send a response if the request was unsuccessful due to a system error
- Send a response if the request was unsuccessful due to an application error

Once you determine when to send a response, you must determine where to send the response. Responses for different conditions may be sent to different Message Senders or to the same Message Sender.

Designing Receivers that Poll Staging Tables

The following receivers are needed to poll the various system staging tables.

- Create a [staging upload receiver](#) that references the Message Class **STGRCVR**. For responses, **All Events** should reference the [staging upload sender](#).
- Create a [staging control receiver](#) that references the Message Class **STGCTLR**CVR. For responses, **All Events** should reference the [staging control sender](#).
- If your implementation uses the NDS message method of communicating outgoing messages, create a staging download receiver that references the Message Class **DWNSTGRCVR**. For responses, **All Events** should reference the download staging sender. NDS messaging is not supported in all products.
- If your implementation uses the [outbound message](#) method of communicating outgoing messages, create an [outbound message receiver](#) that references the Message Class **OUTMSGRCVR**. For responses, **All Events** should reference the [outbound message sender](#).

For all the above receivers, if you need to access multiple environments, simply create receivers for each JDBC connection. Note that you should not add more than one of each of the above receivers pointing to the same JDBC connection. To improve performance for a single JDBC connection you may [configure multiple MPL servers](#).

NOTE: Sample Data for Initial Install. When first installing the system, records for each of the above receivers are provided. Your implementation may use these records or remove them and create your own.

Designing a JMS Queue Receiver

If you need to receive messages through a JMS compatible queue, you need to define a JMS Queue receiver. When designing a JMS Queue receiver you first need to design a [JMS Connection](#) and a [JMS Queue](#).

If you would like to post [responses](#) back to the JMS queue, you may create an [Message sender](#) to send the response to the JMS queue.

Designing a JMS Topic Receiver

If you need to receive messages through a JMS Topic using the publish/subscribe model, you need to define a JMS Topic receiver, which receives messages published under a specific topic. When designing a JMS Topic receiver you first need to design a [JMS Connection](#) and a [JMS Topic](#).

If you would like to post [responses](#) through a JMS Topic using the publish/subscribe model, you may create an [Message sender](#) to send the response to the JMS topic.

Designing a File Scan Receiver

The file scan receiver constantly looks in a given directory for files with a given pattern. When it finds a matching file, it creates a record in the [staging control](#) table to upload the contents of the file into the [upload staging](#) table.

When setting up your XAI environment, identify the interfaces that require uploading a file from a directory into the system through XAI. For each unique file, define a file scan receiver. For each receiver record, indicate the Scan Directory, where new files will be placed, the Scan File, which is the naming pattern to look for and the XAI Inbound Service to use for mapping the data into a system service.

In addition, if you want to specify a character encoding, the following Context record should be defined.

Context Type	Description	Values
Character Encoding	Indicates that the message is character encoded. When the receiver creates a staging control entry for a file, it will add ?enc=?x to the name of the file in the table where x is the value of this parameter. Refer to Sequential Input File for more information.	UTF-8 or UTF-16

Designing an XML File Receiver

The XML file receiver constantly looks in a given directory for XML files with a given pattern. When it finds a matching file, it goes through steps to identify each separate message in the file, determine the appropriate XSL transformation and create a record in the staging upload table.

FASTPATH: Refer to [XML Message File](#) for more information.

When setting up your XAI environment, identify the interfaces, which require uploading an XML file containing multiple XML messages into the system through XAI. For each unique file, define an XML file receiver. For each receiver record, indicate the Scan Directory, where new files will be placed, the Scan File, which is the naming pattern to look for and the collection of XAI groups. XAI groups are used by the system to identify each separate message in the file and to determine the appropriate XSL transformation for each message.

FASTPATH: Refer to [Designing XAI Groups](#) for more information.

Configuring the System for NDS Messages

Refer to the documentation for your product to find out if NDS messaging is supported.

Schema Editor

The Schema Editor is a Graphical User Interface (GUI) tool to create XML schemas. The tool provides wizards to generate schemas from various sources.

Opening the Schema Editor

After launching the schema editor, you are asked to connect to a database. On the Connect dialog:

- Select an ODBC data source pointing to the desired database.
- Enter the data source, user ID, password and database owner required to log on to that database.
- Click **Connect**.

After connecting, the schema editor appears.

Use the File/Open dialogue to select a schema from the schema directory. Refer to [The Options Menu](#) for information about setting the default schema directory.

NOTE: When opening a schema, the schema editor validates the schema and any errors are displayed. Refer to [Validating a schema](#) for more information.

Schema Editor Window

The schema editor allows you to modify individual elements and attributes of a given schema.

Description of Page

Refer to [System Wide Functions for Schema Editor](#) for information about the various menu options available for the schema editor.

Service Name Enter the name of the service to be created in the service name text box. This is the name of the first element under the Body element in the XML document.

CAUTION: Important! When adding new schemas, carefully consider the naming convention for the Service Name. Refer to [System Data Naming Convention](#) for more information.

Adapter The adapter used to process services using this schema.

Internal Service Name If the schema is for an adapter that should invoke a system service, this is the internal name of the service.

Transaction Type Select the transaction type performed by the service. The available values are **Read, Add, Change, Update, Delete, List** and **Search**.

NOTE: The difference between Change and Update is that for Change, all field values must be passed in with the request. Field values that are not passed in to the request are set to null. For Update, you need only pass the primary key field values and the values of the fields to be updated. All other fields retain their existing values.

Left Panel

The left panel of the schema editor displays a tree view of the hierarchical elements in the schema. The (+) expands a node, the (-) collapses a node.

Right Panel

The following attributes appear on the right panel of the Schema Editor. Some fields cannot be modified in the schema editor. The field description indicates when the field is protected.

Tag Name The XML element tag name. This field is protected, but you may modify this attribute to give the element a self-explanatory name by right-clicking on the element name in the left tree-view.

MetaInfo Name Maps the element to a fully qualified field name in the service, for example PER_ID. This field is protected.

Internal Type This property is populated automatically when you generate the schema from your product. The values further define elements and attributes. The values are **page**, **pageBody**, **list**, **listHeader**, **listBody**, **searchHeader**, **codeTableDesc**, **Private**. The values of **codeTableDesc** and **Private** are used to define special types of attributes.

Private attribute A field that does not exist on the server side, but one that you still want to have in the schema.

Description A description of this field.

Content The element type. This field is only available for elements. Possible values are **eltOnly**- element may contain only other elements and no text, **TextOnly**- element may only contain text.

Search Type Services, which perform a Search, may allow searching based on different criteria. The values are taken from the system meta information when the schema is generated. The possible values are **Main**, **Alternate1**, **Alternate2**, **Alternate3**, **Alternate4**, **Alternate5** and **Alternate6**.

NOTE: You would not typically modify this value because it corresponds to a value in the meta information. However, the value is modifiable to accommodate the rare case in which a service may change in a new release. In this scenario, you may prefer to update the schema manually rather than regenerate a new schema for the new version.

Is Part of Primary Key Used to indicate to the XAI server whether or not this field makes up part of the primary key of the record. The values are taken from the metadata information when the schema is generated. Value may be **true** or **false**.

NOTE: Typically you would not modify this value because it corresponds to a value in the meta data information. However, the value is modifiable to accommodate the rare case where a service may change in a new release. In this scenario, you may prefer to update the schema manually rather than regenerate a new schema for the new version.

Min Occurs This field is available for elements only and is used for repeating elements. It defines the minimum number of occurrences for an element. Value may be 0 or 1.

Schema Max Occurs This field is available for elements only and is used for repeating elements. It defines the maximum number of occurrences for an element. Value may be 0, 1 or *.

Limit Number of occurrences This field is available for elements only and is used for repeating elements. If the **Schema Max Occurs** field has been set to '*', define the number of max occurrences here.

XML Data Type The data type for the attribute. Possible values are **number**, **string**, **decimal**, **date**, **dateTime**, and **boolean**.

Server Data Type Indicates the data type of this attribute on the server. This field is protected.

Service Format The format expected by the service. At runtime, XAI converts the Tag format to the Service Format before executing the request. Formats are defined in [XAI Format](#).

Tag Format The format used to format an element/attribute in the schemas. Formats are defined in [XAI Format](#).

Min Length Use this property to define the minimum length of the attribute, if applicable.

Max Length Use this property to define the maximum length of the attribute, if applicable.

Precision This is used for decimal attributes to define the maximum number of digits.

Scale This field is used for decimal attributes to define the number of digits at the right of the decimal point.

Required A value of **Y** indicates that the element must appear in XML document. A value of **N** indicates that the element is optional.

Default value Default value to be used for Request schema, when the element is not supplied as part of the XML request document.

Fixed Value Fixed value to be used for Request schema. This value is used regardless of the value supplied in the request document.

Code Table Field This property is used for attributes that are descriptions of a code table, where the description is not automatically returned by the system service. Use this property to indicate the code whose description should be retrieved by the XAI server.

Code Table Program This property is used for attributes that are descriptions of a code table, where the description is not automatically returned by the system service. Use this field to indicate the program that XAI should call to access the description for the **Code Table Field**.

Creating a Schema

Usually you do not create schemas from scratch; rather you use Schema Creation Wizards to import existing data structure definitions from a variety of data sources:

- System services
- Comma Delimited Files (CSV)
- Database Extract
- Any XML document

Once a schema is created based on the existing data structure, it is displayed in a TreeView on the left panel. Once the imported schema has been edited, it serves as the basis for creating the request and response schemas. When imported, the schema exposes all fields defined in the service. You may want to remove some attributes/elements from the request or response schema.

NOTE: Although the main purpose of the editing process in the creation of the request and response schemas is the elimination of elements, which makes the schema shorter and more understandable, it is not required for processing purposes. Therefore, if you don't mind that you have not used elements in your schemas, you could stay with one schema, which serves as both the request and response schema.

1. Save the Schema as a Request schema with an appropriate name, for example PersonInfoRequestSchema.xml
2. To create the Response schema, which is identical to the request schema, use the Save As Response menu option. This renames the top element of the schema to ServiceNameResponse, for example PersonInfoResponse and save the schema under a different name i.e. PersonInfoResponseSchema.xml. Note that if the request and response schemas are identical then one schema may be used for both and there is no need to create separate schemas.
3. Read in the Request Schema (File/Open) and modify its structure. Depending on the service type, you'll have to modify the contents of the Request Schema. This is usually required when the service is an "Info" service, which requires very few input elements. In such cases you'll delete most of the elements on the schema and only leave the necessary elements required to invoke the service. For example: in the PersonInfo request, you only need the PersonId and the Company elements in the request schema.
4. Read in the Response Schema (File/Open) and Modify its structure. Depending on the service type, you'll want to modify the contents of the Response Schema. This is usually required when the service is an "Add" or "Delete" service, which returns very few input elements. In such cases you'll delete most of the elements on the response schema and only leaves the necessary elements required by the requester of the service.

Adding an Element/Attribute

Usually, you won't have to add element or attributes to a schema. However if the schema already exists and you want to add an element/attribute, you can follow this procedure. Be aware that any element/attribute added here must also exist on the xml metainfo.

- Select the element's node on the TreeView.
- Right click on it and select the 'Add Element' or 'Add Attribute' option in the pop-up menu
- Enter the element/attribute name in the prompt dialog box and click OK.

Removing Elements/Attributes

When generating a schema using one of the wizards, the generated schema may contain information that you do not want to publish as part of the service, or is not required for a particular service. You can remove elements/attributes from the schema, and though these elements/attributes may still exist on the service they are not seen by the XAI service using this schema. To remove an element or attribute:

- Select the node to be removed.
- Right click and select "Delete" from the popup menu.


Renaming an Element

To rename an element:

- Select the element's node on the TreeView.
- Right click it and select the Rename option in the pop-up menu
- Enter the new name in the prompt dialog box and click OK.

NOTE: The information in this table is cached by the XAI server and by the MPL server. Changes to values in this table do not affect the runtime copy of these servers. Refer to [XAI Command](#) for information about refreshing a schema.

Validating a Schema

Although a schema is validated against the metainfo XML file when it is read into the editor or before it is saved, you can perform the validation at any time while the schema is being edited. To validate a schema, click on the toolbar "Validate" button . If the schema fails to validate the schema errors dialog is displayed.

Schema Validation Errors

When the editor fails to validate a schema against the xml metainfo file, it pops up a dialog that lists the errors found in the schema definition. These errors may be of two types:

- An element or attribute in the schema could not be found in the xml metainfo file. You can click **Remove** to remove the element from the schema.
- The data type of an attribute does not match the one defined in the metainfo file. You can click **Correct**, and the editor fixes the data type so it does match.

The **Correct All** button can be used to correct all fields that have data types that do not match the one in the XML metainfo file.

NOTE: Correcting errors does not save the schema definition into a file. You have to save it manually.

If you **Exit** without correcting the errors, the schema displays with the mismatch information highlighted in red.

Registering a Service

Before a service can be used it must be defined in the [XAI Inbound Service](#) table in the XAI registry. A service can be registered in the XAI registry directly from the schema editor. Go to the menu item 'Register Service' in the 'Schemas' menu. The Register service UI page appears. Fill in the required fields.

NOTE: The registry entry for the service can be later modified using the [XAI Inbound Service](#) page.

Testing a Schema

The schema editor provides a testing option.

NOTE: Testing in your product. You may also test your schemas and your services using either [XAI Submission](#) or [XAI Dynamic Submission](#).

System Wide Functions for Schema Editor

Because the schema editor is in an application outside of the standard products, this section introduces some general functions related to the application.

Application Standards

- The schema editor is a Multiple Document Interface (MDI) windows application. It may contain multiple active windows. You may jump from one window to the other using the Window menu option.
- The Editor window is always active. Closing the schema editor window quits the application.
- In the Editor window, a splitter bar is available to resize the schema tree view horizontally.
- Actions may be performed using menu items or by clicking on toolbar buttons.
- All messages are displayed on a status bar at the bottom of the screen. Some may also be displayed using message boxes.

The File Menu

Use the File menu to open existing schemas and to save a schema to a file.

Connect

Connects to the database.

Open - Loading an existing schema into the editor

You can read an existing schema into the editor.

1. Click on the Open toolbar button or select the File/Open menu option.
2. A file selection dialog is shown. Select the schema file name.
3. The editor first validates the schema against the xml metainfo file. If it fails to validate it shows the [Schema Validation Errors](#) dialog.

Save

Save the current schema to a file. Using the current file name.

To save a Schema, click on the Save toolbar button, or select the File/Save menu option. When you save a schema, the editor first attempts to validate the schema. If it fails to validate it against the XML Metainfo file, you are prompted to save it with inconsistency errors or to return to the editor.

Save As

Save the current schema to a file. Use a different file name.

Save As Response

Save a copy of the current schema to a file as a response schema. Use a different file name.

The View Menu

Use the view menu to perform actions on the Tree View nodes or to view errors. The following menu options are available:

Expand All

Expand all nodes in the Tree View

Collapse All

Collapse all nodes in the Tree View

Expand Branch

Expand the selected node and all the node's children

Search (Ctrl+F)

Find a node with a node name containing a given string

Search Again (F3)

Find the next node with containing the search string

View Schema Errors

Display the Schema Validation Errors dialog.

Web Browser

Display the current schema definition on a web browser page

The Schemas Menu

Use the Schemas menu to create, test, validate and register schemas.

- To create schemas from various sources use the **Create menu**.
- To [validate a schema](#) select the **Validate** option (Ctrl+V).
- To create a sample instance and test the schema, select the **Test** option (Ctrl+T).
- To create an entry in the service registry for a service represented by the current schema, select the **Register Service** option (Ctrl+R). Refer to [Registering a Service](#) for more information.

The Export Menu

The options on this menu are no longer supported.

The Options Menu

Always Save As W3C

Turn on this option to save schemas in W3C format by default instead of XDR format.

Always Save As DTD

Turn on this option to save schemas in DTD format by default instead of XDR format.

Preferences

The following options may be set on the preferences dialog (Select Options and then Preferences):

- The Default Date Time Format - used for schema date fields
- The default Schema Directory - used to read/save schemas

- The default Metainfo Directory - used to create/validate schemas
- The XAI Servlet URL - used when executing requests from the test schema dialog.
- The Default Account Id - used when generating sample instances of a schema
- Language - used to access language specific data

The Tools Menu

Schemas tools can be invoked from the **Schema Editor Tools** menu.

Converting Schemas to a W3C compatible schema

Schemas generated in the Microsoft BizTalk-compatible format (XDR format) may be saved in a format compatible with the *October 2000, W3C XML* schema standard. To save a schema in a W3C format:

- Select **Convert to W3C** from the **Tools** menu. The **Convert to W3C** dialog box appears.
- Select the schema(s) to be converted. Multiple schemas may be converted in a single step.
- The name of the W3C schema is the same name as the schema but with an ".xsd" file extension, and is saved to the same directory as the original schema.
- Click **Convert**.

Converting Schemas to a DTD

Schemas generated in the Microsoft BizTalk-compatible format (XDR format) may be saved as a DTD.

- Select the **Convert to DTD** option from the **Tools** menu. The **convert DTD** dialog box appears.
- Select the schema(s) to be converted. Note that multiple schemas may be converted in a single step.
- The name of the W3C schema is the same name as the schema but with a ".dtd" file extension, and is saved to the same directory as the original schema.
- Click **Convert**.

Validating multiple schemas

The schema validation tool can be used to validate the correctness of an XAI schema when compared to the metainfo xml definition used to generate the schema. For each validated schema, the validation tool scans the list of elements/attributes and compares them with those defined in the XML metainfo file. Select **Validate Schemas** in the **Tools** menu.

- The **Validate Schema** dialog box appears.
- The left list box is used to select the directory where the schemas to be validated are stored. By default this is the **Schema Directory** as defined in the preferences.
- On the right, the list of schemas is displayed on a grid.
- Multiple schemas may be validated in a single step.
- To select a schema click on the left button (the first column in the row).
- To select multiple schemas, hold the **Ctrl** key and select the required schemas.
- Click **Validate Schemas**.
- The schema grid is updated. When an attribute defined in the schema is missing from the metainfo file or when the properties of the element do not match defined in the metainfo, a button is displayed at the right of the schema name. Clicking on the **edit** button loads the schema into the editor and displays the **Schema Validation Errors** dialog for that schema. You can correct the schema and save it.

Setting Up Your XAI Environment

This section describes the control tables available to administer your XAI environment.

Message Class

The Message Classes are references to actual Java classes. The Message Class defines the Java class used to implement Receivers, Senders, Adapters and Executors. This information is provided with the system and does not need to be modified for a specific installation.

To view an Message Class, open **Admin > XAI > Message Class**.

Description of Page

The **Message Class** and **Description** are unique identifiers of the Message Class. The **Class Definition** indicates the Java class, implementing the adapter, receiver, sender or executor.

Owner indicates if this Message Class is owned by the base package or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add an Message Class. This information is display-only.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_XAI_CLASS](#).

XAI Envelope Handler

To view your envelope handlers, open **Admin > XAI > XAI Envelope Handler**. This information is provided with the system and does not need to be modified for a specific installation.

Description of Page

Enter a unique **XAI Envelope Handler ID** and **Description**.

Indicate whether the **Envelope Type** is **Custom SOAP Envelope**, **Default** (no SOAP environment) or **SOAP Envelope**. Note that the values of **Siebel Integration Message** and **Siebel VBC** are not supported.

When the envelope type is **SOAP Envelope**, indicate the **Envelope URI**.

Owner indicates if this XAI envelope handler is owned by the base package or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add an XAI envelope handler. This information is display-only.

Setting Up Your Registry

The following section describes the control tables that are logically considered part of the XAI Registry.

XAI JNDI Server

To define a new JNDI Server, open **Admin > XAI > XAI JNDI Server**.

Description of Page

Enter a unique **XAI JNDI Server** and **Description**.

Indicate the Provider URL, which is the URL of the [JNDI server](#).

Indicate the **Initial Context Factory**, which is a Java class name used by the [JNDI server](#) provider to create JNDI context objects.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_XAI_JNDI_SVR](#).

XAI JDBC Connection

To view an XAI JDBC Connection, open **Admin > XAI > XAI JDBC Connection**.

Description of Page

Enter a unique **XAI JDBC Connection** and **Description**.

Use the **Connection Type** to indicate how the JDBC connects to a database. The following connection types are valid:

- **Oracle Defined Connection** indicates the connection is to an Oracle database through a JNDI entry.
- **DB2 Defined Connection** indicates the connection is to a DB2 database through a JNDI entry.
- **JNDI Defined Connection** indicates the connection is using the MQ series classes implementing JMS.
- **Determined by parameter file** indicates that the connection information should be determined by looking at the parameters defined at [Installation](#).

For connection types of **Oracle** or **DB2**, use the **JDBC URL** to indicate URL of the database connection to be initialized at XAI/MPL startup time. Indicate the **Database User** and **Database Password** required for accessing the database. The JDBC connection URL can either be a Type 2 or a Type 4. For example:

- Type 2: jdbc:oracle:oci8:@CD200ODV
- Type 4: jdbc:oracle:thin:@myhost:1521/ CD200ODV

For a connection type of **Determined by parameter file**, indicate the parameter substitutions, which should be accessed from the parameter file for the JDBC URL, database user and database password, for example, @JDBCURL@, @DBUSER@ and @DBENCPASS@.

When the connection type is **JNDI**, indicate the **XAI JNDI Server** and the **JNDI Data Source** name as defined in the JNDI.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_XAI_JDBC_CON](#).

XAI JMS Connection

To define a JMS Connection, open **Admin > XAI > XAI JMS Connection**.

Description of Page

Enter a unique **XAI JMS Connection** and **Description**.

Indicate the **XAI JNDI Server** to be used. Refer to [XAI JNDI Server](#) for more information.

Use the **JNDI Connection Factory** to indicate the lookup keyword in the JNDI server used to locate the JMS connection.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_XAI_JMS_CON](#).

XAI JMS Queue

To define your JMS Queue values, open **Admin > XAI > XAI JMS Queue**.

Description of Page

Enter a unique **XAI JMS Queue** and **Description**.

Enter the **Queue Name** as defined in the JNDI server. This is the JNDI lookup name identifying the queue.

Use the **Target Client Flag** to indicate whether or not the target client is **JMS** or **MQ**.

Select the **XAI JNDI Server** where the queue is defined. Refer to [XAI JNDI Server](#) for more information.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_XAI_JMS_Q](#).

XAI JMS Topic

To define your JMS Topic values, open **Admin > XAI > XAI JMS Topic**.

Description of Page

Enter a unique **XAI JMS Topic** and **Description**.

Select the **XAI JNDI Server** where the topic is defined. Refer to [XAI JNDI Server](#) for more information.

Enter the **Topic Name** as defined in the JNDI server. This is the JNDI lookup name identifying the topic.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_XAI_JMS_TPC](#).

XAI Format

Open **Admin > XAI > XAI Format** to define the various formats.

Description of Page

For each new format, specify a unique **XAI Format** name and **Description**.

Indicate whether the Format Type is a **Currency formatting string**, a **Date/Time formatting string**, a **Phone formatting string** or a **Text formatting string**.

Finally, indicate the **Format Expression**, which defines the formatting pattern to be applied.

Owner indicates if this format is owned by the base package or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add an XAI format. This information is display-only.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_XAI_FORMAT](#).

XAI Adapter

To define a new adapter, open **Admin > XAI > XAI Adapter**.

Description of Page

Indicate a unique **Adapter Name** and **Description**.

Indicate the **Message Class**, which is the name of the Java class, implementing the adapter. The class should be one that is defined for an adapter. The adapter classes provided with the product are **BASEADA**- Core Adapter, **BUSINESSADA**- Business Requests Adapter and **XAICMNDADA**- XAI Command Adapter. Note that the values **LDAPIMPRTADA**, **SIEBELADA** and **STGUPADA** are no longer supported.

FASTPATH: Refer to [Message Class](#) for more information.

The following fields are not applicable for the **BusinessAdapter** adapter.

Use the **XAI JNDI Server** to indicate the name of the WebLogic JNDI server running your product. Refer to [XAI JNDI Server](#) for more information.

Indicate the **Default User** to be passed to your product server when this adapter is executed.

NOTE: If the XML request is sent over an HTTP connection, which has been authenticated, the authenticated User Id is passed to your product.

The **Default Date** format and the **Default DTTM** (date / time) **Format** specify date and date/time [formats](#) to use when a schema does not explicitly indicate formats.

Owner indicates if this XAI adapter is owned by the base package or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add an XAI adapter. This information is display-only.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_XAI_ADAPTER](#).

XAI Executer

To define a new Executer, open **Admin > XAI > XAI Executer**.

Description of Page

Enter a unique **Executer ID** and **Description**.

Indicate the **Message Class** for this executer. The class should be one that is defined for an executer. The executer class provided with the product is **XAIURLEXEC**- XAI Executer.

Indicate the appropriate **Executer URL**.

Owner indicates if this XAI executer is owned by the base package or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add an XAI executer. This information is display-only.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_XAI_EXECUTER](#).

XAI Group

XAI groups are used to process XML files, which contain a collection of [XML messages](#) to be uploaded in batch.

XAI Group - Main

To define your XAI groups, open **Admin > XAI > XAI Group**.

Description of Page

Enter a unique **Group** and **Description** for the XAI Group.

Indicate the **Parser** used for this group. Possible values are **Dom Parser** and **StAX Parser**.

NOTE: **Dom Parser** reads the full XML document into memory and therefore is not ideal for larger XML documents.

Indicate the **XPath** and **XPath Value**, which an XML file receiver uses to identify which group a given XML file belongs to.

- For **StAX Parsers** the XPath is limited to the root element.
- For **Dom Parsers**, the XPath supports defining elements at a lower level than the root element.

XAI Group - Attachments

Open **Admin > XAI > XAI Group** and navigate to the **Attachments** tab to define attachments for your group.

Description of Page

For each entry in the attachments collection, indicate the **Sequence** and the **Root Element**. Use **Include Elements** to indicate if **Parent** elements should be included along with the current element when applying the XAI rules.

FASTPATH: Refer to [XML Message File](#) for more information about how this is used.

XAI Group - Rules

Open **Admin > XAI > XAI Group** and navigate to the **Rules** tab to define rules for your group.

Description of Page

For each entry in the rules collection, indicate the **Sequence**, the **Priority**, the **XPath** name and **XPath Value** and the **XSL File Name**.

NOTE: Include Parent. If your attachment indicates that **Parent** elements should be included, be sure that the parent element is included in the XPath defined here.

FASTPATH: Refer to [XML Message File](#) for more information about how this is used.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_XAI_RGRP](#).

XAI Receivers

XAI Receiver - Main

To define your XAI receivers, open **Admin > XAI > XAI Receiver**.

Description of Page

Enter a unique **Receiver ID** and **Description** for the XAI Receiver.

Indicate the **Message Class** for this receiver. The class should be one that is defined for a receiver. The receiver classes are **DWNSTGRCVR**- Download Staging receiver, **FILESCANRCVR**- Upload Files from a directory, **JMSRCVR**- JMS Queue receiver, **OUTMSGRCVR**- Outbound Message receiver, **STGCTLRRCVR**- Staging Control receiver, **STGRCVR**- Staging Upload Receiver and **TPCRCVR**- JMS Topic receiver, **XMLFILERCVR**- XML File receiver.

FASTPATH: For more information, refer to [Designing XAI Receivers](#) about different types of receivers.

Indicate whether or not this receiver is currently **Active**.

Identify the **Executor ID**. Select the **XAILOCAL** executor if the Message Class for this receiver is **STGCTLRRCVR**. Select the **BYPASSXAI** executor if the Message Class for this receiver is **OUTMSGRCVR**. For all other receivers select the **XAIURL** executor. For more information, refer to [XAI Executor](#).

Indicate whether the **MSG Encoding** is **ANSI message encoding** or **UTF-8 message encoding**.

The **Read Interval** indicates the number of seconds between read cycles.

Start At Time and **Duration** are not currently in use.

If the Message Class for this receiver is **FILESCANRCVR**, **STGRCVR**, **STGCTLRRCVR** or **XMLFILERCVR**, indicate the **XAI JDBC Connection**.

FASTPATH: Refer to [XAI JDBC Connection](#) for more information.

Turn on **Sequential Execution** if the received requests should be processed in [sequential order](#) (instead of multithreaded). If this value is turned on then XAI staging control records created by this receiver are marked for sequential execution.

JMS Information

The following information is only available if the Message Class is **JMSRCVR** or **TPCRCVR**.

Indicate the appropriate **XAI JMS Connection**

FASTPATH: Refer to [XAI JMS Connection](#) for more information.

Indicate the appropriate **XAI JMS Queue**.

FASTPATH: Refer to [XAI JMS Queue](#) for more information.

Indicate the appropriate and **XAI JMS Topic**.

FASTPATH: Refer to [XAI JMS Topic](#) for more information.

File Information

The following information is only available if the Message Class is **FILESCANRCVR** or **XMLFILERCVR**.

Use the **Scan Directory** to indicate where to look for new files.

In **Scan File**, indicate the file pattern. All files with names matching the pattern are uploaded into the staging upload table. For each file found, a record in the staging control table is created.

CAUTION: WARNING. MPL expects all files conforming to the Scan File pattern to be complete. If a file is in the process of being copied into the scan directory and its name conforms to the naming pattern, MPL still attempts to process it and may issue an error for the incomplete file. It is suggested that files first be copied into the scan directory with a different name that does not conform to the naming pattern, for example filename.xml.inprocess. Once the file copy/transfer is complete, rename the file to one that conforms to the naming pattern, for example, filename.xml.

The following information is only available if the Message Class is **FILESCANRCVR**.

Use the **XAI In Service Name** to indicate how the records in the file are mapped and how they are transformed to match a system service request structure.

XAI Receiver - Context

Open **Admin > XAI > XAI Receiver** and navigate to the **Context** tab to define context for your receiver.

Description of Page

The Context collection enables you to define a collection of **Context Types** and **Context Values** defining. Use this collection when you need to store an attribute of a receiver that is not catered for in the current table.

NOTE: The values for the Context Type field are customizable using the Lookup table. This field name is RCVR_CTXTEXT_FLG.

XAI Receiver - Response

Open **Admin > XAI > XAI Receiver** and navigate to the **Response** tab to define where to send responses to requests made by this receiver. Refer to [Designing Responses for a Receiver](#) for more information.

Description of Page

The response collection enables you to define the destination (**Message Sender**) where responses to a request may be sent under various circumstances (**Event**). The events currently defined with the product are **All events**, **Message executed OK**, **Application Error**, **System Error**.

NOTE: The values for this field are customizable using the Lookup table. This field name is ON_EVENT_FLG.

XAI Receiver - Groups

Open **Admin > XAI > XAI Receiver** and navigate to the **Groups** tab to the valid XAI groups for an XML file receiver.

Description of Page

This collection is only available if the Message Class is **XMLFILERCVR**.

For each entry in the Group collection, indicate the **Priority** and the **Group**. Refer to [XAI Groups](#) for more information about defining groups.

Where Used

Receivers are used by the XAI server and by the MPL server to process messages sent to the system from various sources.

XAI Inbound Services

The XAI Inbound Services section in the registry is the main section of the registry. It is used to define the service characteristics. Basically, a service is defined by an Adapter responsible for executing the service, a pair of XML schemas and connection attributes. The Adapter defines the interface with the target application server, while the schemas define the structure of the request XML document expected by the service and the structure of the response XML document generated by the service.

XAI Inbound Service - Main

To update an inbound service, open **Admin > XAI > XAI Inbound Service**.

CAUTION: Important! When adding new inbound services, carefully consider the naming convention of the XAI In Service Name. Refer to [System Data Naming Convention](#) for more information.

Description of Page

Define a unique **XAI In Service Name**. This information is used in the system to identify the service. The service name is also the first XML element after the <Body> element in the XML request/response document. The system generates a unique **XAI Service ID**, which serves as the primary key.

Owner indicates if this XAI inbound service is owned by the base package or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add an XAI inbound service. This information is display-only.

Indicate the **Adapter**, which defines the interface with the target application server.

FASTPATH: Refer to [XAI Adapter](#) for more information.

If adapter for this service should invoke a system service, then indicate the appropriate **Service Name**.

FASTPATH: Refer to [Service Program](#) for more information about defining services.

If adapter is the base package **Business Adapter** then **Service Name** does not appear. Instead, use **Schema Type** to indicate the type of object this service invokes and **Schema Name** to reference the object to invoke. Using this adapter, you may set up service to invoke [business objects](#), [business services](#) and [service scripts](#).

FASTPATH: Refer to [Designing XAI Adapters](#) for more information about the **Business Adapter**.

Use the **Description** and **Long Description** to describe the service.

Check the **Active** switch if this service is enabled and available for execution. If this switch is unchecked, then the service is defined in the registry, but not yet available for public use.

Check the **Post Error** switch to support [inbound message error handling](#) for messages that are not processed via the staging upload table.

Check the **Trace** switch if you would like the trace to be on for this particular service. If the general trace option is not activated, you can force a trace for a particular service.

FASTPATH: Refer to [Server Trace](#) for more information about trace functionality.

When the **Debug** switch is checked, debug information is generated on the XAI console when this service is executed. The debug information can be useful to resolve problems.

Schema Definitions

NOTE: Request Schema and Response Schema are not applicable to services invoking schema-based objects. They do not appear when the **Business Adapter** is used.

The next two properties define the request and response XML schemas. The schemas were created using the [Schema Editor](#) and are SOAP compatible. The schema XML files are expected to be stored in the Schemas Directory on the Web server running the XAI server.

The **Request Schema** is the XML schema defining the service request. The request sent to the server must adhere to the schema definition.

The **Response Schema** is the XML schema defining the service response. The response generated by the XAI server corresponds to the response schema definition.

The same service may perform several actions on a business object. Use the **Transaction Type** to define the default action performed by a service. The transaction type can be provided when invoking a service, by dynamically specifying a transaction type attribute on the Service element of the XML request. This field may take the following values: **Read, Add, Change, Update, Delete, List** and **Search**.

NOTE: The difference between **Change** and **Update** is that for **Change**, all field values must be passed in with the request. Field values that are not passed in to the request are set to null. For **Update**, you need only pass the primary key field values and the values of the fields to be updated. All other fields retain their existing values.

Services, which perform a Search, may allow searching based on different criteria. When the Transaction Type value is **Search**, use the **Search Type** to define the default search criteria. The possible values are **Main, Alternate1, Alternate2, Alternate3, Alternate4, Alternate5** and **Alternate6**.

NOTE: This is a default definition only and it may be overridden at run time when the service is invoked. To override the search type at run time, you should specify the searchType attribute on the Service element of the XML request.

XSL Transformation Definitions

Sometimes, the XML request document does not conform to the request schema, or the response document expected by the service requestor is not the one generated by the adapter. In such cases the request and/or the response documents must be transformed. The XAI server supports transformation through XSL transformation scripts. Transformation scripts may be applied to the request before it is passed to the adapter or applied to the response document before it is sent to the service requestor.

The **Request XSL** is the name of the XSL transformation to be applied to the request document before processing it. The transformation is usually required when the incoming document does not correspond to the XAI service request schema therefore it has to be transformed before it can be processed by the adapter.

The **Response XSL** is the name of the XSL transformation to be applied to the response document when the requester of the service expects the response to have a different XML document structure than the one defined by the response schema for the service.

Click the **WSDL URL** hyperlink to launch a separate window that contains the WSDL definition for the inbound service. Note that the server name and port number for the URL are built using a setting in the common properties file using the XAI HTTP Caller URL setting.

NOTE: Refer to [WSDL Catalog](#) for information on how to obtain the WSDL catalog for all XAI Inbound Services.

XAI Inbound Service - Staging

The staging tab is used to define parameters for services that use the Staging Upload adapter.

FASTPATH: Refer to [XAI Upload Staging](#) for more information.

Open **Admin > XAI > XAI Inbound Service** and navigate to the **Staging** page to define attributes for your upload staging adapters.

Description of Page

Indicate the **Staging File Type** to be processed by the staging upload service. Possible values are **Comma Delimited file**, **Database Extract** and **Sequential file**.

The format of the records in the input file are not in an XML format and do not correspond to an XAI service schema. As a result, the input record must be transformed into an XML message that conforms to an XAI service request schema. Enter the **Record XSL**, which indicates the XSL transformation script used to transform the input record into the appropriate XML message.

For **sequential files** and **Comma delimited files**, indicate the **Input File Name** to be processed.

NOTE: This parameter can be overridden in the **Staging Control** table when a request to execute such a service is made.

When the service takes its input from a **Database extract**, indicate the **JDBC Connection** used to connect to the database that contains the input data.

NOTE: If this value is not populated XAI uses the default JDBC connection, which is the current product database.

FASTPATH: Refer to [XAI JDBC Connection](#) for more information about defining these values.

Use the **Interface Name** to provide a description of the interface being implemented through this service.

XAI Inbound Service - Parameters

This tab enables you to define parameters that are used as selection criteria by the DB Extract staging upload service.

Open **Admin > XAI > XAI Inbound Service** and navigate to the **Parameters** page.

Description of Page

The **Parameters** that were defined under the Request element in the schema are displayed here. They are used to drive the extraction process. This tab only displays the list of parameters. The values for these parameters can later be entered when the control record to invoke this service is created.

FASTPATH: Refer to [Staging Control Parameters](#) for more information.

Owner indicates if this XAI inbound service is owned by the base package or by your implementation (**Customer Modification**). The system sets the owner to **Customer Modification** when you add an XAI inbound service. This information is display-only.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [CI_XAI_IN_SVC](#).

NOTE: The information in this table is cached by the XAI server and by the MPL server. Changes to values in this table do not affect the runtime copy of these servers. Refer to [XAI Command](#) for information about refreshing a service.

XAI Route Type

Refer to the documentation for your product to find out if XAI Route Type is supported.

Running Your XAI Environment

NOTE: The XAI functionality is legacy functionality and not recommended for new implementations. This information remains in the documentation for upgrade purposes. Refer to [Incoming Messages](#) and [Outgoing Messages](#) for information about the recommended features for supporting sending and receiving external messages.

The [XML Application Integration](#) (XAI) module provides the tools and infrastructure that businesses require for integrating their third-party systems with the application.

This section describes the pages used to manage incoming and outgoing information via the XAI tool.

XAI Staging Control

Refer to [Staging Control](#) for more information about the purpose and functionality of this page.

Navigate to this page using **Menu > External Message > XAI Staging Control**.

Description of Page

The **XAI Staging Control ID** is a system generated unique identifier of this record.

The **XAI Staging Control Status** indicates the current state of this record.

Pending This status indicates that this record needs to be processed.

In Progress This status indicates that the background process, which uploads the staging records, is currently processing this record.

Error This status indicates that the background process found a problem with this record. When the data related to this record has been fixed, change the status back to **Pending** so that it will be processed again.

Complete This status indicates that this record has been processed.

The **Number of Uploaded Records** indicates how many XAI staging records related to this staging control record were uploaded.

The **Run Date Time** indicates the date and time this record was processed by the upload process.

The **User**, who created this staging control record, is captured.

If the data to be uploaded is found in a file, indicate the **Upload File Name**, which should include the directory path and file name.

NOTE: Character Encoding. Refer to [Sequential Input File](#) for information about specifying character encoding for a file.

NOTE: Complete File. MPL expects this file to be complete. If the file is in the process of being copied into the directory, MPL still attempts to process it and may issue an error for the incomplete file. The staging control record should only be created once the complete file is ready for upload.

Enter a description of the **XAI Interface Name**. This is simply information and is not used by the system.

Turn on **Sequential Execution** if you want the XAI upload staging records related to the staging control record to be processed in [sequential order](#) (instead of multithreaded).

The **XAI Service ID** tells the system how to create the XML related to the XAI upload staging record to be created. In Oracle Utilities Customer Care and Billing, for example, XAI Services with an adapter type of **CISStagingUpload** may be indicated here.

FASTPATH: For more information about XAI services, refer to [XAI Inbound Service](#).

Use the **Comments** to provide free format information related to this staging control record.

The collection of staging control parameters is used to define selection criteria when the staging control is accessing data from a database table. The **Staging Control Parameter** defines the field used in the WHERE clause and **XAI Staging Control Parm Value** defines the value to be used in the WHERE clause.

Refer to [Staging Control Parameters](#) for more information.

The **Message** collection displays any completion messages for XML requests related to the staging control record.

XAI Upload Staging

The XAI upload page allows you to view the details or to fix errors for a request in the Staging Upload table. This page displays the text of the XML request document linked to the upload.

XAI Upload Staging - Main

To view or modify an XAI upload document, navigate to **Menu > External Message > XAI Upload Staging** and navigate to the main tab.

Description of Page

The **XAI Upload Staging ID** is a system assigned identifier of the XAI upload staging record.

The **Application Service** identifies the internal system service called by the XAI tool to load/update the system data.

NOTE: The upload process does not use this. Rather, the application service is part of the XML Request. This field is used to help in searching for XAI Upload records.

The **XAI Staging Control ID** related to this XAI upload staging record is displayed.

The status of the upload is indicated by the **XAI Upload Staging Status**. Values are **Pending**, **In Progress**, **Complete**, and **Error**. If the record is in error, an error is written to the [XAI Upload Exception](#) table.

Create Date/Time indicates when the record was posted.

Completion Date/Time indicates when the work was completed.

If this XAI upload staging record is a response to an **XAI Download Staging** record, information about the related download staging record is displayed.

If there is an error with this record, you will see the error **Message** associated with this record. The message area is suppressed if there are no problems with the record.

Click the adjacent button to view the long explanation. The long explanation provides information about the cause of the error (and how to fix it).

The upload staging data appears in the **XML Request** text box.

XAI Upload Staging - Response

To view the response to a completed XAI upload staging document, navigate to **Menu > External Message > XAI Upload Staging** and go to the **Response** page.

Description of Page

The **XAI Upload Staging ID** is a system assigned identifier of the XAI upload staging record.

The system's response to the upload XML appears in the **XML Response** text box.

Uploading XAI Staging Records

Staging Control Layout

In order to load XML requests from flat sequential files, comma separated value (CSV) files and from database tables, you need to create a staging control record. The name of this table is **CI_XML_STG_CTL**. The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
XML_STG_CTL_ID	10	Y	N	This is the unique identifier of the record. This value does NOT have to be a random number, but it does need to be unique.
RUN_DTTM	26	N	Date/Time	Date and time this record was executed.
XAI_IN_SVC_ID	10	Y	A/N	This is the XAI service associated with this record. This service should have an adapter of CISStagingUpload .
XML_STG_FILE_NAME	250	N	A/N	This is the location and name of the file containing the data to be uploaded. This is used for comma delimited files and flat sequential files.
XML_INTFC_NAME	30	N	A/N	The name of the interface used to populate this table. This is simply information and is not used by the system.
XML_STG_STATUS_FLG	2	Y	A/N	Must be set to P (Pending) .
NT_XID_CD	30	N	A/N	This field is not in use.
NT_UP_XTYPE_CD	30	N	A/N	This field is not in use.
USER_ID	8	Y	A/N	The ID of the user who created this record.
COMMENTS	254	N	A/N	Use the free format comments, if necessary.
NBR_RECORD_UPLD	10	N	A/N	Leave this blank. This will be populated by the process which creates upload staging records for this control record.

Staging Control Parameters Layout

If your staging control record should read a database table, the XAI service may include selection criteria in its **WHERE** clause. If that is the case, then your staging control record should populate the selection criteria. The name of this table is **CI_XML_STGCTL_P**. The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
XML_STG_CTL_ID	10	Y	N	The unique identifier of the staging control record.

XML_STG_CTL_PARM	18	Y	A/N	Indicate the element in the WHERE clause whose value will limit the selection of records.
XML_STG_CTL_PVAL	250	Y	A/N	Use this to indicate the value used to limit selection of records.

Upload Staging Layout

You create an XAI upload staging record directly for each XML request you wish to make. The name of this table is **CI_XML_STGUP**. The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
XML_STGUP_ID	15	Y	N	This is the unique identifier of the record. This value does NOT have to be a random number, but it does need to be unique.
APP_SVC_ID	20	Y	A/N	This is the foreign key to the application service table, and identifies the application service that is being requested.
COMPLETE_DTTM	26	N	Date/Time	Completion date and time.
CRE_DTTM	26	Y	Date/Time	Creation date and time.
RETRY_COUNT	3	Y	N	Number of retries to process the request.
XML_REQUEST	30,000	Y	A/N	The XML request document.
XML_RESPONSE	30,000	Y	A/N	The XML response document.
XML_STG_CTL_ID	10	Y	A/N	The id of the staging control record linked to this record.
XML_STG_STATUS_FLG	2	Y	A/N	Must be set to P (Pending) .

XMLUP-PR - Purge XAI Upload Objects

Completed XAI upload staging objects should be periodically purged from the system by executing the **XMLUP-PR** background process. This background process allows you to purge all **Completed** XAI upload staging objects older than a given number of days.

We want to stress that there is no system constraint as to the number of **Completed** XAI upload staging objects that may exist. You can retain these objects for as long as you desire. However we recommend that you periodically purge **Completed** XAI upload staging objects as they exist only to satisfy auditing and reporting needs.

XAI Dynamic Upload

This page displays the XAI upload record. However, rather than displaying the XML Request as an XML document, it dynamically builds the screen with prompts and input fields to allow for changes without having to understand the XML. This screen is useful for fixing errors. Once the data is fixed, the status of the record can be changed back to **Pending** to once again attempt to upload this record.

NOTE: The Dynamic Nature of this Transaction. This transaction is designed to display the schema of the represented XAI upload staging. The system reads the XML schema associated with the upload XAI service and builds the tab(s) with the associated prompts and fields. For example, if the upload record is relevant to the Premise Maintenance service, Premise Characteristics and Premise Geo Types tabs are dynamically built.

XAI Dynamic Upload - Main

Navigate to this page using **Menu > External Response > XAI Dynamic Upload**.

Description of Page

The **XAI Upload Staging ID** is a system assigned identifier of the XAI staging record.

The **Application Service** identifies the internal system service called by the XAI tool to load/update the system data.

NOTE: This is not used by the upload program. Rather, the application service is part of the XML Request. This field is used to help in searching for XAI upload staging records.

The **XAI Staging Control ID** related to this XAI upload staging record is displayed.

The status of the upload is indicated by the **XAI Upload Staging Status**. Values are **Pending, In Progress, Complete, and Error**.

Create Date/Time indicates when the record was posted.

Completion Date/Time indicates when the work was completed.

If there is an error with this record, you will see the error **Message** associated with this record. The message area is suppressed if there are no problems with the record.

Click the magnifying button to view the long explanation. The long explanation provides information about the cause of the error (and how to fix it).

The **Transaction Type** indicates the action that should be performed for the application service when this document is uploaded. The valid values are **Add, Change, Delete, List, Read, Search** and **Update**.

Set the **Trace** option to **yes** to request level tracing to be executed inside the XAI tool. This will result in information written to a file, which may be useful in debugging. Refer to the XAI tools documentation for more information.

The bottom portion of the screen will contain field prompts and input fields for the data associated with the XML request linked to this upload record. The system dynamically builds this portion of the page by reading the XML request associated with the upload record. A user can change data in the displayed fields. This will result in a change to the XML request associated with the upload record.

NOTE: This page will also generate other tabs dynamically for any collections that exist for the service being displayed. These will vary based on the service. The response tab is the only other tab, which is always present.

XAI Dynamic Upload - Response

Navigate to this page using **Menu > External Message > XAI Dynamic Upload** and go to the response tab.

Description of Page

The **XAI Upload Staging ID** is a system assigned identifier of the XAI staging record.

The system's response to the upload XML appears in the **XML Response** text box.

XAI Upload Exception

A record is written to the XAI upload exception table for every XAI upload staging record that is in error.

To view the messages associated with the exception records, schedule the **TD-XAIUP** background process. This process generates a To Do entry for every record in the XAI upload exception table.

After correcting the cause of the error, drill into the [XAI Upload Staging](#) page and change the status from **Error** to **Pending** and the system will attempt to process the record again.

XAI Download Staging

XAI Download Staging - Main

To view individual XAI download staging records associated with a notification download staging record, open **Menu > External Message > XAI Download Staging**.

Description of Page

XAI Download Staging displays information about the record.

The **Download Staging ID** and **XAI Route Type** are the primary identifiers for this record.

The **XAI Download Status** is displayed. The values are **Pending**, **Complete**, **Canceled** or **Error**. When a record is in **Error**, it is displayed on the [XAI Download Exception](#) table.

Values from the NDS record associated with this XAI download staging record are displayed including **Service Provider**, **Notification Download Type**, **NT Download Status Flag**, **Retry Count** and the **Context** collection.

XAI Download Staging - Request

To display the XAI Request built by the download staging receiver, open **Menu > External Message > XAI Download Staging**.

Description of Page

The **XML Request** built by the download staging receiver is displayed.

XAI Download Staging - Response

To display the Response to this request, open **Menu > External Message > XAI Download Staging**.

Description of Page

The **XML Response** to this message is displayed.

XAI Download Exception

A record is written to the XAI download exception table for every XAI download staging record that is in error.

To view the messages associated with the exception records, schedule the **TD-XAIDN** background process. This process generates a To Do entry for every record in the XAI download exception table.

Maintaining Your XAI Environment

This section describes various tools provided to enable your XAI administrators to more easily maintain your XAI environment.

XAI Submission

This page exists for testing purposes. It allows you to create an XML request document and submit it to the system, to ensure that the XML has been built correctly.

XAI Submission - Main

To submit an XML document for testing, navigate to **Admin > XAI > XAI Submission** and navigate to the main tab.

Description of Page

This page is used to test XML schemas, which are defined for the XAI tool. Enter an appropriate XML document in the **XML Request** field. Typically, you define the XML schema using the schema editor in the XAI application. Then you would copy and paste the document here, then modify the schema to enter actual data for testing purposes.

When you have entered the document, choose **Save** to submit this document to the system. Note that this request information is not saved anywhere. It simply calls the system with the appropriate service name and executes the XML request.

Navigate to the **Response** tab to view the response.

XAI Submission - Response

To view the response to a XML document for testing, navigate to the response tab.

Description of Page

After choosing **Save** on the main tab to submit a test for an XML request, the response to your request is displayed in the **XML Response** text box.

XAI Dynamic Submission

This page exists for testing purposes. It is similar to the XML Submission page, but it dynamically builds your XML document based on a selected XAI service and data that you enter.

XAI Dynamic Submission - Main

To create and submit an XML document for testing, navigate to **Admin > XAI > XAI Dynamic Submission** and navigate to the main tab.

Description of Page

Select the **XAI In Service Name**, which identifies the XAI Inbound Service called by the XAI tool to load/update the system data. After selecting, click **Load UI**. This button causes the system to read the XML schema associated with the service and build the screen with the associated prompts and fields, which enables a user to enter data.

NOTE: Testing schema. This page tests submitting the schema referenced on an XAI inbound service. If your service references an XSL transformation script, that information is ignored. This page tests post-XSL transformation.

Select the **Transaction Type** associated with this XML request. The valid values are **Add, Change, Delete, List, Read, Search** and **Update**. This information is built into the XML request document.

Set the **Trace** option to **yes** to request level tracing to be executed inside the XAI tool. This results in information written to a file, which may be useful in debugging.

The bottom portion of the screen contains field prompts and input fields for the data associated with the XML request linked to this service. The system dynamically builds this portion of the page by reading the XML Request associated with the service. You can enter data in the displayed fields.

NOTE: This page also generates other tabs dynamically for any collections that exist for the service being displayed. These tabs vary based on the service. The response tab is the only other tab that is always present.

Enter values in the appropriate fields. You need to have some knowledge of what information is needed based on the service and transaction type. For example, if your service is accessing the account record and you want to **read** the record, you only need to provide the account id. However, if your transaction type is **add**, you need to fill in all the fields necessary for adding an account. If you have not entered values correctly, you receive an error in the Response.

When finished entering values in the fields, click **Show XML** to see the XML request built based on the schema and the values of the fields entered.

If everything looks OK, click **Submit** to execute the XML request. Note that this request information is not saved anywhere. It simply calls the system with the appropriate service name and executes the XML request.

You are brought to the Response tab, where you may view the response.

XAI Dynamic Submission - Response

To view the response to the XML document submitted for testing, navigate to the response tab.

Description of Page

After choosing Submit on the main tab to submit a test for the XML request built dynamically, the response to your request is displayed in the **XML Response** text box.

Additional XAI Tools

This section introduces some additional tools to help you maintain your XAI environment.

XAI Service Export

The service export page allows you to export the definition of an XAI Inbound Service to a file. This function may be helpful if you need to copy the definition of this service to a separate environment. To export a service, open **Admin > XAI > XAI Service Export**.

Description of Page

Upon entry into this page, you are provided with the current list of **XAI In bound Services** and their **Description s**. Use the **XAI In Service Name** search field to find the XAI service that you would like to export.

Use the **Export?** column to indicate which XAI service(s) you would like to export. Once you have selected your services, choose **Save**.

NOTE: If multiple services are selected, they are exported together into the same output file.

You are presented with the standard File Download dialogue where you can open or save the file.

XAI Service Import

The service import page allows you to import the definition of an XAI Inbound Service from a file into the XAI service table. This function may be helpful if you need to copy in the definition of this service from a separate environment. To import a service, open **Admin > XAI > XAI Service Import**.

Description of Page

Upon initial entry into this page, you are provided with an input field, where you can enter the file name to import. Click **Browse** to search for the desired file in a directory.

Once the file is identified, click **Read File**, to read in the contents of the file.

NOTE: The format of the file must include tags indicating the column names for XAI Inbound Service table along with the values of the columns. For an example of how the format should be, simply go to the [XAI Service Export](#) page, export a Service and view the format of the resulting file.

Once the file has been read in, the list of XAI services found defined within the file is displayed in the Import grid, identified by their **XAI In Service Name** and **Description**. In the **Import?** column, indicate which services to import.

If a service with this service name already exists in the table, you must check the **Overwrite Existing** switch in order to indicate that the imported file information should replace the current service. An [XAI Inbound Service](#) that is provided as part of the system (i.e., with an owner of **Base Product**) may not be overwritten.

Click Save to proceed with the import. If any problems are found, information is displayed in the **Message Text** column.

XAI Command

Use the XAI Command page to send commands to the XAI and MPL server. To execute a command, open **Admin > XAI > XAI Command**.

Description of Page

The following operator commands may be sent to the XAI server. For each of these commands, you may check **Also Sent to MPL URL**, in which case, the command is also sent to the MPL server. You need to indicate the URL of the MPL server.

Display Registry Use this command to display the current registry information that the XAI instance is running with.

Refresh Code and Description This is related to an attribute in the schema editor where you may indicate that the description of a control table code should be returned along with the code itself. This information is kept in cache memory in the server. As a result, changes made to descriptions have no effect on the runtime server. This command clears the cache of control table codes and descriptions accessed by the server.

Refresh Registry The registry contents are kept in cache memory in the server. As a result, making changes to the registry control tables has no effect on the runtime server. Use this command to refresh the contents of the registry cache with the new values in the registry control tables. The command reloads all registry control table data into the server.

Refresh Schema Schema definitions are stored in cache memory on the XAI server. As a result, modifying a schema definition has no effect on the runtime server. To refresh schema definitions, use the Refresh Schemas command.

Refresh Service Service definitions are stored in cache memory on the XAI server. As a result, modifying an XAI inbound service definition has no effect on the runtime server. To refresh service definitions, use the Refresh Service command. You are prompted to indicate which service to refresh.

Refresh XSL XSL Transformation script definitions are stored in cache memory on the XAI server. As a result, modifying an XSL transformation definition has no effect on the runtime server. To refresh XSL transformation definitions, use the Refresh XSL command.

Trace On Use this command to start the XAI server trace.

Trace Off Use this command to stop the XAI server trace.

XAI Trace Clear Use this command to clear the contents of the trace file.

XAI Trace Swap Use this command to rename the current trace file by appending the date and time to the end. A new trace file is then created with the name as defined in the [Message option](#) page.

The following operator commands can be sent to the MPL server. You must set the URL of the MPL server first.

MPL Refresh Executer Executer definitions are stored in cache memory. As a result, adding or modifying executer definitions has no effect on the runtime server. Use this command to refresh executer definitions. You are prompted to indicate the executer to refresh.

MPL Refresh Receiver Receiver definitions are stored in cache memory. As a result, adding or modifying receiver definitions has no effect on the runtime server. Use this command to refresh receiver definitions. You are prompted to indicate the receiver to refresh.

MPL Refresh Sender Sender definitions are stored in cache memory. As a result, adding or modifying sender definitions has no effect on the runtime server. Use this command to refresh sender definitions. You are prompted to indicate the sender to refresh.

MPL Start Receiver Use this command to start a particular receiver. You are prompted to indicate the receiver to start.

MPL Stop Use this command to stop all MPL activity. It stops all receivers and waits for all executers and senders to complete.

MPL Stop Receiver Use this command to stop a particular receiver. You are prompted to indicate the receiver to stop.

MPL Trace On Use this command to start the MPL server trace.

MPL Trace Off Use this command to stop the MPL server trace.

When you have chosen the appropriate command and indicated any extra information, click **Send Command** to send the command to the server(s).

If you have sent a command to the XAI Server, then the bottom portion of the screen displays the response in the **XAI Response**. If you have sent a command to the MPL Server, then the bottom portion of the screen displays the response in the **MPL Response**. If you have sent a command to both servers, the bottom portion of the screen displays both responses.

MPL Exception

The MPL Exception table is used by the MPL to keep information about requests that resulted in a system error. These are errors that occurred inside the MPL. For example, if the MPL fails to send a request to XAI (maybe WebLogic is down), this is a system error, which would be logged in the MPL exception table.

There are errors that are defined recoverable. This means that the MPL will retry the action that failed, according to the parameters it received.

Server Trace

The XAI server traces every request and response. The requests/responses are written to a trace file on the server. The trace file may be viewed using the [Trace Viewer](#).

Starting the Trace

The log starts automatically based on definitions in the [Message Options](#) in the traceType and traceFile options. To manually start the trace:

- Navigate to **Admin > XAI > XAI Command**.
- Select the **Start Trace** command from the command dropdown
- Click **Send Command**

Stopping the Trace

- Navigate to **Admin > XAI > XAI Command**.
- Select the **Stop Trace** command from the command dropdown

- Click **Send Command**

Trace Viewer

Use the Trace Viewer utility to view the log file. The Trace Viewer is installed when you install the XAI client tools. It can be found in the XAI program group under Start/Programs.

Main Page

When the Trace Viewer starts, select a trace file to view. A trace file may be opened in one of two ways:

- To open a trace file directly from its location on the web application server, use the **File, Open HTTP** menu item and provide the appropriate URL.
- To open a trace file on the local/network file system use the **File, Open** menu item

Description of Page

Once a trace file is opened, it displays a list of all the requests on the left side including the **Service Name**, the **Start Time** and the **End Time**.

To display the XML contained in the request and response entries for a displayed request, select a request entry.


Filtering Options

Since the trace file may contain a very large number of messages, the trace viewer limits the number of messages that can be displayed. It does that by displaying messages traced within the last x number of **Minutes, Hours** or **Days**.

Use the **Max Messages** to limit the number of messages displayed.

NOTE: Default Note. By default, the Trace viewer displays the first **200** messages in the trace file.

To view only errors in the trace, check the **Show only Errors** option.

NOTE: Refresh Display. After changing any of the above filtering options, click the refresh button  in order to redisplay the request entries based on the new options.

The **First Message Found** field indicates the date and time of the earliest entry in the trace file.

Viewing as Text

To view the trace file as text rather than viewing each entry in its XML format, use the **View, As Text** menu option. The contents of the trace file are displayed in text format in a separate window.

Statistics Page

Use the **View, Statistics** menu item to view the statistic page, which displays performance statistics about the XAI services that were executed in the XAI trace file.

For each type of XAI Service and transaction type, it displays the following information based on the requests traced in the XAI trace file:

- The **Service Name** with the transaction type in parentheses
- The **Number of calls** for this service in the listed trace records
- The **Average duration Time** (in seconds)
- The **Max imum duration Time** (in seconds)
- The **Min imum duration Time** (in seconds)

NOTE: Requests Included in Statistics. Only requests falling in the time selection criteria and listed on the main log viewer are processed for calculating the statistics.

To display a Duration Chart for a particular service, check the Service. A chart such as the one below is displayed.

Integrations

This chapter provide high level information about product integrations supported for all products that use Oracle Utilities Application Framework.

LDAP Integration

Organizations commonly use a Lightweight Directory Access Protocol (LDAP) security repository as a source of security credentials. The system provides support for importing users and groups from an external LDAP repository into the product to populate Users and User Groups in the system. Once imported, all user and group functions are available. You can resynchronize your LDAP users and groups at any time.

NOTE: Import only. The system currently supports importing LDAP information. Exporting users and groups from the system to LDAP is not provided.

NOTE: Additional configuration. When importing new users and / or groups, additional configuration is needed in the base product. For example, after importing a new user group and its users, the user group configuration should be updated to define the valid application services for the user group. After importing a new user, additional configuration may be needed on the user such as valid To Do Roles, valid Home Page, etc.

FASTPATH: Refer to [Defining Security and User Options](#) for more information about the application security and what it controls.

This section the functionality provided in the framework application that supports LDAP. Refer to the *LDAP Integration* white paper for more information about typical steps related to the full integration.

LDAP Integration Overview

This topic provides a high level overview of the integration process.

At a high level, the base product provides a process to import user group and / or user definitions from and LDAP repository. This is a one way integration.

- When importing a user, if it is not already found in the system, it will be added; otherwise its attributes will be updated according to the imported information.
- When importing a user group, if it is not already found in the system, it will be added; otherwise its attributes will be updated according to the imported information.
- When importing a user, its user group links will be updated as per the information in the import file. In addition, if there are any user groups linked to the user that are not found system, they will be added (however, the other users linked to that group in the LDAP repository will not be added as part of this step).
- When importing a user group, its user links will be updated as per the information in the import file. In addition, if there are any users linked to the user group that are not found system, they will be added (however, the other user groups linked to that user in the LDAP repository will not be added as part of this step).
- The import will not cause any deletions of the User or User Group to occur.

A Batch Process Initiates the Import

A batch process is used to initiate the import of information from the LDAP repository. **F1-LDAP** may be submitted ad hoc or may be set up in a scheduler to periodically re-sync the information from the LDAP repository into the application.

The batch process uses parameters to define how to connect to the LDAP repository. In addition, parameters are used to indicate which user or group is being imported.

Adjusting Data to Import

The system provides several mechanisms for adjusting data that is being added to the system:

- There is an **LDAP Import Preprocess** algorithm plug-in spot on the [installation](#) record. Algorithms plugged in here are called by the batch process prior to the add or update of any records. It may be used to make adjustments to the data before doing updates in the application.
- Specifically for creating or updating Users, the **F1-IDMUser** business object is used to add and create the user. The standard BO Preprocessing algorithm plug-in spot may be used to adjust data prior to creation.
- The LDAP mapping file supports some attributes to perform simple modifications to data.
 - The **transform** attribute supports values to truncate values or to convert data to upper case.
 - The **autoGenerate** attribute is specific to the User ID field. Setting this to true will trigger code that will automatically populate the User ID based on the user's name. Refer to [LDAP Mapping](#) for more information.

Performing Additional Processing After Import

The system provides a plug-in spot on the [installation](#) record called **LDAP Import**. Algorithms plugged into this spot are called after users or user groups have been added or updated. It may be used to perform any extra processing that may need to be executed.

In addition, for any additional processing related to the creation or update of a User, the standard [Business Object plug-ins](#) may be used for the **F1-IDMUser** business object which the LDAP batch process uses to create or update users.

Configuring LDAP Integration

To interface the LDAP based security repository with the authorization component of the Oracle Utilities Application Framework product the following must be performed:

- The location and port number of the LDAP based security repository must be defined to in the JNDI Server.
- • The LDAP based security repository must be mapped to the Oracle Utilities Application Framework security model. This mapping is expressed as an XML file containing the LDAP query, rules and defaults used in the transformation.
- • The mapping file must be configured on the **F1-LDAP** batch job.

Define the JNDI Server

The first step in the configuration process is to define the location of the LDAP based security repository server so that the interface can connect to the physical attributes of the interface. This is done by creating a [JNDI Server](#).

NOTE: The LDAP server is strictly not a JNDI source but is treated as a JNDI source for the integration.

Enter a reasonable JNDI Server name and description.

Populate the **Provider URL** using the format **ldap://<hostname>:<portnumber>** where **<hostname>** is the host of the LDAP server and **<portnumber>** is the port used for the interface.

For the **Initial Context Factory**, the interface uses the standard **com.sun.jndi.ldap.LdapCtxFactory** provided with java for the LDAP interface. If your vendor supplies a custom context factory it may be used. Refer to the documentation provided with your LDAP based security repository for further information.

Define Mapping

The critical component of the interface is a file that describes the mapping between the LDAP based security repository and the system's security model. This file contains the mapping, rules and queries used by the LDAP batch program to provide the interface. The LDAP batch job includes the reference to the mapping file as a parameter. Refer to [LDAP Mapping](#) for more information on defining the mapping file.

Configure LDAP Batch Process

At this point, many parameters for the **F1-LDAPbatch control** can be updated with system wide configuration.

- JDNI Server, User and Password may all be configured appropriately. Note that it is recommended that the **Security** setting for the Password be set to **Encrypt**.
- The LDAP Configuration File should be populated with the name and location of the LDAP Mapping file.

NOTE: Group and User Parameters. The assumption is that the Group or User input parameters are specific to a given import request and as such would not be populated as part of a configuration step.

NOTE: L2 Cache. The LDAP Import batch process requires the L2 Cache to be disabled since it needs to perform some updates in the outside of the worker threads. Any environment using LDAP Import must set **spl.runtime.batch.L2CacheMode=OFF** in the **threadpoolworker.properties** file. It is recommended to run the LDAP import in its own dedicated threadpoolworker.

LDAP Mapping

An LDAP repository consists of multiple entries. Each entry represents an object in the directory that is identified by a Distinguished Name (DN) and may contain one or more attributes. In a typical LDAP repository there is usually an entry for users and an entry for groups. The connection between users and groups may be implemented in two different ways:

- The users belonging to a group are defined in a special multiple-value attribute on the Group entry.
- The groups to which a user belongs are defined in a special multiple-value attribute on the User entry.

The mapping between LDAP security objects and base security objects is stored in an XML document that can be processed by the LDAP import batch job. As part of setting up your system for LDAP import, you need to define this mapping. The base package provides a sample mapping file called **ldapdef.xml** that can be used as a starting point and changed per your business requirements and your particular LDAP repository.

Once you have defined the mapping XML document, this is configured as a parameter in the **F1-LDAP** batch job.

The XML structure:

- The **LDAPEntry** element maps the LDAP entries to system objects (User or Group). The mapping file must contain one and only one LDAPEntry element for User and one for Group.
- The **LDAPCDXAttrMapping** element within the LDAPEntry element maps attributes in the LDAP entry to attributes in the system object.
- The **LDAPEntryLinks** element describes objects linked to the LDAP entry. When mapping the user entity you need to describe how the groups the user belongs to are retrieved. When mapping the group entity you need to describe how the users contained in the group are retrieved.

The following table describes the attributes to define for each element.

Element	Attribute	Description
LDAPEntry	name	The name of the LDAP entry: - Group - User

Element	Attribute	Description
	baseDN	The base distinguished name in LDAP for this entry.
	cdxEntity	The name of the base product entity to which the LDAP entry is mapped: - Group - User
	searchFilter	An LDAP search filter that is used to locate LDAP entries. A %searchParm% string in that filter is replaced by the value from the user or group parameter from the F1-LDAP batch job submission.
	Scope	Sets the scope of the search. Valid values are: - onelevel (the value normally used) - subtree
LDAPCDXAttrMapping	ldapAttr	The name of the LDAP attribute to be mapped. Note that this may be referenced more than once to allow one LDAP element to map to multiple base product elements. For example, if an email address should be used both for the Login ID and the Email Address.
	cdxName	The name of the base product attribute to be mapped. For User, this is the element within the F1-IDMUser business object. For Group, this is either the 'group' or the 'description'.
	default	The default value that will be assigned to the element referenced in the cdxName attribute when the following occurs: - The LDAP attribute contains a null or empty value - The LDAP attribute does not exist or is not specified. Default values are applied only when creating a new entity and are not applied to updated entities.
	autoGenerate	Set this to true in order to turn on auto generation of the user ID. If this is true, the system will define user id as <first initial of first name>+<last name> all uppercase, to a maximum of 8 digits. If an existing user is found for the generated ID, a number will replace the eighth digit (or be appended to the end). The system will increment the number until a unique ID is found.
	transform	Use this attribute to indicate if a transformation of the data should occur. Valid values: uppercase , truncate . Note that this attribute should not be used in conjunction with the autoGenerate attribute.
LDAPEntryLink	linkedToLDAPEntry	The name of the linked entity (User or Group). Use User when describing the Group entity. Use Group when describing the User entity.
	linkingLDAPAttr	The multiple-value attribute name on the LDAP entity that contains the linked entity.
	linkingSearchFilter	The search filter to be applied to retrieve the list of linked objects, for example: (&!(objectClass=group)(memberOf=%attr%)) The search filter may contain the string % attr % that acts as a substitution string and is replaced at run time by the value of the attribute named "attr" of the imported entity. If the LDAP entry you are describing is a Group and the string is %name%, it is replaced by the value of the "name" attribute of the group you are importing. If the LDAP entry you are describing is a User and the string is %dn%, it is replaced by the "dn" attribute of the User you are importing.
	linkingSearchScope	Sets the scope of the search. Valid values are: - onelevel (the value normally used) - subtree

Sample Mapping

The following XML describes a sample mapping. The example makes the following assumptions:

- The base product attribute **displayProfileCode** is defaulted to "NORTHAM" when adding a new user.
- The LDAP Group entry contains the list of users belonging to the group in the **departmentNumber** attribute.
- The groups to which a user belongs are retrieved by applying a search filter.

```
<LDAPEntries>
  <LDAPEntry name=" User" baseDN="ou=people,dc=example,dc=com" cdxEntity=" user" searchFilter="(objectClass=inetOrgPerson)(uid=%searchParm%)">
    <LDAPCDXAttrMappings>
      <LDAPCDXAttrMapping ldapAttr="uid" cdxName=" user" />
      <LDAPCDXAttrMapping ldapAttr="cn" cdxName="externalUserId" />
      <LDAPCDXAttrMapping cdxName="language" default=" ENG" />
      <LDAPCDXAttrMapping ldapAttr="givenName" cdxName="firstName" />
    </LDAPCDXAttrMappings>
  </LDAPEntry>
</LDAPEntries>
```

```

<LDAPCDXAttrMapping ldapAttr="sn" cdxName="lastName" />
<LDAPCDXAttrMapping cdxName="displayProfileCode" default="NORTHAM" />
<LDAPCDXAttrMapping cdxName="toDoEntriesAge1" default="30" />
<LDAPCDXAttrMapping cdxName="toDoEntriesAge2" default="90" />
<LDAPCDXAttrMapping cdxName="userEnable" default="ENBL" />
</LDAPCDXAttrMappings>
<LDAPEntryLinks>
  <LDAPEntryLink linkedToLDAPEntity="Group" linkingLDAPAttr="departmentNumber" />
</LDAPEntryLinks>
</LDAPEntry>
<LDAPEntry name="Group" baseDN="ou=people,dc=example,dc=com" cdxEntity=" Group" searchFilter=" (&
(objectClass=organizationalUnit)(ou=%searchParm%))">
  <LDAPCDXAttrMappings>
    <LDAPCDXAttrMapping ldapAttr="name" cdxName="Group" />
    <LDAPCDXAttrMapping ldapAttr="description" cdxName=" Description" default="Unknown" />
  </LDAPCDXAttrMappings>
  <LDAPEntryLinks>
    <LDAPEntryLink linkedToLDAPEntity="User" linkingSearchFilter=" (&
(objectClass=inetOrgPerson)(departmentNumber=%distinguishedName
%))" linkingSearchScope="onelevel" />
  </LDAPEntryLinks>
</LDAPEntry>
</LDAPEntries>

```

Oracle Identity Manager Integration

The *Oracle Identity Manager* product allows a site to centralize their user definitions and password rules to manage and deploy across the enterprise set of products. When an employee joins an organization, changes their name or departs an organization their security presence across an enterprise must be appropriately managed. Oracle Identity Manager allows for users to be provision and managed in a central location.

An integration is provided to allow the ability to create, maintain and remove users in the identity management product and sync those changes to the users defined in the application. The following sections provide additional details about the integration with respect to configuration steps required in an Oracle Utilities Application Framework based product. For more information about the configuration required in the identity management product, refer to the *Identity Management Suite Integration* white paper.

Template User Functionality

The user object in this product captures configuration used to control access but also preferences. The identify management product allows for extending the configuration to capture user configuration that is specific to this product. However, it does not support providing searches or dropdowns to select valid values. For example, to define the user's Home Page requires the reference to a navigation option. To set up your business process such that the home page is configured when defining the user in the identity management product dictates that the security user types in the correct navigation option reference.

On the other hand, to define a minimal amount of user information in the identity management product may result in a two step process for defining users: first define them in the identity management product with the basic authentication details and setting system defaults for some important fields, then after submitting the new user to be added to this product, navigate to the [user](#) page in this product and fill in all the configuration that is specific to this product.

The product provides support for defining a template user that can facilitate the definition of users and reduce some of the challenges listed above. The concept is as follows:

- Define a template user for each broad category of users in the system. For example, Oracle Utilities Mobile Workforce Management may define the following template users: Dispatcher, Mobile Worker, System Administrator and Contractor. Each user would define the typical configuration for users of that type including the home page, the user groups, the To Do roles, the portal preferences, etc.
- When extending the configuration in the identity manager product, simply map the information that is unique to a user and in addition, define a field for the template user. For example, you may choose to only capture the Name (first and last), Email address and User IDs for the user along with its Template User (which is mapped to a user characteristic). Additional fields may be included for capture in the identity management product when defining new users as per an

implementation's business needs. For example, if the organization covers multiple time zones, perhaps it is easier to define the user's time zone when defining the user in the identity management product.

- When the new user is uploaded to the system, the interface uses the user BO **F1-IDMUser** to create the user. The BO includes a preprocessing algorithm that looks for the existence of a template user (sent as a characteristic of type **F1-TMUSR**). All the information from the template user will be copied onto the new user record except for the information passed in from the identity manager. The template user is captured on the newly created user via a characteristic for information / audit purposes.
- Once the new user is created, its configuration can now be adjusted, if applicable. Note that the template user is only a tool used when adding a user. Updates to the template user will not “ripple” to all the other users that were created based on this template.

Configuring Template Users

Before configuring template users, all the administrative control tables that are part of User configuration must be defined, including time zones, display profiles, To Do roles, data access roles and user groups.

The next step is to define the user configuration for your system users. During this exercise, you will find that you have broad categories of users. But you will also see that within a given category of user there may be variations in the user privileges and preferences. For example, perhaps there are supervisors within the Mobile Worker role that have more security privileges than a typical Mobile Worker. In addition, there may be variations based on the attributes of the users themselves. For example, maybe your organization exists in multiple time zones and some of your workers are in one time zone and some are in the other.

At this point your security users that are designing their user provisioning procedures must decide the following:

- What information about a new user will be captured in the identity system (besides the expected information like Name, Email and the User IDs)? For example, for the case of multiple time zones, maybe the best solution is to capture the time zone when defining the user.

What information is defined on the template user and how many template users should be created to reduce the need for manual steps or additional data captured in the identity management system? In the case of multiple time zones, proliferating the template users to have one set for one time zone and another set for the other time zone may not make sense since this is one field that is different. However it may be reasonable to create additional templates in the case of variations in the levels of privileges for workers of a different category. So rather than template users for Dispatcher, Mobile Worker, System Administrator and Contractor, your organization may have template users for Dispatcher, Mobile Worker, Mobile Worker (Supervisor), System Administrator, Contractor (Short Term) and Contractor (Long Term).

What information is must be configured one the User record in the application after the user is added? If only a small number of users have a variation from other users, it may be that the easiest way to deal with those variations is to simply update those user records manually. Using the above examples,

- If your organization covers 2 time zones but only a small group of people work in one of the time zones whereas the bulk of the users are in the other time zone, the simplest procedure may be to define the template users for the main time zone and use that for the creation of all users. Then for the small group of users in the separate time zone, navigate to the User page to adjust the time zone after the record is added.
- If only a small number of Mobile Workers are supervisors with separate privileges, rather than defining a special template user for those type of workers, the simpler procedure may be to use the Mobile Worker template and then navigate to the User page to add the additional privileges to the supervisor users after the record is added.

To create a template user, navigate using **Admin > Security > User** .

- Define a User ID that will become the template user reference in the identity management system.
- Be sure to choose a **User Type** of **Template User**.
- Define all the information that should be copied onto a new user that references this user as a template user. Note that **Bookmarks** are not included in the data that is copied from a template user.

NOTE: There is configuration needed in Oracle Identity Management to capture the template user and any other information that the implementation has chosen to define in the identity management product when provisioning a new user. Refer to the *Identity Management Suite Integration* white paper for more information.

Data Synchronization

NOTE: The information in this section applies to the integration with Oracle Utilities Meter Data Management and/or Oracle Utilities Network Management System.

Defining Sync Requests

This section highlights functionality provided to help an implementation manage the communication of data to an external system.

The Big Picture of Sync Requests

Your implementation may need to communicate certain data to external systems. This may be part of a data warehousing requirement or an integration effort. The synchronization process has two main parts. First, the change to the data must be detected and captured. Once that is accomplished, the next step is to manage the communication of that change to the external systems involved. The changes must be captured in chronological order so as to avoid systems going out of sync.

Capturing the Change

The base product has provided the **Audit** plug-in spot on the MO to allow for processing to be called by the system upon detecting a change to an MO. The framework calls the algorithm defined on this plug-in spot in the event a change to the MO has been detected. The base product comes with a generic change data capture algorithm **F1-GCHG-CDCP** that makes use of this plug-in spot to create the sync request BOs. The **Sync Request BO** MO Option is used to define these sync request BOs. The generic algorithm loops through this list of BOs and instantiates them, if no instances of those BOs yet exist in the initial state. The MO and primary key used for these sync requests will be the MO being modified and its primary key.

Sometimes it is necessary to instantiate a sync request whose MO is different from the MO being modified. As an example, a change to the Person record may need to instantiate an SA sync request. In this case, your implementation will have to create your own Audit algorithm that captures this idiosyncratic logic. For examples on how this is done, please refer to the *Oracle Utilities CCB-MDM Integration Implementation Guide*.

Sync Request Maintenance Object

The sync request maintenance object (MO) is used to communicate data elements to an external system. It holds the data elements that are to be synchronized or extracted out of your system and also holds the MO and primary key constituents that drive the sync request. If your organization wishes to use this MO, you can either use the business object (BO) supplied in the base product or configure your own BO.

This MO provides the following functionality:

- A business object option **External System** is provided for business objects of this type. The value specified on this option is used by the base product to create an entry in the JMS Queue.
- A business object option **Outbound Message Type** is provided for business objects of this type. The value specified on this option is used by the base product to create an entry in the JMS Queue.
- An **Information** algorithm is provided (**F1-SYNC-INFO**).

- The standard **Determine BO** algorithm is provided.
- The standard logs are provided.
- The standard characteristics collection is provided.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference F1-SYNC REQ.

For more information about this MO and to review the business objects defined for this MO, navigate to **Admin > System > Maintenance Object** and view the MO **F1-SYNC REQ**.

Sync Request Business Object

The sync request business object (BO) contains the rules that govern the processing of a sync request. The algorithms in this BO control what data elements are synchronized to an external system; how this data is to be communicated to the external system; and how the communication of this information is controlled chronologically to ensure that changes are processed in the right order. The base product provides the BO **F1-SyncRequest**, which contains the following life cycle.

- **Pending.** The initial state of the sync request BO. At this point, a pre-processing algorithm would have already taken a snapshot of what the relevant information looked like prior to the change. A deferred monitor algorithm is specified on this state so as to accumulate all changes prior to sending the sync request to the external system. Furthermore, in order to prevent the sync request from transitioning to the next state while another sync request may not have completed its processing yet, the base package provides the monitor algorithm **F1-WAITRELSR** that skips transitioning if it detects another instance of the same BO in a non-final state for the same MO and primary key combination.
- **Determine If Sync Needed.** This transitory state will contain the algorithm that takes the final snapshot of the relevant information so that it may be compared against the initial snapshot to see if the change affected any of the relevant data elements. The base product provides the algorithm **F1-COMPSNAPS** for this purpose. Your implementation may also add your own rules here to discard the sync request if certain criteria are met.
- **Discarded.** If the comparison of the initial and final snapshots yields no changes, or if it fails any other checks your implementation has introduced, the sync request is transitioned to this state. If any processes are dependent on the sync request transitioning to a final state, your implementation may plug in an enter algorithm on this state to kick off the dependent process.
- **Send Request.** This transitory state is used to hold the algorithm that facilitates the sending of the sync request to the external system. The algorithm used here may write a message to a JMS Queue or write a record to the general process MO or perform any other task that is suited to how your implementation communicates the change.
- **Awaiting Acknowledgement.** This state can be used to hold the sync request from further state transitions until an acknowledgement is received from the external system. We strongly recommend designing some form of acknowledgement that the external system has processed the information as this helps control the chronological flow of information. The base package provides the business service **F1-UpdateSyncRequest** that transitions the sync request to either the next default state (in this case the Synchronized state) if a positive acknowledgement is received; or the state associated with the Rejection transition condition (in this case the Error state) if a negative acknowledgement is received.
- **Synchronized.** Once the information has been processed and acknowledged by the external system, the sync request is transitioned to this state. If any processes are dependent on the sync request processing successfully, your implementation may want to plug in an enter algorithm on this state to kick off the dependent process.
- **Error.** If a negative acknowledgement is received or if the acknowledgement is not received within the prescribed period of time, the sync request is transitioned to this state. Your implementation may wish to create a to do entry when the sync request transitions to this state. The base product provides the enter algorithm **F1-TDCREATE** for this purpose. The exit algorithm **F1-TODOCOMPL** has also been provided to complete any to do entries when leaving this state.
- **Canceled.** A user may transition the sync request to this state from either the **Awaiting Acknowledgement** state or the **Error** state if he wishes to discontinue processing of the sync request for any reason. Your implementation will have

to configure your own cancel reasons to document the transition. If any processes need to be triggered when the sync request reaches this state, your implementation can plug in an enter algorithm containing such logic.

Designing Your Sync Requests

The starting point for designing sync requests is the analysis of the external system's data requirements. Your implementation will need to analyze the external system's data model and compare it against your system's data model. Where the relevant data reside in your system will need to be identified. If the data reside in more than one MO, a decision must be made to either combine the information in one sync request or to break up the sync request into several ones.

For each sync request identified, the driving MO must be determined. The BO for each sync request can then be created as a sub-class of the base BO **F1-SyncRequest**. For an example of how these BOs are configured, please refer to *Oracle Utilities CCB-MDM Integration Implementation Guide* or *Oracle Utilities CCB-NMS Integration Implementation Guide*.

Sync Requests

The Sync Request portal is used to view information about data synchronizations from Oracle Utilities Customer Care and Billing to Oracle Utilities Meter Data Management, or from Oracle Utilities Customer Care and Billing to Oracle Utilities Network Management System. The menu location of the portal depends on your specific edge product.

Sync Request Query

Use the [query portal](#) to search for an existing Sync Request. Once a Sync Request record is selected, you are brought to the maintenance portal to view and maintain the selected record.

Sync Request Portal

This portal appears when a sync request has been selected from the Sync Request Query portal.

The topics in this section describe the base-package zones that appear on this portal.

Actions

The actions zone only appears for sync request business objects that define an actions map. For sync requests that are designed to show the actions directly in the display map, this zone is suppressed.

Sync Request

The Sync Request zone shows display-only information about the sync request. Refer to the zone's help text for additional information about this zone's fields.

Sync Request - Log

The sync request log page contains a standard [Log Zone](#).

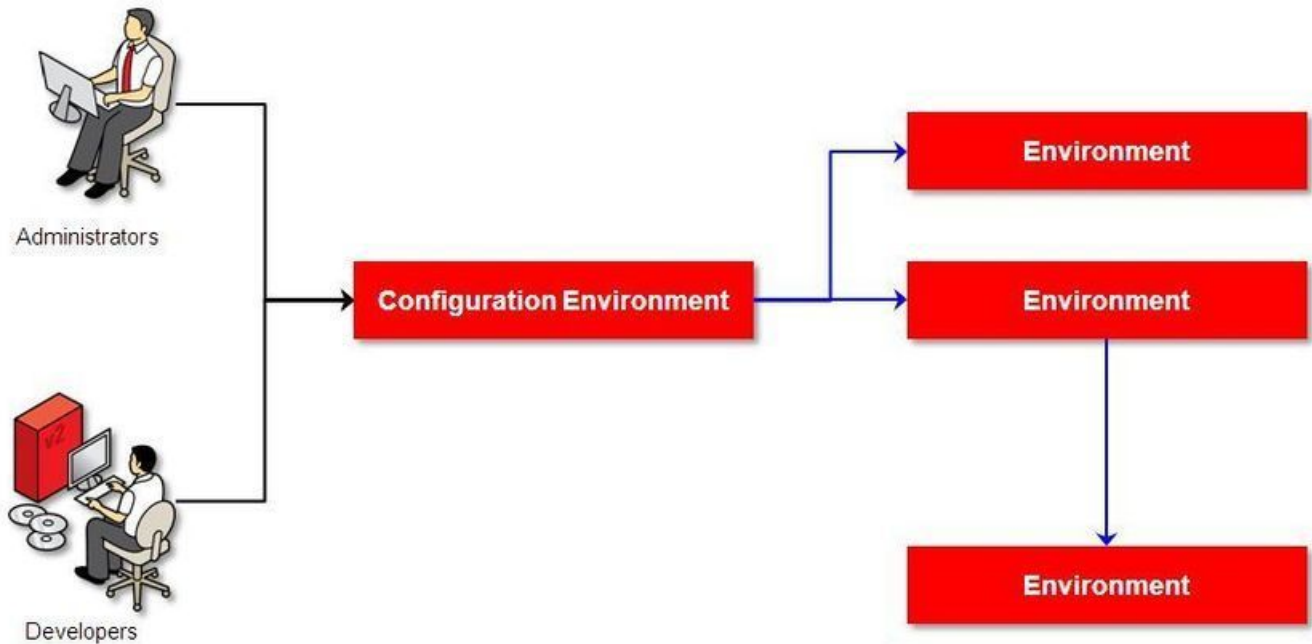
Configuration Migration Assistant (CMA)

This chapter describes the Configuration Migration Assistant (CMA), a facility to enable the export of customer-owned configuration data from one environment to another.

CAUTION: This chapter is intended for users responsible for testing configuration data and performing "what if" analysis in non-production databases.

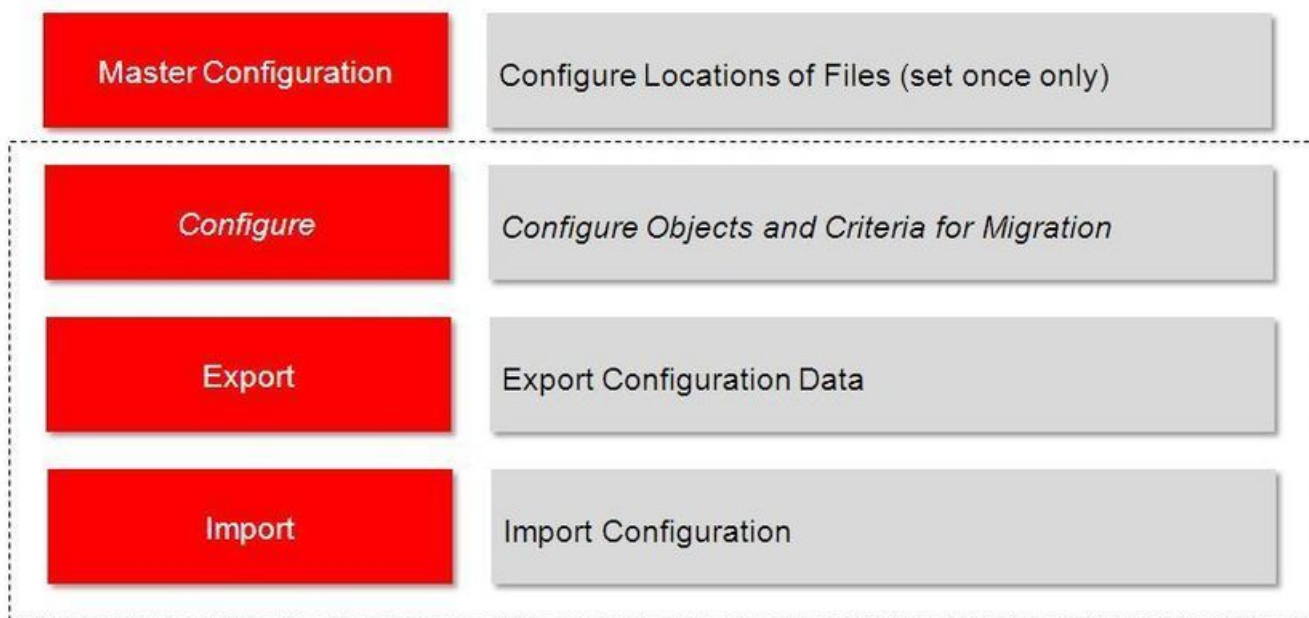
Understanding CMA

The Configuration Migration Assistant (CMA) provides implementers with a flexible, extensible facility for migrating configuration data from one environment to another (e.g., from a development environment to a production environment). Data is exported from the source system to a file. The file can then be checked in to a version control system for reuse, or can be immediately imported into the target system and applied.



NOTE: As used in this chapter, *source* systems are those on which export-related activities are conducted and *target* systems are those on which migration updates are to occur. The two system preparation tasks described in [Migration Assistant Configuration](#) must be performed on both source and target systems.

The CMA Process Flow



The high-level CMA process comprises the following tasks and flows. Each is described in more detail later in this section.

- **One-time tasks**, which must be applied on both source and target systems:
 - Specify the source and target paths and a file extension for import and export data files in the **Migration Assistant Configuration** master configuration record.
 - Create a thread pool that turns off the L2 caching. This is required to support the import step where cached data is being added or updated. It's recommended for all steps (including export and compare) to ensure that the algorithms are working with the data in the database rather than the cache. Refer to [Caching Considerations](#) for more information.
- **Configuration** steps are used to define the data to migrate. This task is performed on the source system and may be defined at any time. Note that the products provide base delivered configuration that may be used as is or used as a template for building more specific configuration for a given implementation.
- Each type of record that may be copied requires a **Migration Plan**. The migration plan is used to identify the maintenance object (MO) for the record (using a business object) and allows for instructions to specify related records that may be included in the migration. Multiple migration plans may exist for a given maintenance object if there are different requirements for migrating records in that MO under different circumstances.
 - The primary instruction in the migration plan could define the physical BO of the record. This signals to CMA that all records for the MO are eligible to be migrated.
 - The primary instruction may define a specific BO, in which case only records that reference that BO in its parental hierarchy are considered.

NOTE: Refer to [Understanding the BO Filtering Process](#) for more information.

- Subsequent instructions may include references to related records. There may be some circumstances where a migration should include subsequent records and some circumstances where they shouldn't based on requirements.

- A **Migration Request** is used to define a set of migration plans to be included together in a single migration attempt. In addition, the migration request may define selection criteria for each migration plan to limit the migration to a subset of data for each MO. Selection may be defined using SQL, an algorithm or explicit primary key values.

FASTPATH: For more information, refer to [CMA Configuration](#).

- The **Export** process includes all the steps needed to select records to be exported from the source environment and create the export file.
 - A **Migration Data Set Export** object is created for a specific migration request.
 - The lifecycle of the Migration Data Set Export business object includes algorithms that select the appropriate records according to the migration request, determine dependencies between records to build groupings of related objects and create the export file.
 - Because there may be a large volume of data in the export, many of the steps in the lifecycle of the migration data set export are executed via the **Migration Data Set Monitor**.

FASTPATH: For more information, refer to [Exporting a Migration](#).

- The file created by the export, which is a BINARY file, needs to be transferred from the export directory to the import directory. The transfer needs to be done in such a way as to preserve the file structure. Refer to [Migration Assumptions, Restrictions and Recommendations](#) for more information.
- The **Import** processes include all the steps needed to read an imported file, compare the data in the file to the data in the target, review the proposed changes and apply the updates. The following is a high level overview of the process.
 - A **Migration Data Set Import** object is created for a given file to import.
 - The next step reads the import file and creates **Migration Transactions** and **Migration Objects** based on the information in the import file. A migration object is created for every maintenance object record to potentially be created or updated. The migration transaction is a grouping record that groups together related migration objects.
 - The next step is for the objects to **Compare** the data being imported against the data for that record in the target environment. If it is found that the migration object is new or represents a change to the existing record in the target environment, appropriate SQLs are generated. If no changes are found, the object is marked “unchanged” and doesn’t progress.
 - Once all the objects are compared, the user may review the objects for acceptance or rejection.
 - When the migration objects are all accepted or rejected, the next step is to apply the objects and update the target environment.

FASTPATH: For more information, refer to [Importing and Applying a Migration](#).

Migration Assumptions, Restrictions and Recommendations

The following sections describe miscellaneous topics that are important to learn with respect to CMA.

Type of Data Migrated

CMA is designed to migrate configuration data only. The tool cannot be used to migrate master or transactional data that contains system-generated primary keys.

The comparison step of the import process will generate appropriate insert or update SQL statements for the following data found in the export:

- Configuration data in a maintenance object with no owner flag. This is purely implementation data.

- Configuration data in a maintenance object with owner flag, where the owner is **Customer Modification**. For example, implementer-specific business objects.
- Configuration data in a maintenance object with owner flag, where the main record is owned by the product but where a child record exists with an owner of **Customer Modification**. For example, implementer-specific algorithms added to a base owned business object.
- Customizable fields in a record that is owned by the product. For example, the priority of a based owned To Do type.

No Record Deletion

The CMA process allows users to define records to copy from a source environment to a target environment. In that way, the import process for the migrated records is able to identify objects to add and objects to change. There is no mechanism for indicating that records in the target environment should be deleted. The absence of those records in the import is not enough because the migration may be only importing a subset to add or update. If data on the target system must be deleted, users must delete the records in the target accordingly.

Note that CMA does support the deletion of child rows of an object as a result of a comparison. This is only applicable to child records that are owned by the implementation.

File Transfer Considerations

When moving the export file between systems, use the binary transfer option of whatever tool you use to move the file so that line-end characters are not converted from Linux-style to Windows-style or vice versa.

It is recommended to avoid using '.txt' for the export file's extension (defined in the [master configuration](#)). That file extension by default implies a non-binary file and tools that perform file transfer may treat this as a non-binary file unless explicitly stated. The recommendation is to define '.cma' as the extension. This is not a recognized file extension and most file transfer tools will transfer the file as is.

Note that if the file gets converted, there are two likely outcomes - either a numeric conversion error, or a buffer under-run error may be received when attempting to import the file.

CMA Configuration

The following sections describe tasks required for CMA configuration.

Master Configuration - Migration Assistant

Before proceeding with your first migration, system wide configuration must be defined. This is captured in the **Migration Assistant Configuration** master configuration record. This must be defined in both the source environment and the target environment.

Navigate using **Admin > Master Configuration**. In the **Master Configuration** list, scroll to **Migration Assistant Configuration** and click the add icon if there is no record yet or the edit icon to modify an existing record.

For more information about specific fields in the master configuration, refer to the in-line help.

NOTE: This record can be updated at any time to change details. The new configuration takes effect on all subsequent exports and imports.

Migration Plans

A migration plan defines one or more types of objects that are eligible for migration. It is essentially a set of instructions describing how the data to be exported is structured, allowing objects to be migrated together as a logical unit to ensure consistency and completeness.

The following topics describe defining a migration plan as well as other topics for a migration plan.

Defining a Migration Plan

To view or define a migration plan, navigate using **Admin > Migration > Migration Plan**.

Use the **Migration Plan Query** portal to search for an existing migration plan. Once a migration plan is selected, you are brought to the maintenance portal to view and maintain the selected record.

CAUTION: Important! If you introduce a new migration plan, carefully consider its naming convention. Refer to [System Data Naming Convention](#) for more information.

The following points provide information about defining **Instructions** for a migration plan.

The **Instruction Sequence** uniquely identifies the instruction. The recommendation is to use increments of 10 to allow insertion of other instructions in the future.

Select **Primary** for the first **Instruction Type**. All migration plans must contain one and only one primary instruction. All subsequent instructions require a **Subordinate** instruction type. In this case, the **Parent Instruction Sequence** must be entered. This number, used to maintain the defined relationships in the exported data, must match an instruction sequence number at a higher level in the hierarchy.

The instruction **Description** provides a business description of the instruction.

Select a **Business Object (BO)** to define the type of object from which data will be derived.

NOTE: Though BOs are specified in each instruction, it's important to understand that each BO is used only for filtering purposes. The migrated data set comprises the complete contents of the *maintenance object* that the business object structure is defined against. For a more detailed explanation of this, see [Understanding the BO Filtering Process](#).

NOTE: Refer to [Identifying Tables to Exclude From Migrations](#) for information about defining child tables to always exclude from a migration.

Traversal Criteria is used to define the relationship between each of the objects in a migration plan. The system provides three options to define how the child object is connected to the parent object so the system knows how to traverse from one object to another. **Traversal Criteria Type** options are **Constraint**, **SQL** and **XPath**. The following points explain each option:

- **Constraint** allows you to select a table constraint that represents a given record's relationship to another record in the system via a foreign key constraint defined in the meta-data. If **Constraint** is selected, the following additional fields are enabled:
 - **Constraint ID** is a unique identifier for the constraint. The search will show the valid table constraints for the MO of the instruction's BO and the MO of the parent instruction's BO.
 - **Constraint Owner** is used to define the owner of the constraint. This is populated automatically when selecting a constraint from the search.
- **SQL** lets you specify SQL join criteria between the parent instruction's object and the child object in the **SQL Traversal Criteria**. The syntax of the the traversal criteria is a WHERE clause (without including the word WHERE). When referring to a field on the parent instruction's object, use the syntax `#PARENT.TABLE_NAME.FIELD_NAME`. When referring to a field on the current instruction's object, use the syntax `#THIS.TABLE_NAME.FIELD_NAME`. For example, the following statement is used on a migration plan for Business Object, where the parent instruction is the BO and the subordinate instruction is used to reference the UI Map that is referred to as a BO option with the option type "FIDU":

```
#PARENT.F1_BUS_OBJ_OPT.BUS_OBJ_OPT_FLG = 'F1DU' AND @trim(#THIS.F1_MAP.MAP_CD) = @trim(#PARENT.F1_BUS_OBJ_OPT.BUS_OBJ_OPT_VAL).
```
- The **XPath** option lets you apply syntax in an XPath expression referencing elements in the instructions' referenced business objects. This is entered in the **XPath Traversal Criteria**. For example, the display map collection

statement in the SQL example noted above would be written as follows in XPath: `#this/mapCd = #parent/businessObjectOption/businessObjectOptionValue AND #parent/businessObjectOption/businessObjectOptionType = 'F1DU'`. This technique allows foreign key references that are mapped inside a CLOB to be referenced.

NOTE: The `#parent` expressions may access elements that are stored in the CLOB and described using `mapXML` and `mdField`. However, the `#this` expressions must refer to fields available in the business object using the `mapField` reference.

Defining **Next Migration Plan** provides the ability to indicate that in addition to copying the object defined in the instruction, any additional instructions included in that referenced migration plan will also be included in an export.

The **Algorithms** grid contains algorithms associated with each instruction. You must define the following for each algorithm:

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).
- Specify the **Sequence** and **Algorithm** for each system event. You can set the **Sequence** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the Sequence in which they should execute.

System Event	Optional / Required	Description
Import	Optional	Algorithms of this type may be used to adjust the data after it is moved to the target system. Refer to Adjusting Imported Data for more information. Click here to see the algorithm types available for this system event.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [F1_MIGR_PLAN](#).

Understanding the BO Filtering Process

Migration plan instructions require the definition of a business object to provide CMA with information about the record related to the instruction.

If the business object is the physical business object for the maintenance object, then CMA assumes that the instruction applies to all records that satisfy the traversal criteria. CMA recognizes the physical BO by comparing the BO to the value defined in the maintenance object option. If the business object defined is not the physical BO, then CMA will limit the records in the instruction to those that explicitly reference this BO or reference a child of this BO as its **identifying BO** value. (In other words, this BO must be in the parentage hierarchy of the records to be included in the instruction.)

NOTE: Unlike Bundling, CMA does not use the BO schema to drive what data is copied for a given record. The BO is only used as a filtering mechanism for selecting records. Refer to [Identifying Tables to Exclude from Migration](#) for information about how to ensure a child table is not included in a migration.

For example, if you define a migration plan for Master Configuration and use the physical business object for the instruction (**F1-MstCfgPhysicalBO**) then all master configuration records are considered for the instruction. If instead the business object you define is Migration Assistant Configuration (**F1-MigrationAssistantConfig**) then only the record related to this business object is included in the instruction.

Migration Plans for Objects with CLOB-Embedded Links

When migrating objects where foreign key references are captured in the object's XML based field, subordinate instructions are needed to define the foreign key references in order for CMA to understand the relationships. This is in contrast to direct

foreign keys where CMA can determine the relationships using constraints. The instructions provide two purposes. For wholesale migrations, where all data (or a large amount of data) is being migrated, the instructions allow CMA to group related objects into transactions. This helps in the apply process at import time to ensure that related objects are grouped together. However, the apply process includes iterative steps to try to overcome dependencies like this so defining the instructions is not critical for this type of migration. For piecemeal migrations, defining instructions ensures that the related objects are included in the migration, if appropriate.

The following are options for creating migration plans with XML-embedded links:

- One option is to use the specific logical (business) BO in the primary instruction to define the object you are copying. With this option, the subordinate instructions may use XPath criteria to define the related foreign key. When this approach is used, a separate Migration Plan must be created for each logical BO. (Refer to [Understanding the BO Filtering Process](#) for more information.) This option would only be used in isolated cases.
- Another option is to create a migration plan that uses the Physical BO as the primary instruction, and then include a subordinate instruction for the real logical BO, using SQL Traversal to join the object to itself by its primary key. Note that with this technique, the records that reference the logical BO will still only be included in the export file once. At this point further subordinate instructions may use XPath notation to define the foreign key data. Using the physical BO as the primary instruction ensures that all records in the MO are considered. The subordinate instructions with the logical BO and XPath notations will only apply to the records that are applicable to that BO. This option is useful for MOs that have a small number of logical business objects with disparate foreign keys.
- Another option is to use the physical BO in the primary instruction and use raw SQLs in the subordinate instruction's traversal criteria to identify the foreign keys using substring commands. A separate Subordinate Instruction is needed for each SQL corresponding to each element occurrence. Using this technique has the same advantages of the previous in that all records for the MO are included in the migration. However, this technique may be useful for maintenance objects with a larger number of business objects expected where each has one or more foreign keys. It's especially useful if many business objects reference the same foreign key. Then only one instruction is required for that foreign key. Note that a single migration plan may use this technique and the XPath technique for different elements.

A migration request may have multiple migration plans for the same maintenance object. That allows for some flexibility and long term maintainability in that the above techniques may be used in multiple migration plans. Consider the following example:

- A product provides base business objects with foreign keys defined in the XML field and provides the appropriate migration plan with instructions. An implementation extends this business object or perhaps creates their own business object for the same maintenance object and includes different additional foreign keys in the XML. Rather than duplicating the base migration plan and adding additional instructions for the additional foreign keys, the implementation can create a second migration plan for the MO with the additional foreign keys defined. A migration request should be defined to include both migration plans. In this case if the implementation has only one custom BO, they can choose to use the custom BO as the primary instruction as described above in the first option.

Defining a Migration Request

Migration Requests are unordered lists of Migration Plans that are to be migrated together. Algorithms, SQL statements, or specific keys are used to specify the set of objects you want to export. When complete, the request describes the complete data set that is extracted to the migration export file when the request is executed.

To view or define a migration request, navigate using **Admin > Migration > Migration Request**.

Use the **Migration Request Query** portal to search for an existing migration request. Once a migration plan is selected, you are brought to the maintenance portal to view and maintain the selected record.

Define one or more **Migration Plans** for the migration request. Note that the order doesn't matter. During the export process, CMA looks at all objects that qualify for the export and will group them together based on their relationships.

For each option, define an appropriate **Selection Type**, which is used to filter which records in the migration plan's object should be selected. The valid values are **Algorithm**, **SQL Statement**, and **Specific Key**.

- For the option **Algorithm**, specify an **Algorithm** for the Key Selection. The algorithm is responsible for returning a list of key values to include in the migration.

NOTE: The algorithms that are eligible to be plugged in here are those valid for the Migration Request — Select system event. Click [here](#) to see the algorithm types available for this system event.

- For the option **SQL Statement**, specify an **SQL Statement** for the key selection. may be used to limit the keys to include. To migrate all records in the table, the SQL statement "1=1" may be used. Refer to the inline help for other examples.
- For the option **Specific Key**, enter one or more primary keys to individually to define the records to include. Click the "+" to enter each additional key. Each row represents one key and supports a multi-part primary key.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [FI_MIGR_REQ](#).

Wholesale and Piecemeal Migrations

The Configuration Migration Assistant is used for two general types of migrations: wholesale and piecemeal. The following sections provide some additional information about these concepts.

Wholesale Migrations

Wholesale migrations are used when migrating all the configuration and/or administrative data from one environment to another. For example, a wholesale migration might be used when migrating administrative data from a development or test environment to a production environment.

A wholesale migration may be comprised of one or more migration requests that in total include all the administrative data to move. Whether one migration request is used or multiple are used depends on the following considerations:

- For each migration request used, a separate migration data set export record (and therefore separate migration data set import record) is needed. The multiple records may cause more user interaction with the data to progress all the records to their final state.
- On the other hand, splitting data into multiple migration requests, grouping information logically together may allow for more reuse.
- In addition, you should consider that the framework product provides base migration requests and your specific edge product may provide base migration requests as well that may or may not include framework migration plans. Using the product provided migration requests is beneficial with respect to maintenance. As features are added to the product (including new administrative maintenance objects), any impact on CMA should be included in the product owned migration requests. If your implementation introduces new custom administrative maintenance objects that should be included in CMA, then custom migration plans and a custom migration request should be added. Your implementation may choose to include only migration plans for your custom MOs (and simply migrate the objects in this table as a separate step in the process) or include other migration plans from the product in your custom migration plan to have a consolidated plan.

Migration plans used in wholesale migrations may be designed to omit subordinate instructions related to explicit foreign keys that are identified through constraints as they are not needed, assuming that the data they are referring to will also be included in the migration.

NOTE: Refer to [Framework Provided Migration Objects](#) for information about migration requests provided in the framework product. Refer to your specific product's documentation for information about addition base provided migration requests.

Piecemeal Migrations

A piecemeal migration refers to migrating a targeted subset of data from one environment to another. Examples of piecemeal migrations include:

- Migration of a new Business objects with its options and algorithms.
- Migration of a new maintenance portal, its zones, and its application service.
- Migration of all outbound message types.

Administrative users can define a migration request with the specific types of objects that should be copied and use selection criteria to limit the records to copy as desired.

The migration plans included in a ‘piecemeal’ migration request should be reviewed carefully to verify what subordinate instructions are included. When copying a specific type of record, there may be related records that should be copied as well and other related records that should not be copied. For example, when copying a custom To Do Type, perhaps any algorithms it refers to should be copied (along with their algorithm types) and its message should be copied, but not its navigation option or its drill keys, which would probably refer to base values.

Identifying Tables to Exclude From Migrations

Some maintenance objects that are eligible to be migrated may include child tables that should not be included in the migration. For example, if an object includes log tables, the entries in the log should reflect the actions on the object in that system, and will be different between the source system and the target system. If you have a custom Maintenance Object that includes tables you don't wish to migrate (such as a log table), use the **Non-Migrated Table** option on the MO to specify this table. All child records for this table will also be ignored during migration.

Another use case to consider is a child “many-to-many” table that connects two administrative objects and exists in the maintenance object of both tables. The child table may be in both MOs for convenience sake, but it may be that one MO is considered more of a “driver” object and the other more of a subordinate. If you are doing a targeted (piecemeal) migration where you want to copy a subset of objects, you may want to only copy the driver object and its children and their data but not their children. For example, in Oracle Public Sector Revenue Management, a Form Type includes a collection of valid Form Change Reasons and in turn the Form Change Reason refers to its Form Types. If an implementation wants to do a targeted migration of a form type and include all its related information, including form rules and form change reasons, it does not want the migration of a form change reason to in turn copy all its form types (and their data).

NOTE: The MO option must be set in both the Source and Target systems for a given MO.

Configuring Custom Objects for Migration

During the implementation and extension of the product, new custom administrative maintenance objects may be introduced. If your implementation would like to migrate records in those maintenance objects using the Configuration Migration Assistant, additional steps must be performed, which are highlighted in the following sections.

Physical Business Object

As described in [Understanding the BO Filtering Process](#), the migration plan requires a business object for its instruction. The business object is used to identify the records eligible for inclusion in the migration. Assuming your custom tables use one or more “logical” business objects for their processing, your implementation must decide if these business objects are appropriate for use by the migration plans, or if a physical BO is warranted. If so, create an appropriate physical BO.

Review MO Option Configuration

The following points highlight maintenance object (MO) configuration that should be reviewed or updated to support CMA:

- If a physical BO was created (above), link it to the MO as an option using the appropriate option type.
- Be sure that your MO defines an appropriate [FK Reference](#) and includes an Option on the MO that identifies the FK Reference. This is used by various portals and zones for CMA when showing detail about records being imported into the target region.
- As described in [Identifying Tables to Exclude From Migrations](#), an MO option is used to identify child tables for an MO that should never be included in a migration. If your custom maintenance object includes a standard Log table, then the recommendation is to list that table as an excluded table. Depending on the specific design of the maintenance object, there may be other child tables to define.

Characteristic Type Configuration

The CMA import process will attempt to create a log record for any administrative object that includes a log table. If your implementation has introduced any custom administrative tables that you plan to include in a migration request and it includes a log table, you must, to ensure that the log creation is successful, add your log table as a valid characteristic entity to the characteristic type **F1-MGO** (Migration Object).

Navigate using **Admin > Characteristic Type > Search** and select the characteristic type **F1-MGO**. Navigate to the **Characteristic Entity** tab and add a row to include the characteristic entity for your custom maintenance object's log table.

Standard CMA Configuration

Create one or more migration plans for the new object, depending on the type of data in the maintenance object and the types of migrations you envision:

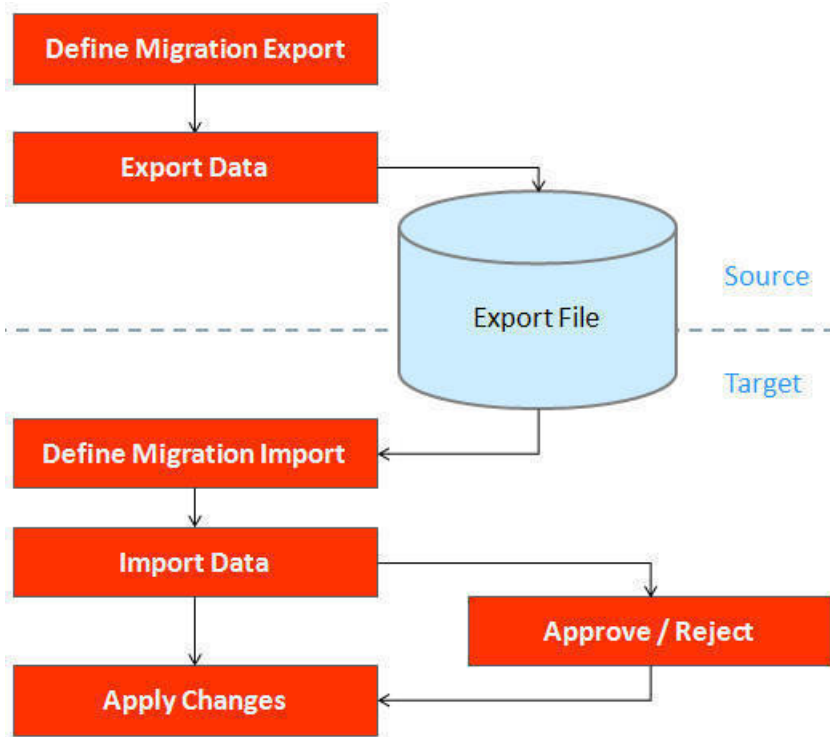
- If you have implemented only one “logical” business object used to define the data in the MO, then a single migration plan that references the this BO (or the maybe the MO’s physical BO) is appropriate.
- If you have implemented more than one “logical” business object, would the data for multiple business objects get copied together? Then perhaps a single migration plan that references the MO’s physical BO is appropriate.
- Are there additional foreign keys defined using mapXML in the business object(s) for the MO? If so, then it is recommended to include sub-instructions to define the links. At this point, if multiple “logical” BOs exist, your implementation may choose to define all the additional elements in the same migration plan or choose to define separate migration plans for each logical BO.
- Your implementation may decide to define more than one migration plan for the same type of record based on the types of migrations you plan to include. For example, you may decide to include a migration plan that copies only the records in this maintenance object. You may decide to define another migration plan that copies the records in this MO along with related records in another MO (for a special type of migration). Having said that, be sure to design the migration plan with reuse in mind to minimize maintenance efforts.

If your implementation has a template migration request to use for piecemeal or wholesale migrations, include the new migration plan(s) as appropriate.

CAUTION: Important! New migration plans and migration requests should follow naming conventions. Refer to [System Data Naming Convention](#) for more information.

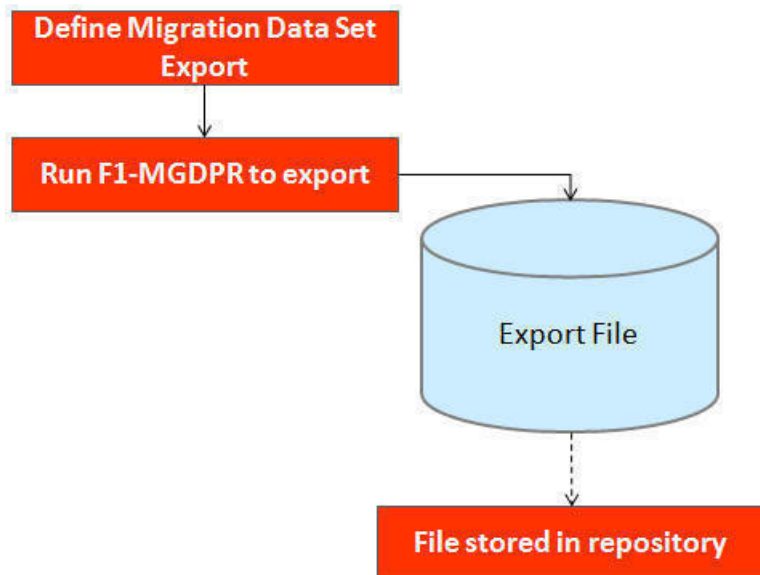
The CMA Execution Process

The following diagram illustrates a high-level view of the Configuration Migration Assistant execution process. The subprocesses illustrated here are described in more detail in the following sections.



Exporting a Migration

The migration export process begins in the source environment by defining a **Migration Data Set Export**, which specifies a defined **Migration Request** and provides a file name and description for the export file. After the data set is defined and saved, the **Migration Data Set Monitor** batch job can be submitted generate the export file.



The following topics provide more detail about this process.

Migration Data Set Export

To migrate data from one region to another, define a **Migration Data Set Export**. This establishes the export file name and identifies the migration request that includes the migration plans and selection criteria for the objects to include in the migration.

To view an existing migration data set export, navigate using **Admin > Migration > Migration Data Set Export**. Use the query criteria to locate the desired data set.

Note that you can also initiate the creation of an export data set from the [Migration Request](#) portal using the **Export** button. The export requires the name of an existing **Migration Request**.

Enter a unique **File Name** for the export. Do not use spaces in the name, and do not enter the file extension or a path. The output location and file extension of the intended export file, which should appear in the **Export Directory** and **File Suffix** labels, are defined in the [Migration Assistant Configuration](#) master configuration record.

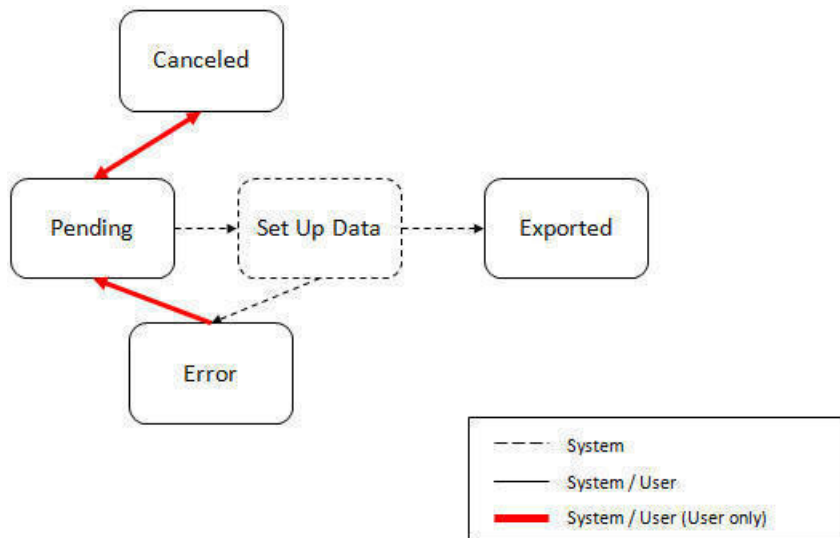
Enter an **Export Description** for the data set.

The **Source Environment Reference** is for information purposes. It should be populated with text that provides a meaningful description of the source environment. The default value is the URL of the source environment.

When the data set is complete, click **Save**.

Export Lifecycle

The following diagram describes the lifecycle of a Migration Data Set Export (data set).



The following points describe the lifecycle.

- The data set is created in **Pending** status.
- A user may choose to temporarily **Cancel** a pending data set to prevent it from being processed. The user can later return it to the Pending state when desired.
- The record remains in Pending until its monitor batch job is submitted. The **Migration Data Set Export Monitor** (F1-MGDPR) selects pending records and transitions them to **Set up data**. Refer to [Running Batch Jobs](#) for more information.
- Set up data is a transitory state that includes the algorithm that does the work of determining the objects to include in the export and group related objects together into a transaction. If everything is successful, the export file is written to the

appropriate file location and the record transitions to **Exported**. If an error is detected, the process stops and the record transitions to **Error**.

- If a record is in error and it is possible to correct the error, the record may be transitioned back to **Pending** to try again.

When the process is marked as **Exported**, the export file can be imported into the target system.

NOTE: The export process creates a file, providing the benefits of having a standalone file. It can be stored in a version control system for audit purposes or provided to others for import purposes.

CAUTION: Under no circumstances should exported data files be edited manually. Doing so could cause data corruption when the file is applied to the target environment.

NOTE: The export functionality is supported using the business object **Migration Data Set Export** (F1-MigrDataSetExport). The expectation is that implementations will use the delivered base business object and its logic and will have no reason to implement a custom business object for the CMA export process.

Importing and Applying a Migration

The import process is broken down into four general steps: Import, Compare, Approve, Apply. The following points provide an overview of the steps.

- **Import.** The first step covers importing the file and creating appropriate Migration Import records in the target environment to facilitate the subsequent steps.
- **Compare.** The compare step reviews each object that is in the import file and compares the object in the import against the equivalent record in the target environment. The comparison step results in noting which objects are unchanged, which are new (and the appropriate SQL to insert them) and which objects are changed (and the appropriate SQL to update them). Based on user configuration at import time, the objects that qualify for the import may be in a state that requires review or may be pre-approved.
- **Approve.** Once the comparison is complete, the user should review the results. There may be records marked for review. All of these records must be approved or rejected before the import can proceed. When the user is satisfied with the results of the comparison and has completed the review, the import is marked to proceed to the Apply step.
- **Apply.** This is the final step and is the step where the records in the target environment are added or updated. Because of potential high volumes of data and because of possible dependencies between records, this step supports two levels of attempting to apply the records. There is an apply step at the object level and an apply step at the transaction level. This will be described in more detail below.

Import Step

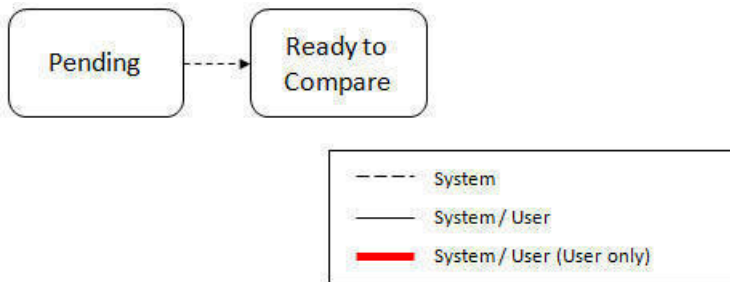
The import process starts with verifying the import directory configured in the [Master Configuration — Migration Assistant](#) for the target environment and ensuring that the exported file is located in that directory. Then, in the target environment, a **Migration Data Set Import** record should be created. The user indicates the file name.

In addition, the user decides what the default status should be for resulting objects.

- The **Default Status for Add** sets the default status for objects that are determined to be *new* during the import comparison process. The default is to automatically set new objects to **Approved** status. Other options are to set any new objects to **Rejected** or **Needs Review** status.
- The **Default Status for Change** sets the default status for objects that are determined to be *changed* during the import comparison process. As with new objects, the default for changed objects is **Approved**, with **Rejected** or **Needs Review** options available.

The file to import contains a list of all the objects included in the export. Any objects that the export step determined to be related have been grouped into “transactions”. Once the Migration Data Set Import (data set) is created, the next step is to read in the file and create Migration Transactions and Migration Objects.

The following is a portion of the Migration Data Set Import lifecycle as it pertains to the import step.



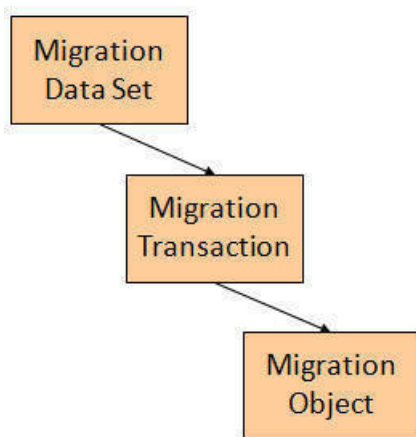
The following points describe the lifecycle.

- The data set is created in **Pending** status.
- The record remains in Pending until its monitor batch job is submitted. The **Migration Data Set Import Monitor** (F1-MGDIM) selects pending records and transitions them to **Ready to Compare**. Refer to [Import Process Summary](#) for more information.

The Ready to Compare state has an algorithm that is responsible for reading the related import file and creating the migration transactions and migration objects. The data set remains in this state until the comparison step is complete.

NOTE: A user may choose to **Cancel** a data set. Refer to [Cancelling a Data Set](#) for more information.

The following diagram highlights the relationships of the resulting migration import records.



The migration transaction and migration object each have their own lifecycle that will help manage the subsequent compare and apply steps. At the end of the import step, the status values of the three types of records are as follows:

- Migration Data Set Import is in the **Ready to Compare** state.
- Migration Transaction is in the **Pending** state.
- Migration Object is in the **Pending** state.

NOTE: The import functionality is supported using business objects supplied by the base product. The expectation is that implementations will use the delivered base business objects and their logic and will have no reason to implement a custom business objects for the CMA import process. The base business objects are **Migration Data Set Import** (F1-MigrObjectImport), **Migration Transaction** (F1-MigrTransactionImport) and **Migration Object** (F1-MigrObjectImport).

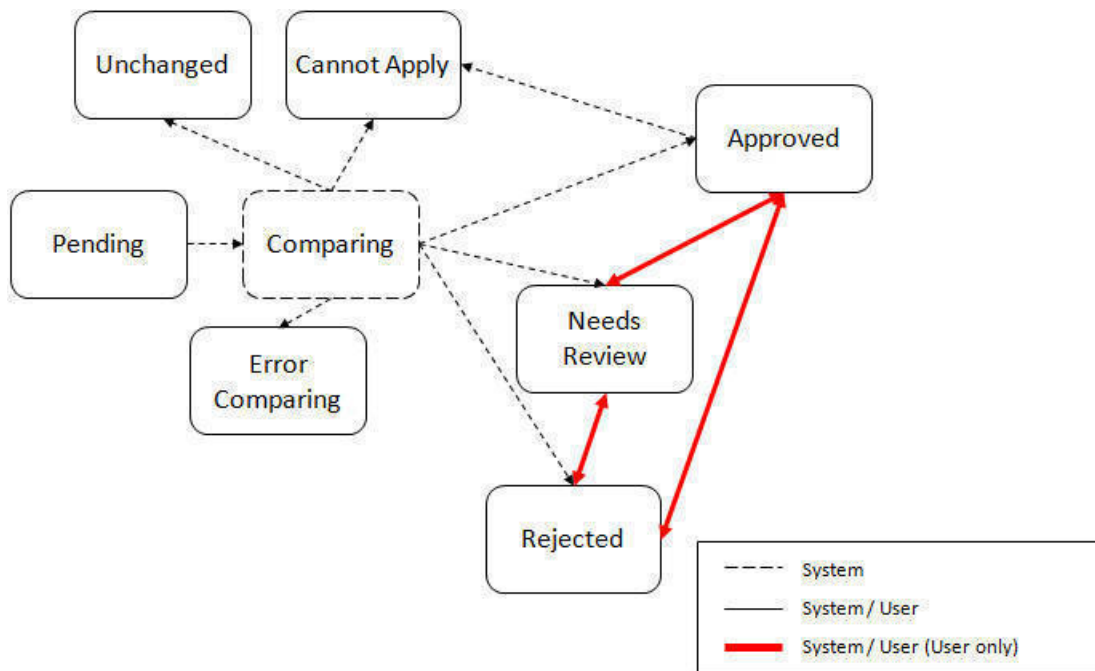
Compare Step

The import step results in the creation of one or more migration objects, one for each record selected in the export based on the export's migration request and its configuration. Related objects are grouped together in migration transactions. The next step in the import process is the Comparison step. In this step, the data captured by the import file for each object is compared to the view of that object in the target environment.

To cater for a possible large volume of objects, the comparison is done via a batch monitor. To aide in performance of the process, the monitor is performed on the migration objects so that it can be run multi-threaded. Once the objects are finished with the comparison, the migration transactions and the migration data set should be updated with an appropriate overall status before continuing to the next step. As a result, the comparison actually requires three steps: Migration Object Comparison, Migration Transaction Status Update and Migration Data Set Export Status Update. The steps are explained in detail in the following sections.

Migration Object Compare

This is the main step of the comparison. The **Migration Object Monitor** (F1-MGOPR) selects pending migration object records and transitions them to **Comparing**. This is a transitory state that includes the algorithm that does the work of comparing. There are various possible outcomes that could occur based on the logic in the algorithm. The following diagram illustrates a portion of the migration object lifecycle that pertains to comparison.



The following points describe the lifecycle.

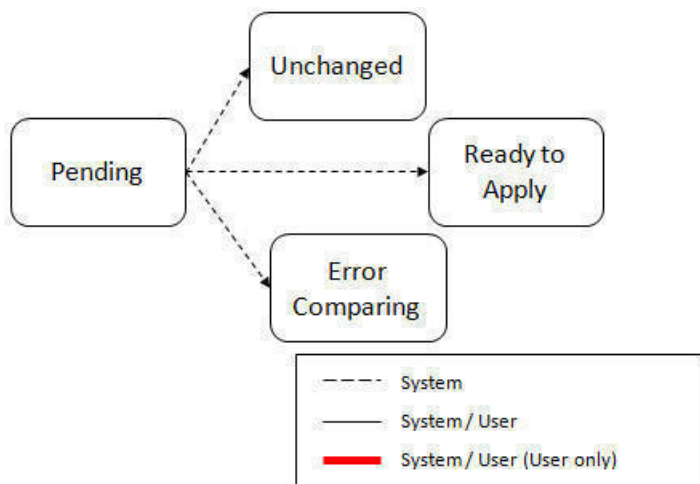
- When **Pending** records are selected by the monitor batch job, it transitions to **Comparing** where the algorithm will determine the appropriate next state.
- If the record in the migration object is found in the target environment and the data is exactly the same, the record transitions to **Unchanged**.
- If the record in the migration object is found in the target environment and the data is different, the algorithm generates the appropriate SQL to be used later in the Apply step to update the record and then transitions to **Approved**, **Needs Review** or **Rejected** based on the Default Status For Change setting captured on the Data Set.

- If the record in the migration object is not found in the target environment, the algorithm generates the appropriate SQL to be used later in the Apply step to insert the record and then transitions to **Approved**, **Needs Review** or **Rejected** based on the Default Status For Add setting captured on the Data Set.
- If there is any issue with attempting to parse the object data from the import, the record transitions to **Error Comparing**.
- If there is any reason that the imported object is not valid for import, the record transitions to **Cannot Apply**. The log will be updated with the error that caused the record to transition to this state. An example is that perhaps the record was exported in a different version of the product and has additional elements that are not recognized in this version.

NOTE: Refer to [Cancelling a Data Set](#) for information about cancelling a data set and its impact on its related objects.

Migration Transaction Status Update

After the import step, the migration transaction remains in the Pending state until all its objects have completed the comparison step. At that point, the status of the transactions should be updated based on the results of their objects. The **Migration Transaction Monitor** (F1-MGTPR) selects pending migration transaction records and runs its monitor algorithms. There are various possible outcomes that could occur based on the logic in the algorithms. The following diagram illustrates a portion of the migration transaction lifecycle that pertains to comparison.



The following points describe the lifecycle possible next states after Pending.

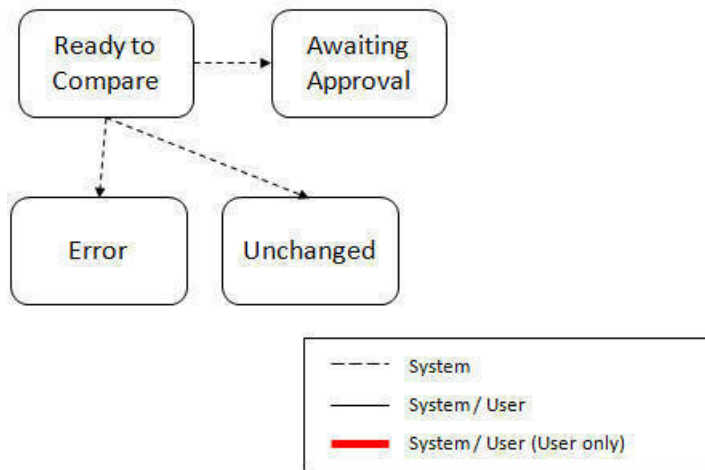
- If any related migration object is in the Error Comparing state, the transaction transitions to **Error Comparing**.
- If all related migration objects are in the Unchanged state, the transaction transitions to **Unchanged**.
- Otherwise, the transaction transitions to **Ready to Apply**. This means that at least one object is in an “apply-able” state.

The transaction remains in the **Ready to Apply** state until a user has approved the data set to move to the Apply step and the transaction’s related objects have attempted to apply themselves. This is described in more detail below.

NOTE: Refer to [Cancelling a Data Set](#) for information about cancelling a data set and its impact on its related objects.

Migration Data Set Import Status Update

Once all the objects and all transactions have been updated via the previous two steps, the migration data set export must be updated based on the results of their transactions. The **Migration Data Set Import Monitor** (F1-MGDIM) selects Ready to Compare data sets and runs its monitor algorithms. Note that this is the same monitor process that is used to select Pending data sets. There are various possible outcomes that could occur based on the logic in the algorithms. The following diagram illustrates the portion of the migration transaction lifecycle that pertains to comparison.



The following points describe the lifecycle possible next states after Ready to Compare.

- If any related migration transactions is in the Error Comparing state, the data set transitions to **Error**.
- If all related migration transactions are in the Unchanged state, the data set transitions to **Unchanged**.
- Otherwise, the transaction transitions to **Awaiting Approval**. This means that there are no errors and at least one object is in an “apply-able” state.

The data set remains in the **Awaiting Approval** state until a user decides that the data set and all its records are ready to progress to the Apply step.

NOTE: A user can choose to cancel a data set at any time while it is in progress. Refer to [Cancelling a Data Set](#) for more information.

Approval Step

Once the comparison is complete and the data set transitions to the Awaiting Approval state, a user needs to progress the data set to **Apply Objects** to trigger the Apply step. The following points describe steps a user may take during the approval step.

- If the data set configuration for the Default State for Add and Change was set to **Approved**, then any migration object that is determined to be eligible for the Apply step will be in the Approved state. In this situation, a user may want to review the data set and its transactions and objects to see verify that the results make sense. At that time, the user is able to move an object to Needs Review or Rejected as appropriate.
- If the data set configuration for the Default State for Add and Change was set to **Needs Review** for either option, then each migration object in the Needs Review state must be reviewed and the user must either Reject or Approve each object before moving to the Apply step.
- If the data set configuration for the Default State for Add and Change was set to **Rejected** for either option, the assumption is that the rejected records don’t need to be reviewed. But if a user finds a rejected record that shouldn’t be rejected, it may be transitioned to Approved (or Needs Review) as appropriate.

Once the user is comfortable with the data set’s results and no more objects are in the Needs Review state, the user should transition the record to **Apply Objects**. This will initiate the Apply step.

NOTE: Refer to [Maintaining Import Data](#) for details about the pages provided to help the user review a data set and its transactions and objects to help in the approval step.

NOTE: A user can choose to cancel a data set at any time while it is in progress.

Apply Step

The apply step is the step where records in the target environment are added or updated. Like the comparison step, the apply step is actually multiple steps to optimally handle high volume and dependencies between records as smoothly as possible. Before explaining the apply steps in detail, the following points highlight the type of data that may be included in a given data set.

1. Records that have no foreign keys and therefore no dependencies on other records. Examples: Message, Display Profile.
2. Records that have foreign keys that may already be in the target. Examples: Algorithms for existing algorithm types, To Do Roles for existing To Do Types.
3. Records that have foreign keys that are new records but also part of the migration. CMA detected the relationship at export time and grouped the objects in the same transaction. Example: Script-based Algorithm Type where the script is also in the migration.
4. Records that have foreign keys that are new records but also part of the migration. CMA did not detect the relationship. This may occur if the reference the foreign key is in a CLOB or parameter column and the migration plan did not include an instruction to explicitly define the relationship. Example, a Zone that references a visibility script.
5. Records that have circular references where both records are new and are part of the migration. CMA detected the relationship at export time and grouped the objects in the same transaction. Example: plug-in Script for a BO plug-in spot. The script references the BO and the BO references an algorithm for the script's algorithm type.

To handle high volume data, the first step in the apply process is to perform the apply logic at the migration object level via a multi-threaded batch job. This should result in all records in categories 1 and 2 above being applied successfully.

For records in categories 3 and 4 above, if a record with a foreign key is added or updated before its related record, the validation will fail. However, if the related record is added first and then the record referring to it is added, validation will pass. Because the migration objects are not ordered, the multi-threaded batch process may not process records in the desired order. To overcome this potential issue, the Apply step has special functionality, described in detail below.

For records in category 5 above, the circular reference will mean that the apply process at the object level will not successfully add or update these records. The apply process at the transaction level will cover these records. This is described in detail below.

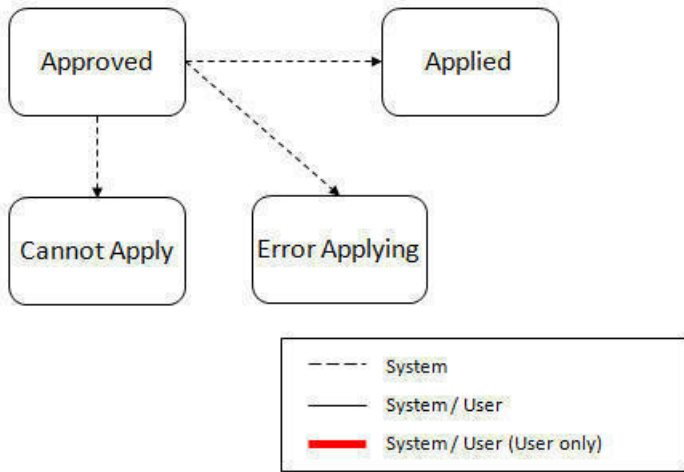
Apply Objects

Once the Data Set is in the state of **Apply Objects**, the **Migration Object Monitor — Apply** process (F1-MGOAP) runs to attempt to apply the objects. The background process in conjunction with the Apply algorithm have special functionality to ensure records in categories 3 and 4 (above) successfully apply during this step:

- The **Migration Object Monitor — Apply** process is a special one that continually re-selects records in the **Approved** state until there are no more eligible records to process.
- When an error is received in the Apply Object algorithm, the algorithm increments an “iteration count” on the migration object record. If the iteration count does not exceed a maximum count (noted in the algorithm), the object remains in the **Approved** state and is eligible to be picked up for processing again. If the iteration count exceeds the maximum defined in the algorithm, the record transitions to the **Error Applying** state.

NOTE: When submitting this Apply batch job, be sure to set the number of threads to a number that does not exceed the number of threads supported by the thread pool worker. Doing this will cause the ‘excess’ threads to wait for the supported number of threads to finish, erasing the benefit of the iteration processing.

The following diagram is the portion of the migration object lifecycle that pertains to the Apply step. Note that this diagram is not complete. A subsequent section provides more information about resolving errors.



At the completion of the Apply monitor process, typically the objects will be in the **Applied** state or the **Error Applying** state. The records in the Error Applying state are in that state for one of two reasons.

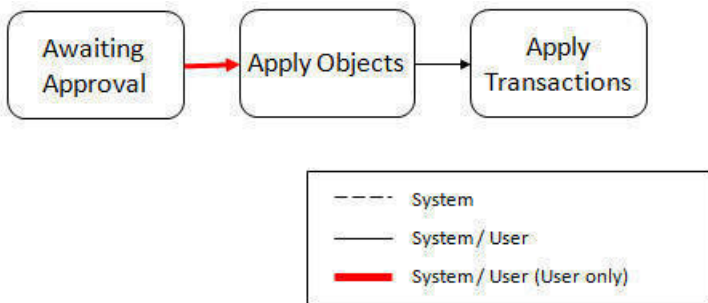
- They are in category 5 described above where the records have a circular reference with another record. For this scenario, the Apply Transactions step described below should successfully apply the records.
- There is some other error that is unrelated to the records in the current migration. In this case, manual intervention is required. Refer to the Resolving Errors section below for more information.

As shown in the diagram, the Apply Objects algorithm may also detect a reason that the object cannot be applied. This may occur if the object in the target environment has been updated since the comparison step, making the SQL captured at that point no longer applicable. If this occurs, after the current migration is fully applied, the original file may imported again, and new comparisons can be generated and applied.

Apply Transactions

Ideally, after the Apply Objects step, all the objects are **Applied** or are in **Error Applying** due to the “circular reference” situation. The typical next step is to turn over responsibility to the transactions. The migration transactions can then attempt to apply their objects in bulk.

In order to ensure that multiple background processes are not trying to select migration objects to run the Apply step, the transactions are only eligible to attempt to “apply my objects” if the Data Set is in the **Apply Transactions** state.



A monitor algorithm (executed by the data set monitor batch process) on the Apply Objects state checks to see if all migration objects are no longer **Approved** and do not have any records in **Error Applying**. If so, it automatically transitions the record to the **Apply Transactions** state.

If any objects are in **Error Applying**, the monitor algorithm does not automatically transition the record. In that case, a user must decide if the record should transition to **Apply Transactions**. The Resolving Errors section below describes alternative steps that the user may take if there are errors.

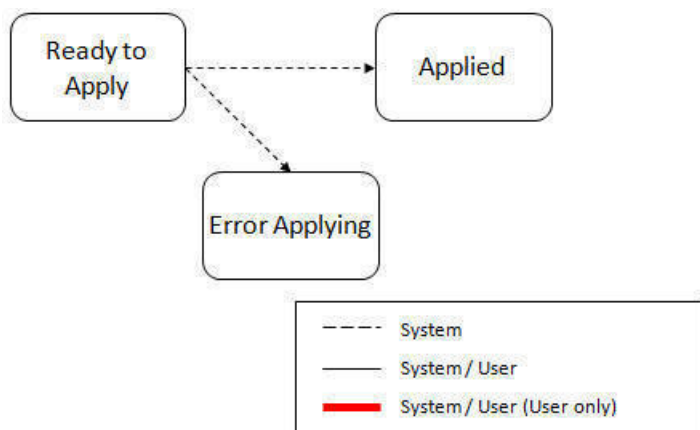
Once the Data Set is in the state of **Apply Transactions**, the **Migration Transaction Monitor — Apply** process (F1–MGTAP) runs. It attempts to apply the transaction’s objects. If no migration objects are in error, the migration transaction

simply transitions to **Applied**. If any of the migration objects are in **Error Applying**, the background process and the Apply algorithm have special functionality to try to overcome dependencies in migrated objects:

- The Apply algorithm selects all migration objects in error and performs all their SQL, then validates all the records. If there are objects in the transaction with circular references, they should pass validation at this point.
- Because there may still be some dependencies across transactions, similar error handling described in the Apply Objects step occurs here. When an error is received in the Apply Transaction's Object algorithm for any of the objects in the transaction, the algorithm increments an "iteration count" on the migration transaction record. If the iteration count does not exceed a maximum count (noted in the algorithm), the transaction remains in the **Ready to Apply** state and is eligible to be picked up for processing again. If the iteration count exceeds the maximum, the record transitions to the **Error Applying** state. Note that if any objects in the transaction are in error, none of the objects are applied. They all remain in error.
- The **Migration Transaction Monitor — Apply** process is a special one that continually re-selects records in the **Ready to Apply** state until there are no more eligible records to process.

NOTE: When submitting this Apply batch job, be sure to set the number of threads to a number that does not exceed the number of threads supported by the thread pool worker. Doing this will cause the 'excess' threads to wait for the supported number of threads to finish, erasing the benefit of the iteration processing.

The following diagram is the portion of the migration transaction lifecycle that pertains to the Apply step illustrating the points above. Note that this diagram is not complete. A subsequent section provides more information about resolving errors.



If at the end of the transaction level Apply process there are transactions in error (and therefore there are still objects in error), a user must review the errors and determine how to fix them. Refer to the Resolving Errors section below for more information.

Resolving Errors

As mentioned in the previous sections, errors may be received after the Apply Objects process runs and the system will not automatically transition the data set to the **Apply Transactions**. Rather, a user must make the decision. This allows for the user to review the errors and make a decision:

- If the errors appear to be dependency related, the user can decide to let the "transactions apply their objects" and transition the data set to **Apply Transactions**, described above.
- If the errors appear to be related to an outside issue that can be manually resolved, the user may choose to fix the issue and redo the Apply Objects step.
- The user may also decide to reject one or more objects to remove them from the migration.

After the Apply Transactions step, if there are still errors, a user must review the records and determine how to proceed

- The user may decide to reject one or more objects to remove them from the migration.
- The user may manually resolve an issue external to the migration and then decide to do one of the following:
 - Redo the **Apply Objects** step. This is recommended if there are a large number of Objects still in error and not a large number of dependencies expected. The benefits of running the Apply Objects multi-threaded will ensure that the process runs efficiently.
 - Redo the **Apply Transactions** step.

Because the objects and transactions are in Error Applying, in order to “retry” the Apply step after manually fixing an error, the system needs to move the records back to the state that allows them to be picked up by the appropriate Apply process. For migration objects, records need to be moved back to **Approved**. For migration transactions, records need to be moved back to **Ready to Apply**. This is accomplished with a Retry Count. This is different from the Iteration count that is used during the apply steps to handle the iterative selection logic. The following points describe the Retry logic for migration objects.

- When the Migration Data Set Import record first enters the **Apply Objects** state, its Object Retry Count is set to 1.
- When Migration Objects first enter the **Approved** state, their Object Retry Count is set to 1.
- When migration objects transition to **Error Applying**, it remains in this state as long as the object’s Retry Count matches the data set’s Object Retry Count (or it is successfully applied via the Apply Transactions step).
- If a user decides to **Retry Objects** (using an action button on the Migration Data Set Import page), the data set’s Object Retry Count is incremented and the Data set returns to the **Apply Objects** state.
- At this point, the monitor on the **Error Applying** state for the objects detects that the object’s Retry Count doesn’t match the data set’s Object Retry Count and that triggers the transition back to **Approved**. This increments the object’s retry count to match the data set’s. Now the objects are eligible to be picked up by the object level Apply process.

Analogous logic exists for the migration transactions.

- When the Migration Data Set Import record first enters the **Apply Transactions** state, its Transaction Retry Count is set to 1.
- When Migration Transactions first enter the **Ready to Apply** state, their Transaction Retry Count is set to 1.
- When migration transactions transition to **Error Applying**, it remains in this state as long as the transaction’s Retry Count matches the data set’s Transaction Retry Count.
- If a user decides to **Retry Transactions** (using an action button on the Migration Data Set Import page), the data set’s Transaction Retry Count is incremented and the Data Set returns to the **Apply Transactions** state.
- At this point, the monitor on the **Error Applying** state for the transactions detects that the transaction’s Retry Count doesn’t match the data set’s Transaction Retry Count and that triggers the transition back to **Ready to Apply**. This increments the transaction’s retry count to match the data set’s. Now the transactions are eligible to be picked up by the transaction level Apply process.

Finalize Apply Step

Once all the migration objects for a migration transaction are in a final state (**Applied**, **Rejected** or **Cannot Apply**), the migration transaction transitions to the **Applied** state. Once all the migration transactions are in the **Applied** state, the Migration Data Set record transitions to the **Applied** and the import is complete.

NOTE: To review the full lifecycle for each record, refer to the Business Object — Summary tab in the application metadata for the base business objects **Migration Data Set Import** (F1-MigrObjectImport), **Migration Transaction** (F1-MigrTransactionImport) and **Migration Object** (F1-MigrObjectImport).

Adjusting Data While Importing

Some records may have data that is specific to the environment it is in and won't apply in the target environment. In such cases, an algorithm plugged into the migration plan instruction may be used to adjust the data when importing. This algorithm is executed in the apply step after the record is inserted but before the validation is executed.

Some examples of records that may require import algorithms.

- Batch Control references its next batch sequence number. This number is only relevant with respect to its environment. The instruction for a batch control can include an algorithm to not overwrite the batch sequence number when copying a batch control.
- Some products include administrative objects that reference a master data object. Master data objects are not copied as part of CMA. An import algorithm may be used to adjust the referenced master data foreign key when importing, for example to reset it (or not overwrite when updating). If the algorithm knows how to find the appropriate master data record to link, that may also be included.

Import Process Summary

The following table summarizes the steps required to complete the import process from start to finish. It highlights what steps are manual and what steps are performed by a batch monitor process. For the Apply steps, there are two parts where multiple next actions are possible. The possible next actions have the same sequence with a letter following the sequence. For each step, the table highlights the Next Action sequence that would occur.

NOTE: When running the Apply batch jobs, be sure to set the number of threads to a number that does not exceed the number of threads supported by the thread pool worker.

Also note that a sequence and action marked in bold is considered the “normal path”.

Step	Seq	Action	Manual / Batch	Portal — Action	Batch Control	Record Impacted — Resulting Status	Next Action Sequence
Import	10	Create Import record	Manual	Migration Data Set Import - Add		Migration Data Set Import - Pending	11
Import	11	Import file	Batch		F1-MGDIM - Migration Data Set Import Monitor	Migration Data Set Import - Ready to Compare Migration Transaction - Pending Migration Object - Pending	20
Compare	20	Migration Object Compare	Batch		F1-MGOPR - Migration Object Monitor	Migration Object - Approved, Needs Review, Rejected, Unchanged or Error Comparing	21
Compare	21	Migration Transaction Status Update	Batch		F1-MGTPR - Migration Transaction Monitor	Migration Transaction - Ready to Apply, Unchanged or Error Comparing	22

Step	Seq	Action	Manual / Batch	Portal — Action	Batch Control	Record Impacted — Resulting Status	Next Action Sequence	
Compare	22	Migration Data Set Export Status Update	Batch			F1-MGDIM - Migration Data Set Import Monitor	Migration Data Set Import - Awaiting Approval, Unchanged or Error	30
Approval	30	Review comparison results, approve / reject as needed	Manual	Migration Data Set Import, drill in to the Transactions / Objects as necessary.			Migration Object - Approved or Rejected (no records should be in Needs Review) Migration Data Set Import - Apply Objects	40
Apply	40	Apply Objects	Batch			F1-MGOAP - Migration Object Monitor - Apply	Migration Object - Applied or Error Applying	41
Apply	41a	Data Set Transition - auto transition to Apply Transactions. Only applicable if no migration objects are in Error Applying	Batch			F1-MGDIM - Migration Data Set Import Monitor	Migration Data Set Import - Apply Transactions	42
Apply	41b	Data Set Transition - manual transition to Apply Transactions. Occurs if user reviews errors and determines that they may be resolved in the Apply Transaction step.	Manual	Migration Data Set Import - click Apply Transactions			Migration Data Set Import - Apply Transactions	42
Apply	41c	Data Set Transition - manual transition to Retry Objects. Occurs if user decides to fix an external error and wants to try the batch level Apply again.	Batch	Migration Data Set Import - click Retry Objects			Migration Data Set Import - Apply Objects	44
Apply	42	Apply Transactions	Batch			F1-MGTAP - Migration Transaction Monitor - Apply	Migration Transaction - Applied or Error Applying Migration Object - Applied or Error Applying	43

Step	Seq	Action	Manual / Batch	Portal — Action	Batch Control	Record Impacted — Resulting Status	Next Action Sequence
Apply	43a	Data Set Transition - auto transition to Applied Applicable if all transactions are Applied	Batch		F1-MGDIM - Migration Data Set Import Monitor	Migration Data Set Import - Applied.	N/a
Apply	43b	Data Set Transition - manual transition to Retry Objects. Occurs if user decides to fix an external error and wants to try the batch level Apply again at the Object level.	Manual	Migration Data Set Import	- click Retry Objects	Migration Data Set Import - Apply Objects	44
Apply	43c	Data Set Transition - manual transition to Retry Transactions. Occurs if user decides to fix an external error and wants to try the batch level Apply again at the Transaction level.	Manual	Migration Data Set Import	- click Retry Transactions	Migration Data Set Import - Apply Transactions	45
Apply	44	Move Objects from Error Applying back to Approved. Occurs if user chose to Retry Objects.	Batch		F1-MGOPR - Migration Object Monitor	Migration Object - Approved	40
Apply	45	Move Transactions from Error Applying back to Ready to Apply. Occurs if user chose to Retry Transactions.	Batch		F1-MGTPR - Migration Transaction Monitor	Migration Transaction - Ready to Apply	42

The following table summarizes the batch monitor jobs that are used in the import process. You can see that there are special monitor processes for the Apply step for both the Object and Transaction. However, for all other states that have monitor logic, the standard monitor process for that MO is used.

Batch Control	Description	Comments
F1-MGDIM	Migration Data Set Import Monitor	Processes data set records in the following states: <ul style="list-style-type: none"> Pending

Batch Control	Description	Comments
		<ul style="list-style-type: none"> Ready to Compare Apply Objects Apply Transactions
F1-MGTPR	Migration Transaction Monitor (Deferred)	Processes transaction records in the following states: <ul style="list-style-type: none"> Pending Error Applying
F1-MGTAP	Migration Transaction Monitor - Apply	Processes transaction records in the Ready to Apply state where the data set is in the Apply Transactions or Canceled state. <p>NOTE: Be sure to set the number of threads to a number that does not exceed the number of threads supported by the thread pool worker.</p>
F1-MGOPR	Migration Object Monitor	Processes object records in the following states: <ul style="list-style-type: none"> Pending Needs Review (check for Data Set cancellation) Rejected (check for Data Set cancellation) Error Applying
F1-MGOAP	Migration Object Monitor - Apply	Processes object records in the Approved state where the data set is in the Apply Objects or Canceled state. <p>NOTE: Be sure to set the number of threads to a number that does not exceed the number of threads supported by the thread pool worker.</p>

Refer to [Running Batch Jobs](#) for more information about managing the batch jobs.

Cancelling a Data Set

A user may choose to **Cancel** a data set to prevent it from being processed at any point during the process.

If related migration transactions or migration objects have already been created, they will not be canceled as part of the data set getting canceled (due to possible high volumes of related records). They will be canceled the next time an appropriate monitor batch process runs. The child records checks to see if the data set has been canceled prior to any state transition.

Additional Note Regarding Imports

The following points describe miscellaneous comments related to Migration Import.

- CMA relies on the fact that database referential integrity constraints are not in place, and that the SQL statements can be run in any order within the transaction. Any archiving solution that requires referential integrity constraints (such as Information Lifecycle Management) would not be possible on this data. Given that CMA migrations comprise administrative data and not transactional data, this should be a reasonable exception.
- The validation that is performed is only via the **Page Validate** service. BO validation algorithms are not executed. Page validation does not include validation of the business object against the schema (for example, for required fields, field sizes, etc.).
- All migration requests can be exported at the same time. On the import side, however, you should consider importing, reviewing, and applying an entire file/data set before moving on to the next one. The reason is that if objects are included

in more than one file, two sets of "inserts" will be generated, but only the first will succeed. The second will cause an insert error on the object, and the transaction would be marked with status "Error Applying". If instead you wait until the first file is applied before importing the second, the second data set will not generate any SQL for the object, since it has already been inserted. It's a matter of efficiency: If you first import all files and then try to apply all, you'll have to identify the duplicated object as an error and then mark the object as rejected before applying the transaction.

Caching Considerations

Because CMA updates administrative data that is usually read from a cache, a thread pool with the L2 cache disabled is required during the execution stages of migration. Contact your system administrator and ask for the name of a thread pool with the L2 cache disabled. This thread pool name must be entered in the batch controls used for CMA processing in the **Thread Pool Name** parameter.

Refer to [Running Batch Jobs](#) for more information about running CMA batch jobs.

The batch section of the *Server Administration Guide* also contains information on preparing and implementing thread pools and turning off the L2 cache.

Also note that after a successful migration, the target region now has new administrative data which needs to be part of the L2 cache. The existing thread pool workers should be restarted by your system administrators. In addition, it is recommended to restart the application server.

Maintaining Import Data

This section describes the portals provided to add, view and maintain migration import data.

Migration Data Set Import

Use the Migration Data Set Import portal to view and maintain migration data set import records. Navigate using **Admin > Migration > Migration Data Set Import**.

Migration Data Set Import Query

Use the query portal to search for an existing Migration Data Set Import record. Once a record is selected, you are brought to the maintenance portal to view and maintain the selected record.

In addition, the query provides an option to specifically search for data sets that have either objects in error or transactions in error.

Migration Data Set Import Portal

This page appears when viewing the detail of a specific migration data set import record.

The topics in this section describe the tabs that appear on the portal.

Migration Data Set Import - Main

This portal appears when a migration data set import record has been selected from the query portal. The topics below describe the base-package zones that appear on the portal.

Migration Data Set Import

The Migration Data Set Import zone contains display-only information about the selected record.

Please see the zone's help text for information about this zone's fields.

Migration Data Set Transactions

This zone is visible once the [Import Step](#) has occurred and lists all the transactions that are related to the data set. To see more information about a specific migration transaction, click the hypertext for its ID. This brings you to the [Migration Transaction](#) portal.

Migration Data Set Impacted Object Summary

This zone is visible once the [Import Step](#) has occurred and lists the objects that are related to the data set. To see more information about a specific migration object, click the hypertext for its ID. This brings you to the [Migration Object](#) portal.

Migration Data Set Objects in Error

This zone is only visible if the data set's status is **Apply Objects**, and there are objects in the status **Error Applying**. It indicates the error for each object. A user may use this zone to review errors after the monitor batch job to apply objects completes. Using the error information shown, the user can choose to transition the record to **Error Applying** or manually fix the cause of the error and click **Retry Objects**.

NOTE: Refer to [Apply Step](#) for more information about resolving errors.

Migration Data Set Transactions in Error

This zone is only visible if the data set's status is **Apply Transactions**, and there are transactions in the status **Error Applying**. It indicates the error for each object. A user may use this zone to review errors after the monitor batch job to apply transactions completes. The errors received when attempting to apply objects at the transaction level may differ from those received when attempting to apply objects at the object level. A transaction log is created for each object error received and these exceptions are shown in this zone.

NOTE: Refer to [Apply Step](#) for more information about resolving errors.

Migration Data Set Import - Log

Navigate to the Log tab to view the [logs](#) for the migration data set import record.

Migration Transaction Portal

This page appears after drilling into a specific migration transaction from the migration data set portal or from the migration object portal.

The topics in this section describe the tabs that appear on the portal.

Migration Transaction - Main

The topics below describe the base-package zones that appear on the portal.

Migration Transaction

The Migration Transaction zone contains display-only information about the selected record.

Please see the zone's help text for information about this zone's fields.

Migration Transaction Objects

This zone lists the objects that are related to the data set. To see more information about a specific migration object, click the hypertext for its ID. This brings you to the [Migration Object](#) portal.

Migration Transaction - Log

Navigate to the Log tab to view the [logs](#) for the migration transaction.

Migration Object Portal

This page appears after drilling into a specific migration object from the migration data set portal or from the migration transaction portal.

The topics in this section describe the tabs that appear on the portal.

Migration Object - Main

The topics below describe the base-package zones that appear on the portal.

Migration Object

The Migration Object zone contains display-only information about the selected record.

Please see the zone's help text for information about this zone's fields.

Migration Object - Log

Navigate to the Log tab to view the [logs](#) for the migration object.

Running Batch Jobs

There are several batch jobs that are part of the CMA process, especially the import step. And in some cases, a single batch jobs may process multiple states in the same business object lifecycle. Implementations must decide the best way to manage the batch job submission depending on how they plan to work.

- **Batch scheduler.** If an implementation wishes to put these batch jobs in the batch scheduler, a given job may need to be included several times to manage progressing the records to completion.
- **Timed Batches.** The batch controls can be configured as timed batches so that they run every N minutes based on the setting. This allows for the batch jobs to run periodically and process whatever is ready. A user doesn't have to manually submit a batch request. Navigate to **Admin > Batch Control > Search** and select the appropriate batch controls. For each one, change the Batch Control Type to **Timed**. Fill in the additional information that appears for timed batches.
- **Manual submission.** The user managing the CMA import process submits the appropriate batch jobs on demand when a particular step is ready. Navigate using **Menu > Batch > Batch Job Submission > Add**, select the appropriate batch control and fill in the parameters as needed.

Note that for the CMA batch processes, especially the "apply" batch processes, the setting of the **Thread Pool Name** parameter is important. This parameter specifies the thread pool that will be executed by the batch. Since the CMA apply processes update administrative data that is usually read from a cache during batch processing, the thread pool named here must be one with the L2 cache disabled. In addition, after successfully applying the migrated data, the L2 cache of all thread pools must be refreshed. For more information, see [Caching Considerations](#).

CAUTION: Be sure that the Thread Count set when submitting the batch job does not exceed the number supported by the thread pool. Otherwise the extra threads will simply wait until the supported number of threads are finished.

Refer to the parameter descriptions in the batch control metadata for more information about filling in the parameters.

For additional details on submission controls, refer to the topic [Batch Job Submission - Main](#) in the Batch Jobs section.

CMA Reference

This section provides additional reference information.

Framework-Provided Migration Configuration

This topic describes special information relating to migration objects provided for use by CMA in the product. Additional objects may be provided by your specific product. Any special information for objects is provided separately in each product's documentation.

You can see all currently available objects and their descriptions by performing a search in the **Migration Plan Query** or **Migration Request Query** pages.

The following points highlight some information about the Framework-provided migration plans and migration requests:

- Fields and characteristic types are not migrated with the object unless specifically indicated.
- The **Application Service** used by an object is migrated only if it is CM-owned.
- The **F1-Language** migration plan should **never** be used in a migration request with other objects. Defined constraints will cause CMA to link all objects with a language table—which is almost all of the system admin data—into one large transaction. Attempting to update and commit all the changes for this one large transaction may cause performance issues and/or out of memory errors. For this reason, this MO has its own migration request, **F1-Languages**.
- The migration request **F1-SchemaAdmin** defines migration plans related to data that is commonly used in schema-based objects.
- The **Batch Control** object optionally references a User. If this user does not exist on the target system, CMA cannot apply the requested changes.
- The base migration plans for MO and BO include instructions to copy option types that use foreign key references to refer to other objects. Note that the data stored in the options are not validated, so defining these instructions is not required when doing wholesale migrations. However, including subordinate instructions for foreign key references is useful for piecemeal migrations to ensure that the related data is included in the migration. If you add additional MO or BO option types that use foreign keys and you want to support piecemeal migrations, you must create custom migration plans and requests for MO and BO, respectively to include these referenced objects in the migration plan. Note that you do not need to duplicate the instructions in the base migration plans. You may define the additional migration plans to only have the additional custom option types. When submitting a migration request for MO or BO you must include both the base migration plans and the custom migration plans in the request.
- The migration plans provided by CMA migrate scripts, schema-based objects and zones, and, through constraints, some of the associated data associated with them. However, data specified through alternate formats (such as through **Edit Data** steps in scripts, referenced in schemas for schema-based objects, or data from mnemonics in zone parameters, etc.) are not identified and combined in the same transaction. It's important to note that this could cause validation errors during **Apply**, and may require retrying or migrating the additional data separately. See [Apply Step](#) for a description of how to reapply the import should an **Applied With Errors** result occur.
- There are two migration plans for **Scripts**. The migration plan **F1-ScriptOnly** migrates just the script and its **Application Service** (provided the Application Service is CM-owned). This is used in the **F1-FrameworkAdmin** migration request, which migrates all admin objects. The migration plan **F1-Script** includes most related objects, but does not migrate any objects referenced in the edit data area steps. It does not move the **Function** maintenance object (which has been deprecated). This migration plan is not included in any base migration requests. It may be included in any appropriate custom piecemeal migration request where scripts and related data should be migrated.
- If your implementation includes a **Feature Configuration** setting for the **F1_DBCONINFO** entry that will be included in a migration request, be sure that the import user on the target region has the appropriate security rights to this entry (**Super User** access mode for the Feature Configuration application service (**CILTWSDP**)).
- All of the provided Framework-owned migration requests include the SQL statement "1=1" so that all records in the plan are migrated. If an implementation wants to limit a migration to a subset of records, the prepackaged migration request may be duplicated and then customized to modify the key selection criteria.
- Many of the XAI-related maintenance objects include references to environment-specific data, and data should be migrated with extreme care. Migration plans are not provided for MOs that contain mostly environment-specific data. Migration plans are provided for the following XAI-related maintenance objects, but these plans are not included in the **F1-XAI-Admin** migration request: **External System**, **XAI Receiver**, **Message Sender**, and **XAI Route Type**.

Migration Requests for Wholesale Migrations

The following framework provided migration requests may be used in a wholesale migration. Refer to your specific product's CMA documentation for its recommendation on which migration requests to use for a full migration of framework and product administrative tables.

- The **F1-SchemaAdmin**(Framework Schema Admin) migration request. This request contains migration plans for objects needed by schema based objects, such as Field and Lookup. It has few dependencies on its data and many other objects include dependencies to it.
- The **F1-FrameworkAdmin** (Framework Admin) migration request. This includes migration plans for most widely used maintenance objects Your product may have chosen to incorporate migration plans from this migration request directly into one of its owned migration requests.
- The **F1-GeneralSystemOptions** (General System Objects) migration request. This includes migration plans for business related objects that may or may not be applicable to all products. Your product may have chosen to incorporate a subset of the migration plans from this migration request directly into one of its owned migration requests.

For each migration request determined to be needed for a wholesale migration, create a migration data set export record and process it as documented in **Exporting a Migration**.

Executing Piecemeal Migrations

Piecemeal migrations involve copying a targeted set of data. It may be that it includes a subset of maintenance objects (for example all Activity Types or all Asset Types) or it may further limit the records within one or more maintenance objects (for example, all ‘electric’ rates). Many piecemeal migrations are ad-hoc, depending on the specific requirements at the time.

If your product has provided migration requests for piecemeal migrations out of the box, it would they would be defined to copy the appropriate type of data. However, the selection criteria would most likely not be limited to a type of record. As such, it could be used as is for a “copy all data in these maintenance objects” migration. For example, the framework provides a migration request for Security Configuration. This may be used to copy all the data in the various security tables. Oracle Public Sector Revenue Management provides a migration request to copy form types and its related data.

If your implementation wants to copy only a subset of data for maintenance objects covered by a base provided migration request, do the following:

1. Duplicate the base migration request.
2. Find the migration plan for the object or objects where you want to limit the data. Enter appropriate Key Selection criteria to select the desired data.
3. Use that migration request to request a migration data set export request.

If your piecemeal migration requirement is not satisfied by a base migration request that may be used as a starting point, review the base migration plans for the records that should be copied. Determine if the migration plans have the desired subordinate instructions to copy (or not copy) the related data as desired.

- If existing migration plans satisfy your requirements, create a migration request that includes those plans (and desired key selection as needed). Use that migration request to create a migration data set export request.
- If existing migration plans do not satisfy your requirements, create migration plans as needed. Create a migration request that includes those plans (and desired key selection as needed). Use that migration request to create a migration data set export request.

Configuring Facts

Fact is an optional configuration tool for simple workflow-type business messages or tasks. The base package does not provide a dedicated Fact user interface because fact is generic by design. Implementations configure their own user interface to visualize the desired custom business process. The topics in this section describe the generic Fact entity and how it can be customized.

Fact Is A Generic Entity

The Fact maintenance object is a generic entity that can be configured to represent custom entities and support automated workflows for a variety of applications. Each fact references a business object to describe the type of entity it is. A status column on the fact may be used to capture its current state in the processing lifecycle controlled by its business object.

The maintenance object also supports a standard characteristic collection as well as a CLOB element to capture additional information.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [FI_FACT](#)

Fact's Business Object Controls Everything

A fact's business object controls its contents, lifecycle and various other business rules:

- Its schema defines where each piece of information resides on the physical Fact maintenance object.
- It may define a lifecycle for all fact instances of this type to follow. Each fact must exist in a valid state as per its business object's lifecycle definition.
- It may define validation and other business rules to control the behavior of facts of this type.

FASTPATH: For more information about business objects, refer to [The Big Picture of Business Objects](#).

Fact Supports A Log

The Fact maintenance object supports a log. Any significant event related to a Fact may be recorded on its log. The system automatically records a log record when the fact is created and when it transitions into a new state. In addition, any custom process or manual user activity can add log entries.

FASTPATH:

Refer to [State Transitions Are Audited](#) for more information on logging. For more information about the various configuration tools available, refer to [Configuration Tools](#). For more information about user interfaces, refer to [Portal and Zone Features](#).

Real-Time Scheduler Administrative User Guide

General Options

Defining Installation Options

This section provides additional setup information related to your installation options record.

Refer to [Framework installation options](#) section for more information.

Base Time Zone

To support businesses spanning across multiple time zones, the system stores all date and time information in a single common time zone known as the base time zone or the server time zone. Furthermore, date and time information is stored in standard time (i.e. independent of any seasonal time shifting adjustments, if applicable, in that time zone).

The system also allows data to be entered and displayed in a different time zone in legal time (i.e. adjusted for seasonal time shifting, managing the conversion back and forth between the data entry and the storage time zones).

Entities associated with a geographic location such as activities, crew shifts and meetings, are entered and displayed in the specific time zone where they occur, the entity time zone. The rest of the application uses the user's preferred time zone to display date and time information.

The CDI portal displays data according to the user time zone; however, users have the ability to toggle time zones as needed.

Crews using the mobile application work their schedule using their local device time zone. The system handles the conversion of the data to and from the server time zone.

NOTE: The server time zone must be correctly specified on the installation options record for the system to work properly.

Formatted Address

Throughout the system address information is formatted for display based on standard address constituents.

NOTE: Logic to format an address resides in an "Address Information" algorithm on the installation options. Click [here](#) to see the algorithm types available for this system event. The system requires such algorithm to be plugged in.

Address Geocoding

The scheduler has to be provided with the geocode of any location that may be included on crew's route. These include locations of tasks, service centers, depots, mobile worker home addresses if they start or end their shifts from home, etc.

Unless a geocode is provided, algorithms on the various business entities that capture an address attempt to geocode the address when the entity is added or when the address changes.

The minimum match quality level for address geocoding is configurable. By default, the base system requires an exact match, which is the maximum quality level. You can control the quality level of geocoding for any entity that has an address by overriding the match quality parameter of the algorithms on each entity with a different minimum quality level, based on your business needs.

NOTE: Logic to geocode an address resides in an "Address Geocoding" algorithm on the installations options. Click [here](#) to see the algorithm types available for this system event.

Defining Locations

This portal is used to define your company's facility locations such as service centers, business offices, depots, etc. These may be used as locations where crews may start and end their shift, attend meetings, etc.

You can access the portal from the **Admin Menu > Resource Management > Location**.

A location's physical address must be geocoded so the scheduler can direct crews to and from that location. Unless explicitly provided, the system attempts to geocode the address using the geocoding service defined on the installation options, expecting an exact match.

NOTE: Logic to geocode the home address resides in an algorithm associated with the location's business object. Review this algorithms and adjust its match quality parameter as needed.

The following zones may appear as part of the portal's **Main** tab page

- **Location List.** This zone lists all locations. Broadcast a record to display the details of the selected record.
- **Location.** This zone provides information about the selected location.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_LOCATION](#).

Configuring Mapping Options

The CDI map as well as other zones showing items on a geographic map require that you set up options on a feature configuration record.

Find the [Feature Configuration](#) record for the **Map Server Configuration** feature type and define the various options as needed. (It may need to be defined if it does not exist).

Defining Geographic Areas

A geographic areas defines the boundary of an area in your territory.

Using a geographic area, you can represent regions such as the entire company territory, and certain dense population areas to be displayed as custom layers in the CDI map similar to alert areas. Refer to [CDI Map](#) for more information.

You can also define speed profile areas that allows, the scheduler to calculate more accurate travel times by adjusting the average travel speeds within a specific area. Refer to [Understanding Speed Profiles](#) for more information.

You can access the Location portal from the **Admin Menu > Service Management > Geographic Area**.

The following zones may appear as part of the portal's **Main** tab page

- **Geographic AreaList.** This zone lists all geographic areas. Broadcast a record to display the details of the selected record.
- **Geographic Area.** This zone provides information about the selected area.
- **Overrides.** This zone lists speed overrides for specific dates that apply to this area. The zone is only visible for speed profile areas.

Use the **Geographic Area Map** zone on the portal's **Map** tab page to define the map boundaries of the area.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_GEOGRAPHIC_AREA](#).

Setting Up Global Configuration

Many aspects of the application relies on setting up the **Global Configuration** master configuration record.

Refer to [Framework master configuration portal](#) for more information.

Service Areas

Understanding Service Areas

Typically, not all crews are allocated to do work in the whole territory and not all dispatchers monitor the whole territory. Service areas represent a division of your company's service territory into more manageable smaller areas. These areas may represent geographic regions, business units, or any other breakdown that is meaningful for your organization.

Examples of where service areas are used may be as follows:

- An activity is associated with a single service area that covers its location and the type of work requested.
- Crews on specific shifts may be allowed to work in specific service areas.
- Dispatchers may be responsible for monitoring specific service areas.
- A particular scheduler may cover one or more service areas.

This section describes concepts related to service areas.

Different Service Areas For Different Service Types

If multiple types of services are provided and the service territory is divided differently for each type of service, each service area must be associated with a corresponding service type. In the same way activity types need to be associated with the corresponding service type. For example, your company's electric service areas and activities differ from your gas service areas and activities. Service areas of the same service type cannot overlap geographically.

This information is used by the system when it [determines the service area for an activity](#). The system finds the service area that geographically includes the activity location and that belongs to the same service type associated with the activity's type.

Geographic Boundaries

The actual geographic boundaries of a service area may be captured as part of the service area record. While optional, there are benefits in capturing the service area's boundary when possible.

When this information is available, the system [determines the service area for an activity](#) by looking for a service area that its geographical boundaries includes the geocode of the activity's location.

Another benefit for recording the boundaries of service areas is that they can be displayed on geographic maps. The CDI map shows service areas as additional layer information available for dispatchers. The service areas covered by a scheduler's area are shown on the scheduler area portal.

If service areas follow postal code boundaries, their boundaries can be composed within the application under the following conditions:

- Your organization has a license to use Oracle Spatial.
- Postal code boundary data obtained from an external source is uploaded to a designated table.

Refer to [Prepare Postal Code Boundary Data](#) for more information.

Alternatively, geographic boundaries for a service area may be drawn in an external Geographic Information System (GIS) tool and the resulting ordinates (comma delimited latitude and longitudes) can then be copied and uploaded to the system.

Service Area Hierarchy

Service areas may be organized in tree-like hierarchy levels representing broader territories, regions, offices, etc. For example, your service territory may be organized in "Division", "District", and "County" levels where each service area belongs to a "County".

Setting up a service area hierarchy is optional and mainly used for reporting purposes and advanced searching in the CDI Gantt and other key online queries. When hierarchy information is available, these queries allow the user to also filter by levels higher than the service area.

The meaning of each hierarchy level is entirely dependent upon your business practices. You can define up to 3 levels with level 1 being the highest, level 2 below level 1, and level 3 below level 2. Refer to [Define Service Area Hierarchy Levels](#) for more information.

Using the "Service Area Hierarchy" portal to set up the entire hierarchy structure via manual entry or by uploading a spreadsheet.

You must define the entire service area hierarchy before you can associate service areas with it.

NOTE: Service areas must all belong to the lowest level defined in the hierarchy. For example, if the hierarchy uses 2 levels only, all service areas must reference a level 2 node in the hierarchical structure.

Configuring Service Areas

Design the boundaries of your service areas as well as the hierarchical structure above them. When ready, follow the steps in the order they are described in this section to configure the service areas.

Defining Service Area Hierarchy Levels

Enable each applicable service area hierarchy level by populating its corresponding override description on the **MI_SER_ ARE_HIR_LEV_FLG** lookup to describe its role in your business structure.

Defining Service Area Hierarchy

This portal is used to define the various nodes in a service area hierarchy.

Refer to [Service Area Hierarchy](#) for more information.

You can access this portal from the **Admin Menu > Service Management > Service Area Hierarchy**.

The following zones may appear as part of the portal's **Main** tab page

- **Service AreaHierarchy List.** This zone lists all service area hierarchy nodes. Broadcast a record to display the details of the selected record. You may also use this zone to download and upload hierarchies to and from an Excel spreadsheet.
- **Hierarchy Node.** This zone provides information about the selected service area hierarchy node.
- **Hierarchy Tree.** This zone provides a tree view of the entire service area hierarchy.
- **Service Areas In Hierarchy.** This zone lists service areas associated with the hierarchy.
- **Service Areas Not In Hierarchy.** This zone lists service areas yet to be associated with the hierarchy.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [MI_HIERARCHY](#).

Prepare Postal Code Boundary Data

If service areas follow postal code boundaries, the following steps must be done:

- Find the [Feature Configuration](#) record for the **Map Server Configuration** feature type. (It may need to be defined if it does not exist). Choose the option type **Oracle Spatial Licensed** and define a value of Y.
- Postal code boundary data has been obtained from an external source and uploaded to the **M1_POSTAL_CODE_BOUNDARY** table.

Refer to [Geographic Boundaries](#) for more information.

Defining Service Areas

This portal is used to maintain service areas which define boundaries of areas in your territory.

Refer to [Understanding Service Areas](#) for more information.

You can access this portal from the **Admin Menu > Service Management > Service Area**.

The following zones may appear as part of the portal's **Main** tab page

- **Service AreaList.** This zone lists all service areas. Broadcast a record to display the details of the selected record.
- **Service Area.** This zone provides information about the selected service area.
- **Covering Schedulers.** This zone lists schedulers covering the selected service area.

The **Service Territory** zone on the portal's **Map** tab page is used to upload and view the geographic boundaries of the service area as well as view all other service areas in the territory for the same service type, if any. Use the **Edit Text** button and enter in the "Polygon Definition" the boundary definition of the area. This may be the resulting ordinates from drawing the geographic boundaries in an external Geographic Information System (GIS) tool, or if applicable, the list of postal codes defining this service area.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_SVC_AREA](#).

Postal Code to Service Area Mappings

If service areas follow postal code boundaries but you cannot capture the geographic boundaries of your service areas for any reason, the base product's **Determine Service Area** algorithm uses the **Postal Code to Service Area Mapping (M1-ServiceAreaLookup)** extendable lookup to map a postal code to a service area.

For easier setup, you may use the **Upload Postal Codes for Service Territory (M1-UpPostCd)** BPA script to upload these mapping records from an Excel spreadsheet.

If multiple types of services are provided and the service territory is divided differently for each type of service, you will need to use a separate extendable lookup for each service type as follows:

- Subclass the base extendable lookup for each service type supported.
- Specify the service type as a business object option.
- Use a separate Excel spreadsheet for each service type.
- When using the upload BPA script, select the appropriate extendable lookup for the service type mapping you are currently uploading.

If boundary data is not available and your implementation relies on the base product logic to [determine the service area for an activity](#) based on postal code, set up mapping between each postal code and the service area that covers it. Refer to [Postal Code to Service Area Mappings](#) for more information.

NOTE: If previously your organization has set up the extendable lookup and now you have the ability to capture service area boundaries, you may use the **Convert Postal Codes to Service Area Boundary (M1-CnvPosCd)** BPA script to convert your existing mapping data into service area boundaries. You no longer need to maintain the extendable lookup mapping.

Capabilities

Understanding Capabilities

A capability is a skill, equipment, or other attribute of a crew which makes the crew qualified to perform work.

Refer to [Resource Capabilities](#) for more information.

This section describes how capabilities are used to achieve compatibility between work requirements and resources.

Worker Capabilities

This section describes key concepts related to worker capacities and how they are used.

Skill

Skills define an ability or talent needed to perform certain types of work and as such, may support different level of expertise for the same skill set.

Skill Levels

A skill set defines relationships between skills where each represents a specific level of expertise.

In the following example, the listed skills represent levels of expertise within a skill set. They are grouped together by referencing the same set number:

- Level 1 - Novice Inspector
- Level 2 - Standard Inspector
- Level 3 - Expert Inspector

Skill levels are used to make sure that skilled resources are being used as efficiently as possible. The scheduler can assign a crew to an activity based on their level of skill so as not to waste more experienced resources on a simple task, or to be sure to have a worker with a high level of skill where needed.

In this example, if an activity requires a "Novice Inspector" skill, the scheduler would first look for novice inspector crews before considering standard or expert inspectors.

Skill levels are optional. You don't need to specify level information if the skill does not require levels.

Skill preference can also be requested at the skill level. For example, an activity may require a novice level electrician, but would prefer an intermediate level electrician.

Skill Efficiency

A skill level can also specify an efficiency, or numerical value which quantifies the relative speed or competency of the skill level. For example a worker at "expert" skill level probably works faster than a worker at "novice" level. The higher

the number, the more efficient the worker. The skill efficiency value affects how the scheduler determines the scheduled duration for activities where skill requirements are specified.

The activity specifies the estimated amount of time it should take the work to be completed by a resource possessing the required skills. This estimated work duration represents the amount of time allotted for the activity on the schedule.

For an activity that requires a single minimum skill level and a skill efficiency is associated with each level, the scheduler adjusts the scheduled duration based on the assigned resource's skill level relative to the level that is required by the activity. As a result, if a more efficient resource is assigned, work may be completed faster. The percentage of the scheduled duration that is impacted by skill efficiency might also be specified on the activity type.

For example, an activity specifies an estimated duration of 60 minutes and at minimum requires an electrician of intermediate skill level. Assume that 30 minutes of the duration time is required for set up and clean up, and 30 minutes is required for electrical work if performed by a worker with intermediate electrical skills. This means that the efficiency duration impact percentage defined on the activity type would be set at 50% (only half of the 60 minutes requires skilled work).

With the skill efficiency set at 2 on the expert level skill, it is assumed that the expert only needs 15 minutes for the electrical work portion of the activity. Depending on the worker, the activity's scheduled duration would be calculated as follows:

- Intermediate worker = 60 minutes scheduled duration
- Expert worker = 45 minutes scheduled duration

If this value is not populated, the value is assumed to be 100% meaning that the skill efficiency impacts the entire scheduled duration.

NOTE: Skill Efficiency only applies to skills representing skill levels. Also, this functionality is ignored when the activity references more than one required skill as well as when multi person crews are used. This is not supported for complex activities.

Attribute

An attribute defines a condition that is true about the mobile worker or an item they need to possess to perform certain types of work. For example, having a security clearance, building access, drug testing, permits, language proficiency, etc.

Vehicle Equipment

Equipment is any device or apparatus used to perform certain types of work.

NOTE: This is a generic reference to a type of item and not a reference to a specific item.

Limited Time

Some capabilities may be limited in time. For example, a skill that requires a license that must be renewed every year or an attribute representing a security clearance that is effective for a limited period of time. In the same way a vehicle may be associated with a certain type of equipment for a specific period of time only.

When you set up a mobile worker or a vehicle with such capability, you would be asked to provide the applicable effective period.

Defining Worker Capabilities

This portal is used to define worker skills and attributes.

Refer to [Worker Capabilities](#) for more information.

You can access the Location portal from the **Admin Menu > Resource Management > Worker Capability**.

The following zones may appear as part of the portal's **Main** tab page

- **Skill List.** This zone lists all skills. Broadcast a record to display the details of the selected skill.
- **Attribute List.** This zone lists all attributes. Broadcast a record to display the details of the selected attribute.
- **Worker Capability.** This zone provides information about the selected capability.

Adding a new skill level to an existing skill set is achieved by duplicating an existing skill level of the skill set.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [MI_CAP_TYPE](#).

Defining Equipment

This portal is used to define equipment needed to perform certain types of work.

Refer to [Vehicle Equipment](#) for more information.

You can access the Location portal from the **Admin Menu > Resource Management > Equipment**.

The following zones may appear as part of the portal's **Main** tab page

- **Equipment List.** This zone lists all equipment. Broadcast a record to display the details of the selected equipment.
- **Equipment.** This zone provides information about the selected equipment.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [MI_CAP_TYPE](#).

Tasks

Understanding Activity Types

This section describes various features controlled by activity type configuration.

Refer to [About Activities](#) for more information.

Activity Business Object Controls Everything

Different types of activities involve different details and rules and as such, they are implemented using different activity business objects. It is important that you reference the appropriate activity business object when defining an activity type.

The base product provides various activity business objects to handle common types of work. Your implementation may introduce more custom business objects to adequately represent the data and business rules associated with a particular type of work. Refer to [Designing Activity Types](#) for more information.

The Activity's Host System

The activity type record describes work that may be requested from a specific host system. As such, it is required that the activity type references the external system record set up for the specific host system and thus the same activity type cannot be used for multiple host systems.

The corresponding identifier of the activity type in the host system may optionally be specified on the activity type. This may be used to default a new activity's type from its host's activity type when applicable.

All outbound messages to the host system related to activities of this type are directed to the external system defined for the host system.

Refer to [Host External System Configuration](#) for more information.

Service Type

If your company provides multiple service types and divides your service territory into different service areas for the different service types, you should state the service type associated with each activity type.

Refer to [Different Service Areas For Different Service Types](#) for more information.

Default Resource Capabilities

The activity type may define worker and vehicle capabilities needed to perform activities of this type. This information may be extended or overridden on the activity.

Refer to [Resource Capabilities](#) for more information.

When Can Work Be Performed

This section describes options that may be used to control when activities of this type can be scheduled.

Default Due Date

When the host system sends in an activity without a due date, the system can default it if such option is configured on the activity type. The activity type may state a number of days or hours within which the activity should be completed. Days and hours may refer to calendar or business days and hours, respectively.

You must specify a [work profile](#) if choosing to default due date by business days or business hours. This is because the system relies on the work profile definition of business days and hours for this type of work.

For example,

- A work profile is created specifying that work can be performed between 9 AM and 4 PM Monday through Thursday and 9 AM and 1 PM on Friday.
- An activity type references this work profile and also indicates that work must be performed within 24 business hours.
- If an activity of this type comes in from the host with a time window that starts on a Thursday at 3 PM with no end time window, the default due date and time is set to the following Wednesday at 2 PM (1 hour on Thursday, 4 hours on Friday, 7 hours on Monday and Tuesday, and 5 hours on Wednesday).

On the work profile you may also designate a "Minimum Business Day" value to establish the minimum duration that can be considered as a business day when defaulting an activity's due date by business days. This ensures that only time windows that are long enough are considered.

For example, if this is set at 5 hours and the activity must be completed within 3 business days on the activity type, but one of the days is a 4 hour day, that day will not be considered a full business day for the purpose of defaulting the activity due date.

Work Calendar

An activity type may reference a work calendar that defines public holidays during which work of this type should not be scheduled.

Work Profile

Certain types of work may only be performed on specific days and business hours. For example, cutting a meter for non-payment is not allowed on weekends or hydrant flushes can only be performed during the graveyard shift.

A work profile describes the business hours during which work is allowed to be performed and may be referenced on one or more activity types that follow this profile.

Typically work profiles represent a regulatory restriction that does not allow an activity to be scheduled outside of these windows. You should configure work profiles as appropriate to typical work patterns and to restrict activity types according to regulatory requirements or other requirements. With a work profile properly defined for the activity type, the system ensures that the activity is not scheduled outside of the established time window.

The system uses this information when it generates arrival time windows (ATW) for an activity.

Refer to [When Can The Activity Be Actually Scheduled?](#) for more information.

You can also define a "Minimum Late Time" to establish the maximum duration, in minutes, that activities associated with this profile can be scheduled late. Setting this to zero indicates that no lateness is allowed. If this value is not specified, lateness is not restricted.

NOTE: Changes to work profiles after they have been established and added to activity types do not impact existing activities, only activities created after the change.

Default Appointment Booking Group

When the system receives an appointment booking request from the host, it uses the appointment booking group specified on the request to identify the actual time windows during which appointments are allowed to be scheduled. If no appointment booking group is supplied by the host, the system uses the activity type's default appointment booking group.

You may specify an appointment group as well as a [work profile](#). In this situation, the scheduler would offer appointment times available considering both the work profile and the appointment booking group. For example, if the work profile specifies that work can only be performed Friday mornings, appointments for Friday afternoon will not be offered.

Refer to [Appointment Booking](#) for more information about the web service.

Determine Activity Priority

Typically the host system does not set the activity's scheduling priority but rather lets the system determine it based on activity type configuration as well as an optional activity business priority number.

For example, collection activities can be prioritized highest overdue amount first. The host system would include a business priority number equal to the amount overdue. Given this, the activity type logic would derive the scheduling priority based on stepped ranges of amounts overdue:

- Up to \$100 set scheduling priority to 1
- Up to \$1000 set scheduling priority to 5
- etc.

Another example may be for the host to provide a business priority number equal to the number of customers impacted so that the activity that impacts the most customers is prioritized higher.

Using stepped ranges allows you to set same type of work activities with different scheduling priorities based on business needs. If a scheduling priority cannot be determined based on a business priority number, the default priority specified on the activity type is used.

If your organization needs to maintain the concept of priority based on activity type but allow activities of same type to prioritize differently, it is important to keep scheduling priority ranges allocated to specific types of work and let the stepped list be a further list of ranges within the range designated for the activity type. This way the scheduling priority will

represent the preference between one type of work to other types as well as the preference within activities of one type of work.

Taking a Break While On Site

By default, breaks are planned to be taken in between activities. However, some activities may require the crew to remain on site for a long time. In this situation the crew may be entitled to a planned break on site. The activity type indicates whether breaks may be scheduled to be taken while the crew is on site or not.

If the activity type is set up to allow breaks to be planned while on site, the scheduler plans such breaks accordingly. The crew is provided with a visual cue on the mobile application's task list as to whether an activity is scheduled with some break time while on site.

Complex activity types always allow breaks to be scheduled while on site.

Advance Allocation

Certain types of work might need to be locked to their currently assigned crew at some time before the planned start time of the shift. When allocation is done in advance, dispatchers and planners can know in advance which crew is slated to perform the work. This information can be useful for planning purposes as per your business needs.

Activity Time Tracking

The system tracks important events such as status updates on the assignment log. This information can be used to summarize how much time was spent by the crew on travel, actual work, taking breaks on site, and whether there was any wait time on site.

Refer to [Crew Time](#) for more information.

You can configure the activity type to summarize that information and present it to the crew at activity completion time so they can review and adjust their time as needed.

NOTE: This feature is independent from timesheet functionality; however, if the crew has made adjustments to reported times on an activity, they are included in the timesheet total adjusted times summary.

This feature is automatically enabled for complex activity types as they tend to be long tasks during which a crew may have taken breaks on site and it is very important to keep track of an accurately reported work time to be subtracted from the overall complex activity.

Handling Expired Activities

If an activity has not been worked by its due date, the scheduler stops attempting to schedule it.

Depending on your configuration and business practices, activities of certain types may be configured to handle the situation as follows:

- Enable automatic extensions so that the scheduler will continue to try and schedule expired activities beyond the original effective time window by a set number of days or hours.
- Cancel the activity.
- Create a To Do for a user to review and make the necessary changes.

NOTE: Logic to handle the business rules associated with expired activities resides in an "Activity Expiration" algorithm on the activity's business object. Click [here](#) to see the algorithm types available for this system event.

Assist Activity Types

The activity type indicates whether this type of activity may be created for the purpose of assisting another crew with their work.

Refer to [Assist Activities](#) for more information.

It is recommended that your organization set up assist activities to require manual acknowledgment by the assisting crew. This serves as notification that they are aware of the fact that they have been assigned to assist another crew.

Creating Activities By The Crew

As per your business practices, crews may or may not be allowed to create activities in the field.

Activity type configuration supports the following options:

- The crew may be allowed or not allowed to create an activity of this type. For example, you may set up connects or disconnects such that they can be worked when dispatched as part of the shift schedule but never created in the field.
- The crew may be allowed or not allowed to create related activities while working an activity of this type. A related activity is an activity worked at the same location. If they are allowed to do so, you may optionally restrict the list of activity types they may use to create related activities.

Contracting Work

By default, work can only be performed by company resources. However, certain types of work can be allowed to be performed by contractors.

The activity type indicates whether activities are allowed to be performed by company crews only, any resource (including contractors), or restricted to shift-based crews (which can be composed of internal resources and/or contractors). This information can be overridden on the activity.

Refer to [About Contractors](#) for more information.

Understanding Service Classes

A service class is a broad categorization of activity types, allowing an activity type to belong to more than one service class. Each activity type must belong to at least one service class.

Examples of service classes include an electricity service class for all activities associated with electricity service, a gas service class for all activities associated with gas service, and a collections service class for all activities associated with collection, non-payment, and cutting off services.

Some of the main usages of service classes are as follows:

- Crews may not be allowed to perform certain classes of work on specific shifts. Activities of the types included in these service classes would not be assigned by the scheduler to these shifts.
- The primary function of a crew on a specific shift is defined using a service class. The scheduler is encouraged to prefer activities of the types included in the service class over others when assigning work to the crew on the shift.
- Crew shifts may be configured to reserve certain percentage of their capacity to activities of certain types using a service class.
- Dispatchers may monitor certain classes of work.

The scheduler uses this classification when scheduling work to shifts, based on their primary function and the type of work they are allowed to perform. Dispatchers also use this classification when they choose to monitor work by their service class association.

Review your activity types and group them to various service classes to meet your business requirements.

Designing Activity Types

The product can be used by many different types of businesses and can support a diverse set of activities ranging from field service orders, to installations and inspections, and even home deliveries.

The underlying processes of scheduling, routing, and data transfer are quite similar; the primary differences are in the details, such as the activity information received from the host, the necessary completion information to be captured and sent back to the host, and how that data should be validated and processed.

The base package provides a standard activity business object that serves as the foundation for all other activity specific business objects. This business object provides the basic logic to manage the activity lifecycle, handle scheduling and routing, and transfer data between the host and the mobile devices.

On top of this strong foundation, each implementation can build custom business objects by extending what is in the base system rather than creating them from scratch. The customization process should begin with an analysis of what already exists in the base system, followed by a thoughtful design phase that identifies exactly what additional information and processes are needed. Only when the design phase is complete should an implementer begin using system configuration tools to create or customize system entities.

This section discusses topics related to customizing and extending the base activity functionality.

The Lifecycle Business Objects

The base system provides lifecycle activity and assignment business objects that handle core functionality common to all types of activities. Additional subclass business objects are provided to support specific types of activities.

Activity and assignment business objects are unique in that they typically require some amount of customization to adequately represent the data and business rules associated with a particular type of work.

As a rule, custom activity and assignment business objects should always be a subclass of the corresponding base lifecycle business objects so that core functionality remains upgradable and consistent across of types of activities.

NOTE: Never clone the lifecycle activity or assignment business objects.

Activity And Assignment Business Objects Are Related

As part of the dispatch process, activity rules spawn an assignment record for the assigned resource using the **Assignment Business Object** option defined on the activity business object.

The activity business object defines the details received from the host system for an activity. An assignment is a copy of the activity record made available to the assigned resource, and where they also enter completion information. As such, the assignment structure includes a copy of key activity details as well as completion details.

When the assignment is created, key details are copied from the activity onto the assignment record. Details to copy are those specified in the data area referenced by the **Dispatched Details** option defined on the assignment business object.

When these details are changed on the activity, an audit rule on the activity business object makes the same changes onto the current assignment for the activity so as to keep it up to date with latest activity details. In the same way, an audit rule on the assignment business object sends an updated copy of the assignment to the crew's mobile device (or to the assigned contractor company) so as to keep their copy of the assignment up to date.

Extending Existing Business Objects

You should not need to create a new custom business object to achieve the following:

- Enforce additional custom rules. Your implementation can plug-in additional custom algorithms to existing business objects.
- Inactivate base algorithms. Though should be exercised with caution, your implementation can turn off the execution of a base algorithm using designated BO options.
- Use an alternate User Interface for the business object. Your implementation can override the base user interface used to display and maintain a business object, both on the server and on the mobile applications. Refer to the corresponding configuration techniques supported by the server and mobile applications.

If your implementation needs to perform more customization than that, subclass the corresponding lifecycle business object to introduce a new custom business object. Refer to [Custom Activity Business Objects](#) for more information.

Custom Activity Business Objects

Below is a high-level summary of the steps required to create custom activity and assignment business objects to support the various types of work performed by your business.

Analyze The Types Of Work

Identify all custom types of activities required for your implementation.

You will need to create a different activity business object for each type of work that requires different details to be sent from the host, completion details to be sent to the host, or validation logic. For example, one type of work might require that payment history is available to the crew as part of the activity details, while another might send a prior service record. Similarly, one type of work might require that the crew captures a particular kind of measurement or device reading, while another might capture customer satisfaction ratings; both types would require custom logic to validate the captured data.

For each activity type you identified above, do the following:

- Identify all data to be received from and sent back to the host.
- Design the UI maps to be used to display and capture data. You will need to design screens for the server application and, if MDTs are used in the fields, the mobile application as well.
- Identify all rules needed to extend the base business object for this type of activity, including algorithms that will be needed to handle validation of any data to be captured, monitor processes, etc. Make a list of these items so you can be sure to address all of them in the development process.
- Identify the data that needs to be synchronized to the crew. Do not include any data the crew does not need to know in order to perform their work.

Configure The Custom Business Objects

The following describes the high-level steps needed to configure custom activity business objects that handle the necessary data and rules to support such activities:

- Set up the custom activity business object as follows:
- If applicable, define the details that may be received from the host system in a custom data area.
- The details should not include completion details.
- Make sure to leverage [UI Hints](#) to automatically render these details as part of the activity's user interface on the server application.

- Use this data area to extend the base data areas that describe activity details that may be sent in from the host system and dispatched to the crew. If the lifecycle business object is M1 owned, extend the "Extended Activity Data Details (M1-DataDetails)" data area. If M2 owned, extend the "Extended Utility Data Details (M2-DataDetails)" data area.
- Create a custom activity business object as a child of the base activity lifecycle business object; thus, all events associated with the parent business object are automatically supported for the custom business object.
- The custom business object's schema must include the base activity business object schema as well as the custom activity details data area, if any.
- Finally, associate with the custom business object any custom algorithms and service scripts required to display or capture the additional activity data, validate it, or handle any other custom business processes.
- Set up the custom assignment business object as follows:
- If applicable, define in a custom data area the completion details relevant to this type of activity to be captured by the crew and sent to the host system.
- Make sure to leverage [UI Hints](#) to automatically render these details as part of the assignment's user interface on the server application.
- Use this data area to extend the base data areas that describe activity completion details that may be sent to the host system. If the lifecycle business object is M1 owned, extend the "Extended Utility Completion Details (M1-CompletionDetails)" data area. If M2 owned, extend the "Extended Utility Completion Details (M2-CompletionDetails)" data area.
- Create a custom assignment business object as a child of the base assignment lifecycle business object; thus, all events associated with the parent business object are automatically supported for the custom business object.
- The custom business object's schema must include the base assignment business object schema as well as the custom activity completion details data area, if any.
- Finally, associate with the custom business object any custom algorithms and service scripts required to display or capture the additional activity completion data, validate it, or handle any other custom business processes.
- Reference the custom assignment business object as the **Assignment Business Object** option on the custom activity business object.
- Define an activity type record referencing your custom activity business object; it already knows the assignment business object to use via the business object option. Refer to [Defining Task Types](#) for more information.
- Add your activity type to an appropriate service class.
- If your implementation uses the mobile application to work such activities, extend it as follows:
- Add business rules to the mobile application to validate and process the completion data as needed.
- Design and implement the mobile user interface sections. This includes the sections used to display the additional activity information sent from the host system as well as those used by the crew to capture the additional completion information.
- Add the new activity type, the custom business objects and all the necessary items needed to support them to your designated custom deployment parts.
- Create and activate a new deployment.

NOTE: Refer to the mobile application training material for a more detailed instruction on how to extend and customize the mobile application.

Configuring Activity Options

Follow the steps described in this section to configure various activity related options.

Defining Site ID Configuration

When an activity does not reference an explicit site ID for site grouping purposes the system, by default, combines all site address elements to compose the site ID before it sends the activity to the scheduler.

Refer to [Site Identifier](#) for more information.

If your business requires a subset of the site address elements to be used, find the [Feature Configuration](#) record for the **Site Id Configuration** feature type (It may need to be defined if it does not exist), and add options to list the relevant fields to use instead.

NOTE: If your data is such that it does not allow you to use a subset of the site address elements, but rather you need to parse your address elements and use custom rules to compose a site ID, this feature configuration would not be useful. For this situation, it is recommended that you either correct the data in the host system when possible, or add custom parsing rules to compose the site ID in the host system and send it along with the activity. Alternatively, you can add custom activity post processing business object rules that set the site ID on the activity when it is added or when the address changes.

Host External System Configuration

Outbound messages to an activity's host system are directed to the external system defined for the host system on the activity type. The kind of messages supported by each host system depend on the type of integration implemented.

Refer to [Crew May Be Managed Externally](#) for learn more about the type of integrations supported.

Set up an [External System](#) record that handles all the outbound messages supported by the host system.

Refer to [Configuring Outbound Messages](#) for more information.

Defining Remark Types

This portal is used to maintain remark types. Standard remarks define common indications that can be entered when an activity is completed and are sent to the host system as part of the completion details. The activity type defines the types of remarks valid for this type of work.

You can access this portal from the **Admin Menu > Service Management > Remark Type**.

The following zones may appear as part of the portal's **Main** tab page

- **Remark TypeList.** This zone lists all remark types. Broadcast a record to display the details of the selected record.
- **Remark Type.** This zone provides information about the selected remark type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [MI_RMK_TYPE](#).

Defining Priority Profiles

This portal is used to maintain priority profiles. The scheduler applies a factor onto the activity's scheduling priority to calculate its priority over time. This factor is defined by a "Priority Profile" as a function over the activity's broad time window. The priority profile also defines whether activities associated with this profile are considered [mandatory or optional](#). An activity is associated with the priority profile referenced on its activity type.

You can access this portal from the **Admin Menu > Service Management > Priority Profile**.

The following zones may appear as part of the portal's **Main** tab page

- **Priority Profile List.** This zone lists all priority profiles. Broadcast a record to display the details of the selected record.
- **Priority Profile.** This zone provides information about the selected priority profile.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_PRI_PROF](#).

Defining Work Profiles

Work profiles define operational time windows and other time restrictions regarding when certain types of work can be performed. An activity is associated with a work profile referenced on its activity type.

Refer to [Work Profile](#) for more information.

This portal is used to maintain work profiles.

You can access this portal from the **Admin Menu > Service Management > Work Profile**.

The following zones may appear as part of the portal's **Main** tab page

- **Work Profile List.** This zone lists all work profiles. Broadcast a record to display the details of the selected record.
- **Work Profile.** This zone provides information about the selected work profile.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_APPT_BOOK_GRP](#).

Defining Appointment Booking Groups

Appointment booking groups define valid appointment time windows. Different types of work may be associated with different appointment times. .

Refer to [Appointment Booking Group](#) for more information.

This portal is used to maintain appointment booking groups.

You can access this portal from the **Admin Menu > Service Management > Appointment Booking Group**.

The following zones may appear as part of the portal's **Main** tab page

- **Appointment Booking Group List.** This zone lists all appointment booking groups. Broadcast a record to display the details of the selected record.
- **Appointment Booking Group.** This zone provides information about the selected appointment booking group.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_APPT_BOOK_GRP](#).

Uploading Common Attachments

Create common attachments for supporting documents needed to be available for the crew when they work specific activity types. Refer to [Attachments Related To The Type Of Work](#) for more information.

Refer to [Defining Task Types](#) to set up activity types.

Defining Activity Types

Set up an activity type for each type of work handled by your business.

Refer to [Defining Task Types](#) to set up activity types.

Defining Service Classes

This portal is used to maintain service classes.

Refer to [Designing Activity Types](#) for more information.

You can access this portal from the **Admin Menu > Service Management > Service Class**.

The following zones may appear as part of the portal's **Main** tab page

- **Service Class List.** This zone lists all service classes. Broadcast a record to display the details of the selected record.
- **Service Class.** This zone provides information about the selected service class.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_SVC_CLS](#).

Defining Task Types

Anything the crew needs to perform or spend time on is represented as task to be incorporated as part of the crew's shift schedule. Most tasks are activities the crew needs to perform but they can also be to attend meetings, replenish stock, handle goods at the depot, or take a planned break.

A task type defines default attributes for all tasks of a particular type.

The system supports the following classes of task types:

- **Activity Types.** You may define different activity types to describe the various types of work your business handles. Refer to [Understanding Activity Types](#) for more information.
- **Break Types.** You may define different types of breaks crews are entitled to take during their shift. Refer to [Breaks](#) for more information.
- **Non Productive Task (NPT) Types.** You may define different types of administrative tasks a crew may perform during their shift. Refer to [Non-Productive Tasks \(NPTs\)](#) for more information.
- **Depot Task Types.** You may define different depot task types for different depot records. Refer to [Depot Task](#) for more information.
- **Period of Unavailability (POU) Task Types.** You may define different types of meetings or events a crew may attend. Refer to [Meetings](#) for more information.

This portal is used to maintain all classes of task types.

You can access this portal from the **Admin Menu > Service Management > Task Type**.

The following zones may appear as part of the portal's **Main** tab page

- **Task TypeList.** This zone lists all task types. Broadcast a record to display the details of the selected record.
- **Task Type.** This zone provides information about the selected task type.
- **Service Classes.** This zone lists the service classes that include the selected task type. This zone is only visible for activity types.
- **Attachments.** This zone is used to maintain the list of common attachments associated with the selected task type. Refer to [Attachments Related To The Type Of Work](#) for more information. The zone is only visible for activity types.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_TASK_TYPE](#).

Crews

Understanding Crew Hierarchies

Crews may be organized in tree-like hierarchy levels representing the business units they are associated with. For example, your business practices may require crew hierarchy levels of "Business Unit", "Department", and "Office" levels where each crew belongs to an "Office".

Setting up a crew hierarchy is optional and mainly used for reporting purposes and advanced searching in the CDI Gantt and other key online queries. When hierarchy information is available, these queries allow the user to also filter by levels higher than crew.

The meaning of each hierarchy level is entirely dependent upon your business practices. You can define up to 3 levels with level 1 being the highest, level 2 below level 1, and level 3 below level 2. Refer to [Define Crew Hierarchy Levels](#) for more information.

Using the "Crew Hierarchy" portal to set up the entire hierarchy structure via manual entry or by uploading an Excel spreadsheet.

You must define the entire crew hierarchy before you can associate crews with it.

NOTE: Crews must all belong to the lowest level defined in the hierarchy. For example, if the hierarchy uses 2 levels only, all crews must reference a level 2 node in the hierarchical structure.

Configuring Crew Hierarchy

Design the crew hierarchical structure and when ready, follow the steps in the order they are described in this section to configure the hierarchy.

Defining Crew Hierarchy Levels

Enable each applicable crew hierarchy level by populating its corresponding override description on the `M1_CREW_HIR_LEV_FLG` lookup to describe its role in your business structure.

Refer to [Understanding Crew Hierarchies](#) for more information.

Defining Crew Hierarchy

This portal is used to define the various nodes in a crew hierarchy.

Refer to [Understanding Crew Hierarchies](#) for more information.

You can access this portal from the **Admin Menu > Resource Management > Crew Hierarchy**.

The following zones may appear as part of the portal's **Main** tab page

- **CrewHierarchy List.** This zone lists all crew hierarchy nodes. Broadcast a record to display the details of the selected record. You may also use this zone to download and upload hierarchies to and from an Excel spreadsheet.
- **Hierarchy Node.** This zone provides information about the selected crew hierarchy node.
- **Hierarchy Tree.** This zone provides a tree view of the entire crew hierarchy.
- **Crews In Hierarchy.** This zone lists crews associated with the hierarchy.

- **Crews Not In Hierarchy.** This zone lists crews yet to be associated with the hierarchy.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_HIERARCHY](#).

Configuring Crew Options

Follow the steps in the order they are described in this section to configure crew related options.

Defining Mobile Worker Types

A mobile worker type defines attributes of a mobile worker's job description. For example, a mobile worker type can define the assumed skills or attributes of all mobile workers of that type.

Refer to [Mobile Workers](#) for more information.

This portal is used to maintain mobile worker types.

You can access this portal from the **Admin Menu > Resource Management > Mobile Worker Type**.

The following zones may appear as part of the portal's **Main** tab page

- **Mobile Worker Type List.** This zone lists all mobile worker types. Broadcast a record to display the details of the selected record.
- **Mobile Worker Type.** This zone provides information about the selected mobile worker type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_RESRC_TYPE](#).

Defining Vehicle Types

A vehicle is associated with a vehicle type that defines attributes common to all vehicles of that type. For example, it can define the assumed equipment of all vehicles of that type.

Refer to [Vehicles](#) for more information.

This portal is used to maintain vehicle types.

You can access this portal from the **Admin Menu > Resource Management > Vehicle Type**.

The following zones may appear as part of the portal's **Main** tab page

- **Vehicle Type List.** This zone lists all mobile worker types. Broadcast a record to display the details of the selected record.
- **Vehicle Type.** This zone provides information about the selected mobile worker type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_RESRC_TYPE](#).

Defining Crew Types

A crew type record defines a particular type of crew, such as a one-person crew or a multi-person crew, etc.

Refer to [About Crews](#) for more information.

This portal is used to maintain crew types.

You can access this portal from the **Admin Menu > Resource Management > Crew Type**.

The following zones may appear as part of the portal's **Main** tab page

- **Crew Type List.** This zone lists all crew types. Broadcast a record to display the details of the selected record.
- **Crew Type.** This zone provides information about the selected crew type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_RESRC_TYPE](#).

Configuring Crew Shift Options

Follow the steps in the order they are described in this section to configure crew shift related options.

Defining Shift Types

A crew shift type is closer to the common meaning of the term 'shift' which typically refers to the time of day during which a crew performs work, such as first shift or second shift. A shift type defines attributes common to all crew shifts and shift templates of this type, such as their typical start and end times, duration, etc.

Refer to [About Crew Shifts](#) for more information.

This portal is used to maintain shift types.

You can access this portal from the **Admin Menu > Resource Management > Shift Type**.

The following zones may appear as part of the portal's **Main** tab page

- **Shift Type List.** This zone lists all shift types. Broadcast a record to display the details of the selected record.
- **Shift Type.** This zone provides information about the selected shift type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_CREW_SHFT_TYPE](#).

Defining Shift Cost Profiles

Global and shift-based costs are used by the scheduler to evaluate and compare possible schedule solutions. A less costly schedule is preferred over a more costly one. Shift cost profiles are predefined shift-based cost factors that are used as multipliers against global costs. You can assign a shift cost profile to one or more shift templates or to individual shifts.

For example, a shift-based overtime cost of 3 means that overtime for all crew shifts associated with this profile is three times more than the overtime cost defined globally.

NOTE: Before you define shift cost profiles, study your organization's cost requirements and how they relate to specific service areas and/or service classes. Small changes to shift-based factors may result in significant schedule changes.

The shift cost profile record may also include a list of relevant service areas and/or service classes. These associations do not restrict the use of the shift cost profile in any way; you can associate any shift cost profile to any shift. Rather, the system uses these associations to recommend appropriate shift cost profiles when you are defining crew shifts.

Refer to [About Crew Shifts](#) for more information.

This portal is used to maintain shift cost profiles.

You can access this portal from the **Admin Menu > Resource Management > Shift Cost Profile**.

The following zones may appear as part of the portal's **Main** tab page

- **Shift Cost Profile List.** This zone lists all cost profiles. Broadcast a record to display the details of the selected record.

- **Shift Cost Profile.** This zone provides information about the selected cost profile.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_SHFT_COPRF](#).

Defining Shift Related Task Types

Shift configuration may include planning of breaks and Non Productive Tasks (NPT).

Refer to [Defining Task Types](#) to set up Break and NPT types.

Periods of Unavailability (POUs)

Understanding POU Configuration

A period of unavailability (POU) is a specific period of time during which one or more resources are not available to perform work; for example, meetings or training sessions crews attend as part of their shift, a mobile worker's vacation leave, a vehicle's planned maintenance time, etc.

A POU type defines basic characteristics of all POUs of a particular type, for example:

- A POU type specifies whether or not POUs of that type can be recurring and if so, specifies the POU template business object used to create recurring POUs of that type.
- A POU type also identifies the class of resource (crew, mobile worker, vehicle, etc.) that can be associated with POUs of that type.

Refer to [About Periods of Unavailability \(POUs\)](#) for more information.

Defining Period of Unavailability Task Types

For crew meeting POUs, a POU task is automatically created by the system for each invited crew in order to incorporate the meeting as a task into their schedule.

The system uses the task type defined on the POU type to create the POU Task. If not specified, the default task type specified on the global master configuration record is used.

Refer to [Defining Task Types](#) to set up POU task types.

Defining Period of Unavailability Types

This portal is used to maintain POU types.

You can access this portal from the **Admin Menu > Resource Management > Period of UnavailabilityType**.

The following zones may appear as part of the portal's **Main** tab page

- **Period of UnavailabilityType List.** This zone lists all POU types. Broadcast a record to display the details of the selected record.
- **Period of UnavailabilityType.** This zone provides information about the selected POU type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_POU_TYPE](#).

Dispatchers

Understanding Dispatch Areas

When dispatchers log on to start their shift, they select dispatch areas they will monitor during that shift. A dispatch area includes one or more [service areas](#) as well as one or more [service classes](#), indicating the type of work the dispatcher monitors in the specified areas.

The system considers crews on specific shifts, contractor capacities, and activities to match the criteria defined by a dispatch area as follows:

- The crew shift covers at least one of the service areas defined by the dispatch area, and the crew is allowed to perform work associated with at least one of the service classes defined by the dispatch area.
- The activity's service area is included in the list of service areas defined by the dispatch area and its activity type is included in at least one of the service classes defined by the dispatch area.
- If your organization uses contracted work capacities, then similar to crew shifts, a contractor capacity has to cover at least one of the service areas defined by the dispatch area as well as cover at least one service class defined by the dispatch area. If the capacity is defined for a specific activity type only, that activity type has to belong to a service class covered by the dispatch area.

Refer to [Managing Crews And Activities](#) for more information.

Defining Dispatcher Types

A dispatcher type record defines a particular type of dispatcher.

Refer to [About Dispatchers](#) for more information.

This portal is used to maintain dispatcher types.

You can access this portal from the **Admin Menu > Resource Management > Dispatcher Type**.

The following zones may appear as part of the portal's **Main** tab page

- **Dispatcher Type List.** This zone lists all dispatcher types. Broadcast a record to display the details of the selected record.
- **Dispatcher Type.** This zone provides information about the selected dispatcher type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_RESRC_TYPE](#).

Defining Dispatch Areas

This portal is used to maintain dispatch areas. A dispatch area is a predefined set of service areas and service classes that can be monitored by one or more dispatchers.

Refer to [Understanding Dispatch Areas](#) for more information.

You can access this portal from the **Admin Menu > Service Management > Dispatch Area**.

The following zones may appear as part of the portal's **Main** tab page

- **Dispatch Area List.** This zone lists all dispatch areas. Broadcast a record to display the details of the selected record.

- **Dispatch Area.** This zone provides information about the selected dispatch area.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_DISP_AREA](#).

Customizing The CDI Portal

Being the primary point of user interaction for dispatchers, the CDI portal is highly customizable. This section describes various customization options your implementation can use.

Refer to [About CDI](#) for more information.

Gantt Customizations

The following section describes the various areas in which the scheduling Gantt may be customized to meet your business needs.

General Gantt Configuration

Like with all CDI portal zones, the crews listed in the Gantt's left pane are restricted to those in the [dispatcher's scope](#) of currently monitored crew shifts. If the dispatcher has applied a [filter](#) then the list is temporarily narrowed down based on the filtering criteria. For each crew, one or more of their shifts are displayed in the right pane along with their schedule of tasks.

The system automatically refreshes the information being displayed to ensure an up to date picture of the schedule is presented to the dispatcher.

NOTE: Logic to provide the Gantt with the list of crew shifts and their scheduled tasks, as well as all the necessary information and visual cues it needs in order to present them, resides in a plug-in. Click [here](#) to see the algorithm types available for this system event.

Similarly, when the Gantt view is toggled to present the tasks scheduled for a single shift (Task View), the system lists the shift's tasks in the left pane and their schedule and available time windows in a time line in the right pane.

NOTE: Logic to provide the Gantt with all the necessary information and visual cues it needs to present the schedule of a single shift in "Tasks View" resides in a plug-in. Click [here](#) to see the algorithm types available for this system event.

These algorithms and more are provided as part of a default configuration that controls various aspects of the Gantt functionality as well as its visual appearance.

You may override these settings as per your company's preferences; for example you can

- Provide custom icons and choose different colors instead of those used by default
- Choose more columns (from a predefined list) to appear by default on the "List of Crews" pane
- State whether your company prefers the Gantt's main timeline view to list individual shifts on the left pane instead of the default option to group them by crew
- and more

NOTE: The default configuration used by the CDI Gantt, as well as other Gantt zones, is defined in the **Gantt Configuration (M1-GanttConfiguration)** extendable lookup. Review and adjust these settings for all or some of the Gantt zones.

Frequent Data Refresh

Key operational zones are set up to frequently refresh their data to reflect the most recent updates.

Ensure the automatic refresh interval is properly set up for the following zones:

- KPI Summary
- Alert Queue dashboard zone
- Backup Alert queue dashboard zone

The refresh interval for these zones is defined on their maintenance portals. Individual users can override these settings in the zone refresh parameter in user preferences.

NOTE: It is recommended to set up the "Activity Search" zone to not automatically refresh. This zone is assumed to be used to find specific activities as needed.

Right-Click Context Menus

You can add custom menu options to the context menu that appears when a user right-clicks a task or shift in the [Scheduling Gantt](#) zone.

The system supports several customizable context menus:

- M1-GANTT-ACTIVITY. This menu is used for a single activity that is not completed.
- M1-GANTT-ACTIVITY-MULTI. This menu is used for multiple activities that are not completed.
- M1-GANTT-BREAK. This menu is used for a break task.
- M1-GANTT-DEPOT-TASK. This menu is used for a depot task.
- M1-GANTT-DEPOT-TASKMULTI. This menu is used for multiple depot tasks.
- M1-GANTT-POU. This menu is used for a POU task.
- M1-GANTT-SHIFT. This menu is used for a single shift in CDI scheduling Gantt.
- M1-GANTT-SHIFTSHIFT_PORTAL. This menu is used for a single shift on Crew Shift portal Gantt.
- M1-GANTT-TASK. This menu is used for any other single task.

NOTE: Menu lines added to these context menus must invoke a BPA script to perform the work, and the BPA script must follow a specific schema. Each context menu provided with the base package contains at least one item that serves as an example of how to implement such a menu item.

Gantt Filters

The **CDI Gantt Filters (M1-GanttFilters)** extendable lookup defines filters provided by the base product that your implementation may choose to customize.

Refer to [Filter The List of Crews](#) for more information.

To inactivate a base filter, set the corresponding record in the extendable lookup as inactive.

You can introduce additional custom filters as needed to meet your business requirements. Follow the steps below to set up a new Gantt filter:

- Create a data explorer query zone that returns a list of shift identifiers based on your business requirement. The zone must contain only the shift id as an output column.
- Create a data explorer (FWLZDEXP) Business Service for your zone. The Business Service must map the zone's returned Shift ID column to an element named 'shiftId'. If you wish to default any input parameters, you may do so by using the standard schema standards (e.g. default = "xyz").
- Add your business service to the CDI Gantt Filters extendable lookup.

NOTE: It is your responsibility to ensure that the SQL is efficient and references the dispatcher's data scope. The Gantt algorithm does not validate that the returned shifts are indeed relevant to the dispatcher. Refer to the base filters for examples.

Hover Text

The text displayed when hovering over a task on the scheduling Gantt may be different from the task's standard information throughout the system.

NOTE: Logic to format the detailed description displayed on the Gantt for a task resides in an "Task Detailed Information" algorithm on the task business object. Click [here](#) to see the algorithm types available for this system event.

In the same way, the text displayed when hovering over a shift on the scheduling Gantt may be different from the shift's standard information throughout the system.

NOTE: Logic to format the detailed description displayed on the Gantt for a shift resides in an "Shift Detailed Information" algorithm on the shift business object. Click [here](#) to see the algorithm types available for this system event.

Visual Cues

Many of the visual cues associated with shifts and tasks are controlled by predefined styles in the **Gantt Style (M1-GanttStyle)** extendable lookup.

Your implementation may configure the system to show a custom icon in the task bar for certain types of activities and POU. To do so, perform the following:

- Register the custom icon on the [Display Icon Reference](#) portal.
- Select the icon reference name on the activity type or POU task type.

Showing an icon in the task bar for activities is optional. Note that even when an icon is specified on the activity type, the dispatcher may opt to not display it as their preference. Refer to [Customize Your Gantt Display](#) for more information.

For POU tasks though, an icon is always shown in the task bar to distinguish these tasks from other classes of tasks. The base icon is shown by default if a specific icon is not defined on the POU task type.

Task Coloring Rules

By default, a task bar's background color is determined based on the class and state of the task.

Your implementation may introduce additional color rules for the dispatchers to use. For example, color tasks based on their activity type, how close the activity is to its due date, its service area, etc. Dispatchers can toggle between the color rules and set a color rule as their default.

Use the **Gantt Color Rule (M1-ColorRuleSet)** extendable lookup to define more color rules as needed.

CDI Map Customizations

The following section describes the various areas in which the CDI Map may be customized to meet your business needs.

CDI Map Layers

Map layers are defined by the "CDI Map Layer" extendible lookup. Each map layer definition indicates whether the layer should be loaded and displayed when the CDI Map is initially loaded. You can override the default indication as needed.

You can add to the extendable lookup custom layers of the following types:

- A custom layer may reference a MapViewer theme name you have predefined in the MapViewer repository. Note that this theme must not assume any context, i.e. it is called without any input from the CDI map.

- The system allows you to define geographic areas for various usages. Refer to [Defining Geographic Areas](#) for more information on how to set up these areas. Once defined, a custom layer may be added to show all active geographic areas of a specific usage. For example, if you have defined a geographic hazard area that crews should not enter, that area can be shown on the CDI map.

Hover Text

The system calls the service script configured on the map layer entry in the extendable lookup to format the information displayed when the user hovers over an icon in that layer. You can use a custom service script as needed, so long as it includes the base script schema as part of its schema to ensure upgradability.

The text displayed when hovering over a task or a shift on the map may be different from the task's standard information throughout the system.

NOTE: The logic used to display detailed information about a related shift or task resides in the same algorithms used to format the task and shift [Gantt hover text](#). The algorithms are designed to compose a different hover texts when called from the Gantt and another when called from the map.

Alerts

Understanding Alert Types

An alert type controls the business rules associated with raising and evaluating alerts for a certain situation meaningful to your business.

Refer to [About Alerts](#) for more information.

This section describes various features controlled by alert type configuration.

Alert Business Object Controls Everything

The base product provides various alert business objects to handle most types of alerts. It is important that you reference the appropriate alert business object when defining an alert type.

Below are some examples of alerts supported by the base product. Review their logic and define corresponding alert types for those relevant to your implementation.

- **Missed appointment** alerts are generated when an appointment has been missed.
- **Unacknowledged emergency** alerts are created when a crew does not acknowledge an emergency alert within a specified period of time. Refer to [Unacknowledged Emergency Alert](#) for more information.
- **Underscheduled Complex Activity** alerts are generated for complex activities that cannot be scheduled beyond a specified threshold.
- **Synchronization** alerts are created when the crew does not receive a dispatched activity's assignment from the server within a specified period of time. Refer to [Synchronization Alert](#) for more information.
- **Cancel Activity** alerts are created on an attempt, typically initiated by the host system, to cancel an activity that has been dispatched to the crew. Refer to [Activity Cancelation By Host](#) for more information.
- **Crew Idle** alerts are generated when the crew shift is idle for a specified period of time.
- **Crew Taking Too Long** alerts are generated when the crew has been on site for some predefined minutes since the estimated completion time of an activity.
- **Hazard Area** alerts are generated when crews enter a hazard area. Refer to [Hazard Areas](#) for more information.

- **Vehicle Leave Area** alerts are generated when vehicles leave a certain geographic area. Refer to [Vehicle Leave Areas](#) for more information.
- **Panic** alerts are created when a crew in distress presses the "Panic" button on their mobile device. Refer to [Request For Help](#) for more information.
- **Timed Event** alerts are created when a crew has created an event with an expiration date and time to indicate that the dispatcher should check on them if the event expires. Refer to [Request For Help](#) for more information.

Your implementation may extend or customize the base business objects or introduce custom business objects as needed. Refer to [Designing Alert Types](#) for more information.

Alert Creation

Most alerts are generated in the background. The system frequently checks for the condition associated with each active alert type and raises alerts for each situation that meets the criteria.

Alerts created by base product rules explicitly exclude offline crews from being checked for alerts related to their work progress. Refer to [Offline Mode](#) for more information.

NOTE: Logic to create alerts of a given alert type resides in an "Alert Creation Algorithm" defined on the alert type. Click [here](#) to see the algorithm types available for this system event. The **Alert Type Monitor** background process calls this algorithm for each active alert type. The batch process needs to run frequently to create alerts in a timely manner. Refer to [M1-ATMTR](#) batch control for more information.

A creation algorithm is not required if there is another mechanism that raises alerts of this type. Some types of alerts may be raised on demand by a user or by other business rules throughout the application. For example, a crew may raise a "Panic" alert from the field to request immediate help. This type of alert is created right way upon the crew's request. This type of alert does not need to reference any creation algorithm on the alert type record. Refer to [Request For Help](#) for more information.

Alert Creation Threshold

The creation algorithm is typically designed to use some sort of a threshold that is taken into account, amongst other criteria, when determining whether an alert of this type should be raised or not. The threshold is set on the alert type and its meaning is only known to the logic creating the alert. For example, a crew idle alert is raised if a crew is idle more than the amount of minutes specified in the creation threshold.

Valid Activity Types

For activity related alerts, you may specify a restricted list of activity types for which this type of alert would be raised. If no activity types are specified, alerts of this type are raised for any type of activity.

Alert Distribution

An alert reports a problem related to an activity, a crew on a specific shift, or a resource at a specific location.

The system attempts to assign the alert to a logged on dispatcher that is currently monitoring the activity or crew, or covers the location associated with the alert. If no dispatcher is currently monitoring the subject of the alert, the system further looks for a logged on dispatcher that is eligible to watch it based on their dispatcher record settings, but did not elect to monitor the area or service class on today's dispatcher shift.

As a last resort the system attempts to assign the alert to any logged on dispatcher, if this is allowed by configuration.

When multiple candidate dispatchers are available to receive the alert, the system balances the load by preferring the one with the least number of open alerts.

An email is sent to the email address specified on the alert type to notify a supervisor in case the alert could not be assigned to any dispatcher.

NOTE: Logic to assign an alert to a dispatcher resides in an algorithm on the alert's business object. These rules can be customized during implementation to meet your organization's business requirements.

Alert Priority

Some alerts may need to be addressed before others. You can specify a priority on the alert type to represent the priority of alerts of this type relative to other alerts. The priority dictates the order by which alerts are presented to dispatchers in their alert queue.

You can also specify on the alert type whether or not alerts of this type should make a sound when they appear in the alert queue to help get the attention of the dispatcher. The sound file to use as well as other parameters need to be populated on the [global configuration](#) record for alert sound to be enabled.

Alert Evaluation

The system automatically closes an alert if the underlying condition no longer exists or is no longer relevant. For example, an alert related to a crew being idle would automatically close when the crew is no longer idle or if the crew's shift is completed. Similarly, a missed appointment alert should be automatically closed if the activity is subsequently canceled.

NOTE: Logic to evaluate an existing alert and close it if it is no longer valid resides in an "Alert Evaluation Algorithm" defined on the alert type. Click [here](#) to see the algorithm types available for this system event. The **Alert Monitor** background process calls this algorithm for each open alert. The batch process must run frequently to properly evaluate and close existing alerts. Refer to [MI-ALMTR](#) batch control for more information.

However, alerts that were raised by a user on demand like a crew's panic alert, can only be manually closed by a dispatcher after they contact the crew and make sure they are no longer in danger.

NOTE: When applicable, it is important to associate alert types with an evaluation algorithm to ensure that alerts do not remain in any dispatcher's queue when the underlying issue has already been resolved.

Snoozing Options

The system supports a snooze feature, similar to the snooze on an alarm clock. You can configure alerts of a given type to snooze for a configurable amount of minutes after the dispatcher acknowledges the alert. When such alert is acknowledged, the alert is dropped from the dispatcher's alert queue while the dispatcher tries to address the issue, and reappears after that amount of time, if the conditions are still true for the alert.

As a general rule, you should allow an alert to snooze if it is important that the alert be closed only when the underlying issue has been resolved, and you want to allow the dispatcher for more time to resolve it. When the snooze time expires, the system evaluates the underlying condition and closes the alert if the condition no longer exists.

For some types of alerts, you may want to allow the dispatcher to adjust the snooze time themselves when they acknowledge the alert.

NOTE: If the alert type does not support snoozing, alerts of this type are automatically closed once a dispatcher acknowledges them irrespective of whether the underlying issue is resolved or not.

Can A Dispatcher Manually Close Alerts?

Typically alerts should be closed when the underlying condition is resolved and as such, should not allow a user to manually close it. However, some types of alerts may be designed to allow such user intervention.

NOTE: The logic that creates the alerts has to be designed to handle manual closure so that it does not recreate the alert again once closed. Refer to the description of the algorithm that creates the alerts to better understand if it handles manual closure of alerts.

Should The Alert Expire?

You can configure alerts of given type to automatically expire and close after some predefined amount of time, irrespective of whether the underlying condition is resolved or not.

Should The Alert Recycle?

Recycling an alert refers to the process of reassigning it to another dispatcher. You can configure alerts of given type to recycle if the assigned dispatcher doesn't acknowledge the alert within a configurable amount of time. If a recycle time is not specified on the alert type, alerts of this type are never recycled based on time; however, they will still be recycled when the assigned dispatcher logs off.

Designing Alert Types

The product can be used to raise many different types of alerts related to crews and activities.

The underlying processes of assigning the alert to a dispatcher, waiting for the user to act upon the alert, and managing the closure of an alert are quite similar; the primary differences may be the conditions under which the alert is raised and closed as well as unique actions a user may take to address it. The base package provides several alert business object that demonstrates typical alert handling use cases.

This section discusses topics to consider when designing custom types of alerts.

When Should The Alert Be Raised?

Understand the business conditions for which to raise alerts of this type.

Event Driven Alerts

If alerts are to be raised upon a specific event, you need to design an algorithm that will be triggered by the event to create an alert. Refer to [Request For Help](#) for an example.

Typically alerts are raised for conditions that have already been met so they are in effect from the time of their creation. However, critical alerts may be raised in advance to ensure a dispatcher is notified of the situation in a timely manner. Refer to [Unacknowledged Emergency Alert](#) for an example.

Monitored Situations

Many alerts are not event-driven, but rather are raised based on a situation that is determined by frequent analysis of the data. For such alerts, an [Alert Creation](#) algorithm needs to be developed based on your business requirements. Consider the following when designing such algorithm:

- Is there a configurable [threshold](#) to be applied before creating an alert?

- Can the alert be [manually closed](#)? If you have to allow manual closure, please be sure to build this logic into the alert creation algorithm to avoid raising repeated alerts for same issue.

When Should The Alert Be Closed?

Understand the business rules that allow the system to automatically close alerts of this type. When applicable, design a corresponding [Alert Evaluation](#) algorithm and specify it on the alert type.

Use The Base Business Objects

You should be able to use the base alert business objects to implement most of your custom alerts. Implement your custom creation and evaluation algorithms and reference them on the alert type.

If you need more details to be captured on an alert or leverage a different lifecycle, your implementation would need to design a new custom alert business object to handle these requirements.

Defining Alert Types

This portal is used to maintain alert types.

You can access this portal from the **Admin Menu > Service Management > Alert Type**.

The following zones may appear as part of the portal's **Main** tab page

- **Alert Type List.** This zone lists all alert types. Broadcast a record to display the details of the selected record.
- **Alert Type.** This zone provides information about the selected alert type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [MI_ALERT_TYPE](#).

Key Performance Indicators (KPIs)

Understanding Key Performance Indicators (KPIs)

Key performance indicators (KPIs) are quantifiable measurements related to crews and activities a dispatcher currently monitors. For example, a KPI might measure the number of idle crews or the number of missed appointments.

Refer to [KPIs](#) for more information.

This section describes various features controlled by KPI configuration.

Calculation Algorithm

A KPI record defines what is being measured and how to measure it. A KPI is either crew shift-oriented (for example, the number of idle crews) or activity-oriented (for example, the number of appointments missed).

NOTE: Logic to calculate the KPI measurement and provide the necessary information associated with it resides in a "Calculation Algorithm" defined on the KPI record. Click [here](#) to see the algorithm types available for this system event.

The calculation algorithm may be designed to use some sort of a threshold that is taken into account, amongst other criteria, when determining the KPI value. The threshold is set on the KPI record and its meaning is only known to the calculation algorithm.

If a KPI is actionable, the KPI value appears as a link on the KPI Summary zone on the CDI portal. This allows the dispatcher to drill in to the KPI specific CDI portal where actions can be taken to address the underlying condition. If a KPI is not actionable, the KPI appears as plain text for information purposes only.

Refer to [Monitoring KPIs](#) for more information.

The KPI calculation plug-in spot supports two modes, **Summary** or **List**. If called in **Summary** mode, the algorithm returns the calculated value of the KPI as well as an indication as to whether it is actionable or not. If called in **List** mode, it returns the list of activities and/or shifts that contributed to the KPI value.

The system calls the KPI calculation algorithm in **Summary** mode for each KPI monitored by a dispatcher and displays the KPI value in the KPI Summary zone on the CDI portal. When the dispatcher navigates to the KPI Specific CDI portal for a selected KPI, the system calls the KPI's calculation algorithm in **List** mode, narrowing the information displayed on the portal accordingly.

Alert Based KPIs

Some KPIs may measure the same underlying conditions as some of the alerts raised by your implementation. For those KPIs, the base product provides a KPI calculation algorithm that counts the number of open alerts of a specific type that are related to activities or crews monitored by the dispatcher.

Note that alerts are considered regardless of whether or not they are assigned to the dispatcher. For example, if there are multiple dispatchers covering the same dispatch area, the KPI counts would appear for all of these dispatchers even though the alerts are distributed between them.

Severity Thresholds

The KPI record defines the normal, warning, and severe thresholds values for the measurement. If the KPI value is less than the normal threshold, the KPI will not be presented to the dispatcher, even if it is actionable.

Designing Your KPIs

The following is a high-level summary of the steps involved in designing KPIs for your organization:

- Determine what the KPI measures and identify whether the KPI is related to activities or to crews.
- Determine whether there is an existing alert type that measures the underlying condition. If so, you can use the KPI calculation algorithm provided with the base package. If not, you must create an algorithm to calculate the KPI that you would later on reference on the KPI record.
- Identify the beginning of the normal, warning, and severe ranges for the KPI.
- Define the KPI Record. Refer to [Defining KPIs](#) for more information.

Defining KPIs

This portal is used to define a Key Performance Indicator (KPI) record.

You can access this portal from the **Admin Menu > Service Management > KPI**.

The following zones may appear as part of the portal's **Main** tab page

- **KPI List.** This zone lists all KPIs. Broadcast a record to display the details of the selected record.

- **KPI Type.** This zone provides information about the selected KPI.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_KPI](#).

Procedures

Understanding Procedure Types

Procedure types represent the type of forms a crew is required to fill out as part of working their shifts.

Refer to [About Procedures](#) for more information.

This section describes various features controlled by procedure type configuration.

Related Entity

On the procedure type you specify whether it is related to an activity type, a vehicle type, or a primary function. Once defined, the procedure types may be referenced on the corresponding entities.

Primary Function procedure types referenced on service classes determine the procedures required when a crew takes on a certain primary function role defined by this service class. This may be at shift start time or when there is a change in primary function mid shift.

Vehicle Type procedure types may be referenced on vehicle types to determine the procedures required when a vehicle of this type is used or changed on a shift.

Activity Type procedure types may be referenced on activity types to determine the procedures required before the crew can start work on the activity.

Questionnaire

Typically, procedures consist of simple questionnaires with predetermined expected answers (yes/no, text, a number, etc). The base product is shipped with business objects to support this type of procedure but you may customize your procedure business object to meet your business needs.

The base product business objects implement a simple questionnaire form that allows you to define the specific questions or steps that need to be followed to complete the procedure. These can be in the form of yes or no questions, text or numerical values that must be entered, valid ranges, etc.

Message categories are used for the text messages for all steps in the procedure. It is recommended that you designate one or more message categories for procedure related messages. For example, you may have a message category for vehicle inspection or activity safety checklist.

NOTE: All custom message categories must be numbered 90000 and above.

Procedure message categories need to be included as part of your deployment type configuration. When adding or changing step messages, you need to create and activate a new mobile application deployment to reflect your recent changes. Refer to [Creating a Deployment](#) for more information.

Effective Dated Procedures

Some types of procedures may only be required at certain periods. If you specify an effective period on the procedure type, the system would only enforce it during that time.

Attendance

You can set the procedure type to require that the crew takes attendance of who was present during completion of the procedure. All of the mobile workers assigned to the crew shift are listed for the crew leader to indicate whether or not they were present.

Override Results

The system does not allow the business activity to commence until all required procedures are passed. If procedure override is allowed, the crew can bypass a failed procedure so it is considered as passed.

Depending on your business practices, the procedure type may or may not be configured to allow the crew to override or bypass a failed procedure as necessary.

Designing Procedure Types

Analyze your procedures and procedure forms to determine how they can be transformed to be used with this feature.

Plan your procedure type configuration as follows:

Classify your procedure forms as being related to activity types, vehicle types, or service classes.

Determine whether crews are allowed to override and proceed the requested change to the shift or activity if the questions are not answered correctly, or if the procedure fails and requires remediation by the crew before the shift or activity can continue.

Determine whether taking attendance is required..

- Review the actual steps to complete the procedure and determine whether the base questionnaire business object may be used or if a custom business object is needed. If using the base questionnaire business object, create message categories to define procedure steps.
- Use the Procedure Type portal to define your procedure types, referencing the procedure business object base or a custom one. Refer to [Defining Procedure Types](#) for more information.
- Associate the procedure types with activity types, vehicle types, and service classes as needed.

Defining Procedure Types

This portal is used to display and maintain a procedure type.

You can access the portal from the **Main Menu > Resource Management > Procedure Type**. You are brought to a query portal with options for searching for a specific procedure type. Once a procedure type has been selected you are brought to the maintenance portal to view and maintain the selected record.

The **Procedure Type** zone on the portal's **Main** tab page provides information about the procedure type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_PROCEDURE_TYPE](#).

Timesheet

Understanding Timesheet Configuration

This section describes various features controlled by timesheet configuration.

Refer to [About Timesheets](#) for more information.

Timesheet Business Objects Control Everything

The base product handles the process of preparing timesheet records for all mobile workers that worked as a crew using the "Crew Timesheet Process" business object. The crew process timesheet record is created using the **Timesheet Process Typeglobal configuration** parameter.

The crew process business object initiates the individual timesheet records for each mobile worker using the "Resource Timesheet" business object. These timesheet records are created using the **Timesheet Typeglobal configuration** parameter.

If these timesheet types are not specified on the Global Configuration record, timesheet functionality is not enabled for your organization.

Your implementation may extend or customize the base business objects or introduce custom business objects as needed. Refer to [Designing Timesheet Types](#) for more information.

Work And Travel Time 'Buckets'

You may track travel time as well as time working on a task using different "buckets" for different task types.

Designing Timesheet Types

Timesheet functionality allows your organization to more easily use the time information which is collected within the application as shifts are completed. Analyze how timesheet information is used by your organization and design customizations as needed.

This section discusses topics related to customizing and extending the base timesheet functionality.

Integration with Timekeeping Systems

Timesheet information can be exported to external timekeeping systems or other applications as needed. The actual process to interface a timesheet record to an external system is not provided out of the box and has to be customized.

A timesheet record in the processed state is ready for extract to an external system. Depending on how your system integrates with the external timekeeping system, you can create a custom algorithm that on entry to the processed state, sends a message to the external system.

Alternatively the external system may poll for processed timesheet records.

Approval

The "Resource Timesheet" business object assumes no further approval is necessary once the mobile worker accepts their timesheet record and marks it reviewed. Once reviewed, the timesheet record is considered processed and ready to be interfaced to an external system, if such custom process has been implemented.

You may introduce custom logic to require manager approval as follows:

- Set up a custom algorithm to automatically transition a reviewed timesheet record to the **Pending Approval** state instead of the **Processed** state. The algorithm should be based on the **Generic Business Object Status Monitor (F1-GEN-BOMON)** algorithm type.
- Plug your custom algorithm on the **Reviewed** state before the base algorithm.
- A user would need to approve or reject the timesheet record as needed.

Configuring Timesheet Options

Follow the steps in the order they are described in this section to configure timesheet options.

Defining Timesheet Types

This portal is used to display and maintain a timesheet type.

You can access the portal from the **Main Menu > Resource Management > Timesheet Type**.

The following zones may appear as part of the portal's **Main** tab page

- **Timesheet Type List**. This zone lists all timesheet types. Broadcast a record to display the details of the selected record.
- **Timesheet Type**. This zone provides information about the selected timesheet type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_TIMESHEET_TYPE](#).

Defining Custom Timesheet Time Types

By default travel and work time is summarized across all task types. If your business requires that travel and work are summarized separately for specific types of work you may introduce new timesheet time types in the **Timesheet Time Type (M1-TimeSheetTimeType)** extendable lookup and reference them on the corresponding task types. The timesheet time types should be set up to conform to your external timekeeping system. For example, you may want all meter work to be grouped into one timesheet time type and all collections work in another

Enable Timesheet Functionality

To enable timesheet functionality, reference the timesheet types for the crew process and for the mobile worker in the [Global Configuration](#) record.

This is also where you state whether or not mobile workers are allowed to adjust the timesheet times as calculated by the system at the end of their shift.

Also, ensure that the **Timesheet Monitor** batch process is run periodically. The batch is mainly needed to handle situations where timesheet calculation is waiting for some task completion errors to be resolved. Refer to [M1-TIMSHT](#) batch control for more information.

Mail

Defining Mailing Lists

This portal is used to maintain mailing lists.

Refer to [About Mail](#) for more information.

You can access this portal from the **Admin Menu > Mail Management > Mailing List**.

The following zones may appear as part of the portal's **Main** tab page

- **Mailing Lists.** This zone lists all mailing lists. Broadcast a record to display the details of the selected record.
- **Mailing List.** This zone provides information about the selected mailing list.
- **Mailing List Users.** This zone allows you to view users currently in the selected mailing list as well as remove users from the list.
- **Add Users.** This zone allows you to add more users to the selected mailing list

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [MI_MAILING_LIST](#).

Defining Mail Templates

This portal is used to maintain mail templates.

Refer to [About Mail](#) for more information.

You can access this portal from the **Admin Menu > Mail Management > Mail Template**.

The following zones may appear as part of the portal's **Main** tab page

- **Mailing Lists.** This zone lists all mailing lists. Broadcast a record to display the details of the selected record.
- **Mailing List.** This zone provides information about the selected mailing list.
- **Mailing List Users.** This zone allows you to view users currently in the selected mailing list as well as remove users from the list.
- **Add Users.** This zone allows you to add more users to the selected mailing list

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [MI_MAIL](#).

Transfer of Goods

Understanding Transfer of Goods

This section describes topics related to configuring transfer of goods options.

Refer to [About Transfer of Goods](#) for more information.

Depot Related Business Objects

The following entities may need to be defined using designated depot related business objects:

- **Activity Types.** Depot based activities involve either loading goods at a depot center and delivering them to a customer location or, collecting goods from a customer location and returning them to a depot center. Whether the activity is picking up goods or dropping them off is defined on the activity's type. The activity type needs to be defined using the depot based business object.
- **Activities.** Depot related activities specify depot information and other goods related information. These activities and their assignments need to be managed by depot related business objects.
- **Break Types.** If breaks need to be taken at the depot, their break type should be defined using the depot related business object.
- **Vehicle Types.** If your business leverages maximum [load capacity limits](#) of vehicles, these are defined on the vehicle type. To specify these limits on a vehicle type your implementation must use the capacity related vehicle type business object that supports vehicle capacities.
- **Crews.** For single person crews that need to be restricted to specific depots use the depot related crew business object.
- **Crew Shifts.** If your business leverages maximum [load capacity limits](#), these may be overridden on a specific shift for exception handling. This functionality requires that you use the depot based shift business object.

Depot Task Types

The scheduler generates depot tasks as needed when planning depot runs to and from depot centers.

The system uses the depot task type specified on the depot or depot profile level and, if not specified, it uses the depot task type referenced on the [global configuration](#), to generate the depot tasks.

NOTE: If a depot task type cannot be determined for a depot, the scheduler disables the scheduling of all activities associated with the depot.

Using Capacity Limits

Using load capacity limits as a scheduling consideration is optional. If this functionality is needed, you may specify up to 5 types of capacity limits on designated lookup fields.

The types of load capacity limits to consider such as weight, volume, size etc must be configured before they can be used to describe the load on the activity, the maximum load a vehicle can carry, as well as the maximum load a depot can handle during a time windows.

Refer to [Defining Types Of Load Capacity](#) for more information.

Configuring Transfer of Goods

The following guidelines provide a basis for transfer of goods configuration.

Defining Types Of Load Capacity

If you are using cumulative capacities to describe goods, such as weight, volume etc, override the descriptions of up to 5 capacity types defined in the lookup **MI_CUMULATIVE_CAPACITY_FLG** to enable them.

If you are using non cumulative capacities to describe goods, such as size, override the descriptions of up to 5 capacity types defined in the lookup **M1_NONCUM_CAPACITY_FLG** to enable them.

If you are using maximum cumulative capacities to describe maximum load, such as maximum weight, maximum volume etc, override the descriptions of up to 5 maximum capacity types defined in the lookup **M1_MAX_CAPACITY_FLG** to enable them.

If you are using maximum non cumulative capacities to describe maximum load, such as maximum size, override the descriptions of up to 5 maximum capacity types defined in the lookup **M1_MAX_NONCUM_CAPACITY_FLG** to enable them.

Set up corresponding scheduler configuration costs for applicable cumulative and noncumulative capacity types. For example, if you defined "cumulative capacity 1" as "weight" then make sure you provide the cost for weight in the "cumulative capacity cost 1" parameter.

Defining Depot Related Task Types

The following task types need to be set up to handle depot related work:

- Define depot task types as needed. Refer to [Depot Task Types](#) for more information.
- If the crew handles the loading of process of goods at the depot based on a detailed list of items provided by the host system use the **Depot Task Items (M1-DepotTaskItems)** business object.
- If the crew does not handle an itemized load, use the **Depot Task (M1-DepotTask)** business object.
- Define depot related activity types.
- Use the **Depot Related Activity Type (M1-DepotRelatedActivityType)** business object to define types of activities that involve distribution or collection of goods.
- Reference the **Depot Related Activity (M1-DepotRelatedActivity)** business object on these types of activities.
- If you are using only this business object to define activity types, you may want to update the other activity business objects to not allow new instances.
- If breaks should be scheduled at the depot, define depot break types.
- Use the **Depot Break Type (M1-DepotBreakType)** business object to define break types that should be scheduled at the depot.
- If you are using only this business object to define break types you may want to update the other base break type business object to not allow new instances.

Refer to [Defining Task Types](#) to set up all these task types.

Defining Depot Related Vehicle Types

If your implementation needs the ability to specify maximum load capacities on vehicle types, use the **Vehicle Type with Capacities (M1-VehicleTypeWithCapacity)** business object.

If you are using only this business object to define vehicle types you may want to update the other base vehicle type business object to not allow new instances.

Refer to [Defining Vehicle Types](#) to set up vehicle types.

Defining Depot Related Crew Types

For single person crews that should be restricted to specific depots, use the **Depot Related Single Person Crew (M1-DepotSinglePersonCrew)** business object. Reference this business object when defining crew types.

If you are using only this business object to define crews you may want to update the other base crew business object to not allow new instances.

Refer to [Defining Crew Types](#) to set up vehicle types.

Defining Depot Related Shift Types

If your implementation needs the ability to specify override maximum load capacities on a specific shift and other depot restrictions, use the **Depot Related Shift (M1-DepotRelatedShift)** business object. Reference this business object when defining shift types.

If you are using only this business object to define crew shifts, you may want to update the other base crew shift business object to not allow new instances.

NOTE: Depot related information is only applicable for actual shifts and not templates. The standard shift template business object should be used on the shift type even for depot related shifts.

Defining Depots

This portal is used to maintain depots and depot profiles.

Refer to [Depots](#) for more information.

You can access this portal from the **Admin Menu > Resource Management > Depot**.

The following zones may appear as part of the portal's **Main** tab page

- **DepotList.** This zone lists all depots. Broadcast a record to display the details of the selected record.
- **Depot.** This zone provides information about the selected depot.
- **Override Time Windows.** This zone lists any override time windows for the depot. You can also use this zone to add a new override time window.
- **Cutoffs.** This zone shows a history of cutoff requests for the depot. The zone is only visible for a distribution depot.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [MI_DEPOT](#).

Contractor Management

Understanding Contractor Configuration

This section describes concepts related to contractor management administration. This is only applicable if your company utilizes the contractor functionality.

Review your contractor agreements to determine which contracting model it follows and configure the system accordingly.

Refer to [About Contractors](#) for more information.

Contractor Users

Irrespective of the contracting model used, any contractors that require access to the application must be set up as users and their user record must reference the resource group of their company. This is defined on the **Miscellaneous** tab of the User portal.

The system also supports the following method of associating contractor users with their company records:

- Mark the user as a contractor without specifying the specific company they work for. This is also defined on the **Miscellaneous** tab of the User portal.
- When done setting contractor users, go to the Resource Group record for each contractor company and associate the corresponding users to it.

NOTE: If not associated with a contractor's resource group users are considered company users. As such, they are allowed access to entire data.

Contractor users must only be allowed access to the certified application components for contractor users to ensure they have access to data related to their company only. Refer to [Shift Based Model Security](#) and [Capacity Based Model Security](#) for more information.

Shift Based Model Security

In the shift based model only mobile workers need to have access to the system. All other user roles are assumed to be company users.

The base product secures the following portals to ensure the data accessed by a contractor mobile worker is restricted to their data only:

- Activity Search. All base query options are secured.
- Activity
- Assignment
- Crew Shift Search. All base query options are secured.
- Crew Shift
- Break / Non Productive Task
- POU
- Depot Task
- My Calendar portal

Contractor mobile workers must be associated with their contractor company on the user record and belong to a user group that provides limited access to these transactions.

NOTE: Do not provide access to any transaction that is not in the secured list.

NOTE: It is recommended to define a designated shift based mobile worker user group as well as a template record that belongs to that user group. Use the template to set up shift based contractor users. Refer to the **Accelerator** environment provided with the product for a recommended user and user group configuration.

Capacity Based Model Security

NOTE: If your company integrates with the contractor company's system via web services, there is no need to grant access to any contractor user.

When applicable, part of your business practice might be to provide contractor users access to the Contractor Work Management portal or the contractor mobile application to manage work committed to their company. Any individual who will need to access these parts of the application needs to be set up as a contractor user.

Refer to [Managing Committed Work](#) for more information.

The base product secures the following portals to ensure the data accessed by a capacity based contractor users is restricted to their data only:

- Contractor Work Management
- Activity Search. All base query options are secured.
- Activity
- Assignment
- Capacity Search. All base query options are secured.
- Capacity

Contractor users must be associated with their contractor company on the user record and belong to a user group that provides limited access to these transactions.

NOTE: Do not provide access to any transaction that is not in the secured list.

NOTE: It is recommended to define designated capacity based manager and worker user groups as well as template records that belongs to these user groups respectively. Use the appropriate template to set up contractor managers and workers as needed. Refer to the **Accelerator** environment provided with the product for a recommended user and user group configuration.

Contracting Work

Settings on the activity type indicate whether the activity should be worked by company crews only, any resource (including contractors), or restricted to shift-based crews (which can be composed of internal resources and/or contractors).

Review your activity types to determine which types of activities can be worked by contractors.

Configuring Shift Based Contractors

Plan your shift based contractor configuration as follows:

- Define a Resource Group Type for shift based contractors. Refer to [Defining Resource Group Types](#) for more information.

For each shift based contractor company, complete the following set up:

Set up a resource group for the contractor company based on the shift based resource group type.

- Set up their mobile workers as contractor users in the system, associating the user records with the contractor company's resource group. Refer to [Contractor Users](#) for more information.

Set up their mobile workers, vehicles, and crews referencing the contractor company they belong to.

Create crew specific shift templates or subscribe these crews to common templates to generate their shifts just as you would for company crews.

Set up MDTs as needed if these crews would be using the mobile application.

NOTE: Only mobile workers should be provided with user access to the system. All other roles are assumed to be company employees. Your organization should define contractor users based on user groups that provide them with access only to the certified transactions that apply data security.

Configuring Capacity Based Contractors

Plan your capacity based contractor configuration as follows:

- Define a Resource Group Type for capacity based contractors. Refer to [Defining Resource Group Types](#) for more information.

For each capacity based contractor company, complete the following set up:

- Set up a resource group for the contractor company based on the capacity based resource group type. Review the contractual agreements with the contractor company to properly reflect expectations around notification lead time, approval time, and service area coverage.
- When applicable, set up their users as contractor users in the system, associating the user records with the contractor company's resource group. Refer to [Contractor Users](#) for more information.
- Create capacity weekly templates to generate their capacities as per contractual agreements.
- If the contractor company integrates with your system via web services, you need to customize the message exchange between their services and the base product services. Refer to the [Contractor Management](#) integration section for more information.

Defining Resource Group Types

This portal is used to maintain resource group types. Resource group types define classifications for various types of contractor companies doing work with your organization.

Refer to [Resource Groups](#) for more information.

You can access this portal from the **Admin Menu > Resource Management > Resource Group Type**.

The following zones may appear as part of the portal's **Main** tab page

- **Resource Group Type List.** This zone lists all resource group types. Broadcast a record to display the details of the selected record.
- **Resource Group Type.** This zone provides information about the selected resource group type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [MI_RESRC_TYPE](#).

Configuring Capacity Model Options

This section describes topics related to configuring capacity based contractors options.

Defining Capacity Types

Capacities represent an external contracting company's agreement to take on a certain amount of work of a certain type in specific service areas for a period of time. Capacities can be defined by number of hours or number of activities, as defined on their capacity type.

This portal is used to maintain capacity types.

You can access this portal from the **Admin Menu > Resource Management > Capacity Type**.

The following zones may appear as part of the portal's **Main** tab page

- **Capacity Type List.** This zone lists all capacity types. Broadcast a record to display the details of the selected record.

- **Capacity Type.** This zone provides information about the selected capacity type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_CREW_SHFT_TYPE](#).

Integration with Contractor Systems

This step is only relevant if you need to integrate with the contractor company's system.

When an activity is committed to a capacity contractor, the system sends the activity details to the contractor. Such message is also sent when key activity details have changed or the activity has been recalled from the contractor company.

Outbound messages to a contractor company's system are directed to the external system referenced on their resource group record. Set up an [External System](#) record to support such communication.

Refer to [Managing Committed Work](#) for more information.

The kind of messages supported are configured in an extendable lookup. Add a record to the **External System Messages (M1-ExternalSystemMessages)** extendable lookup that matches the external system code you have set up for the contractor company's system. Populate the appropriate message information to enable it or mark it as not applicable.

Refer to [Configuring Outbound Messages](#) for more information.

Mobile Application

Understanding Mobile Devices

Mobile devices such as laptops, cellular phones, tablets, and other mobile devices may be used to interface with the system via the supported mobile applications. The mobile application and data reside locally on the mobile device allowing the crew to work offline as needed.

This section describes concepts related to device management administration.

Mobile Platform

The mobile solution consists of a thin runtime installed on the device, the mobile applications, and metadata. The mobile applications are built using HTML5 and JavaScript on top of the Oracle Utilities Mobile Library (OURL) platform. The application communicates with the server via RESTful Services exposed by the main application.

The mobile applications require metadata (such as lookups, business object definitions and more) to function. All files for all mobile application reside in a shared location on the server. It is the metadata that controls the behavior and functionality unique to each mobile application. The metadata package for each mobile application is also known as a deployment.

The URL entered at application startup is the URL used for both communication with the server and locating the mobile application HTML5 and JavaScript files. The deployment is downloaded once the user selects which mobile application they will be using.

Refer to [Understanding Deployments](#) for more information.

This approach makes it easier to push out applications changes to crews without having to reinstall the application on the device. The footprint on the device is kept to the minimal allowing the rest to be downloaded at start time from the server. The thin runtime can be downloaded from the application store and is assumed to be rarely updated. Customers can make as many changes as they like to the application HTML5 and JavaScript files without having to physically reinstall anything on the device.

NOTE: This section does not apply to mobile applications implemented using the legacy Java based mobile communication platform. These applications are still supported in this release for existing customers only. Support for such applications will be dropped in the next release. Refer to [Understanding The Legacy Mobile Platform](#) for more information.

Mobile Device Terminals (MDT)

You must create an MDT record for each physical device that will use a mobile application to communicate with the main server application. The MDT record has to be uniquely identified by a device tag.

The product provides a replicator tool on the MDT portal that may be used to create many MDT records that share the same device tag prefix.

An MDT Type defines attributes common to mobile devices of a certain type such as the [mobile platform](#) applicable to that type, logging, ports, attachment maximum size, and so on.

NOTE: You should only use the **Restful Service Device** mobile platform to take advantage of the base products mobile applications. All other mobile platforms represent various options of the legacy Java based mobile communication platform that will be dropped in the next release.

Registration

The first time a mobile user runs the mobile application, they will be required to register their mobile device.

The system will ask for the MDT Tag and the URL of the server application. The MDT Tag is a unique, user-defined identifier that is assigned to the device when the MDT record is created on the server application.

As part of the registration process, the system posts the device's unique ID on the MDT record on the server. Various mobile platforms like iOS, Android, and Windows provide a system level API to get the device's unique hardware ID which is used as the device's unique ID.

Once the device is registered, the mobile user will be able to log on to the application from that device without further prompting.

All the platforms return the same ID consistently, except for iOS that assigns a unique ID for the application when it is installed. This means that for iOS, the MDT's unique ID changes each time the application is uninstalled and installed again. In this situation, you would need to reset the unique ID stamped on the MDT record after reinstalling the application. Use the **Reset Registration** action on the MDT portal to do so.

Sending GPS Data

The MDT Type specifies whether the device has GPS capability, the GPS port, how frequently to poll the device, and how frequently to send data to the server. Whether or not GPS data is sent to the server is controlled by the mobile application.

When this functionality is enabled, the mobile application sends batches of GPS data to the server in the frequency configured on the MDT type. The GPS data is associated with the crew using the device.

Refer to [Sending Current Location](#) for more information.

Managing Attachments

The MDT Type may also set a limit for the total size for attachments on mobile devices of that type. When a user attempts to download an attachment, the system validates that the total size of the download plus what is already on the mobile device will not exceed the maximum for the mobile device.

The total size specified should take into account common attachments that are included as part of the mobile application [deployment part](#) configuration. These are automatically downloaded along with the mobile application.

Data Encryption

You have the option of enabling or disabling data encryption globally as well as for specific devices.

Use the **Data Encryption**[global configuration](#) parameter to either turn it on or off globally.

Each MDT has a data encryption setting which you can set to either use the global default, or turn it on or off for this device.

Understanding Deployments

A deployment refers to a specific cut of application metadata and control records needed to support a mobile application of a specific type and language. The deployment is downloaded to the mobile device terminal (MDT) by the corresponding mobile application.

This section describes topics related to managing mobile application deployments.

Deployment Types

A deployment type represents the configuration needed to support an application of a given type. You can configure multiple types of applications by setting up a deployment type for each.

A deployment type configuration defines the following aspect of the mobile application:

- The application items included in the application, by referencing deployment parts.
- The initial function that starts the business flow on the device when the application is launched.
- The MDT Types compatible to run this type of application.
- The authorized user groups that are allowed to use the mobile application.
- The application message categories used by the mobile application.

Deployment Parts

Deployment parts provide a way to create reusable groups of application items. A deployment part is a collection of application components, such as business objects, lookups, task types, and many more that are needed to support the mobile application.

The base product provides preconfigured deployment parts that include the base mobile application components. You only need to configure parts to include your custom entities. If you need to add custom deployment items, it is strongly recommended that you include them in designated custom deployment parts and not directly add them to base product deployment parts. Make sure to include your custom deployment parts as part of your deployment type setup.

Create deployment parts to represent subsets of your custom deployment items based on their type or usage. You should share deployment parts across deployment types when applicable to prevent redundancy and improve management of deployment configuration. For example, use a designated deployment part for control data, another for items that are common across platforms, another with user interface items for a specific platform etc.

To simplify the collection of items in a deployment part it is recommended that you create an export bundle per deployment part so that implementers can add their new or changed items directly to the corresponding bundle using the dashboard zone. Once items are collected, the bundle can be uploaded to the deployment part. You can always maintain items manually on a deployment part.

Consider where attachments must be automatically pushed to crew and add them to your deployment parts.

Deployment

A deployment can only be created using the **Create Deployment (M1- DPLOY)** batch process. Once your deployment type setup is complete, submit the batch to create a new deployment for your deployment type for a specific language.

Once the batch job runs and the deployment has been created, it has to be manually activated on the deployment portal before it can be deployed to any devices.

Out of Date Deployments

Deployments become out of date when a new mobile application version is released, the deployment type configuration changes, or any underlying application component that is part of the configuration changes.

Mobile users are warned at login to their mobile device that they are using an out of date deployment. The user is allowed to continue, but is advised to contact their administrator to synchronize their deployment. Refer to [Downloading Deployments](#) for more information.

The product provides a batch process that evaluates existing active deployment records and marks those that are out of date if it detects any metadata changes. The background process also purges out of date deployments that are older than a specified number of days.

NOTE: The **Deployment Evaluation and Purge** batch process should run on demand or periodically based on your business needs to avoid situations where out of dates deployments are being used. Refer to [M1-DPUTD](#) batch control for more information.

Also note that as part of accepting a new product release, all existing deployments are marked as out of date. New deployments are required to be created and activated.

When an active deployment is marked out of date, the system creates a To Do entry in order for a user to create a new deployment as needed. Creating a new deployment for the same deployment type and language automatically closes any remaining To Do entries reported for out of date deployments for the same configuration.

Downloading Deployments

Once a deployment has been activated, it is available for download to mobile devices. When the mobile user logs on to the mobile application, the system looks for the most recent version of all active, qualified deployments based on the user's user group, language, and MDT type.

If no deployments are found, the system displays an authorization error message and prevents the user from continuing.

If only one deployment is found and it is the same as the one currently deployed, then the deployment is not downloaded.

If only one deployment is found and it is newer than the one currently deployed or there is none currently deployed, then the deployment is downloaded to the mobile device.

If more than one deployment is found, the user can select the one they want to download.

NOTE: A new deployment will not be downloaded if unprocessed data from a previous shift still exist on the device. The system warns that user of the situation. If the user is authorized to use the deployment currently on the device, it will be used; otherwise, an authorization error will be displayed

Mobile Application Versioning

The mobile application communication architecture involves both client side and server side elements. These must be compatible or else errors will occur. To enforce compatibility, the system validates the mobile application version as follows:

- When the mobile device attempts to connect to the server, the system versioning check validates whether or not the mobile application version on the mobile device is compatible with the mobile application version on the server. If the versions are not compatible, the device is not allowed to work online and no communication is sent to and from the server.
- In this situation, the mobile device must be upgraded to the correct version before it can work online again. If there was data on the mobile device, the user can continue offline and manually collect any data they need before the mobile device is upgraded.
- When a deployment is created on the server, it is marked with the version number of the server side mobile application components that created it. Then when a mobile device logs on to download a deployment, only deployments with compatible mobile application versions can be downloaded. This prevents the incompatible version situation described above.

NOTE: Before the server is upgraded, all crews using the mobile application must be allowed to end their shift successfully from their mobile devices and logoff. This will prevent the need for manual data copying. Use the crew shift search query option to find such shifts and coordinate their successful completion before server side upgrade.

NOTE:

NOTE: Also note that, with each new product version, existing deployments stamped with the older version are marked as out of date.

The **About** screen on the mobile device shows the mobile application version number. This version is also marked on the MDT record on the online mobile device screen. Deployments show the mobile application version as well on the online user interface.

Setting Up Deployments

This section summarizes steps involved in configuring your deployment types:

- Review the various types of devices your crews would be using and set up corresponding MDT types.
- Define MDT records for each mobile device.
- Create deployment parts to represent subsets of your custom deployment items based on their type or usage.
- Set up a deployment type for each type of application you need to support. For example, the product supports a mobile application for a crew to work their schedule, and another application for a contractor worker.
- Associate deployment parts to appropriate deployment types.
- Associate MDT Types to appropriate deployment types.
- Identify the user groups authorized to use each type of mobile application and associate them with appropriate deployment types.
- Associate MDT Types to appropriate deployment types.

- When you have finished defining each deployment type, you must run a batch process to create a deployment for it and each language supported by your company.
- Activate each deployment.

Understanding The Legacy Mobile Platform

This section provides information relating to the Mobile Communications Platform (MCP).

NOTE: Custom mobile applications implemented using the legacy Java based mobile communication platform are still supported in this release for existing customers only. Support for applications implemented on the legacy platform will be dropped in the next release.

Mobile Application Connection Modes

The mobile application can be used in a disconnected or connected mode.

Disconnected Mode

In disconnected mode the mobile application and data reside locally on the mobile device allowing the crew to work offline as needed. This means the physical device has to be compatible with the mobile application requirements it runs locally.

You must create an MDT record for each physical device that will communicate with the main server application.

Refer to [Understanding Mobile Devices](#) for more information..

Connected Mode

In connected mode neither data nor the mobile application reside locally on the accessing mobile device. Instead the data and mobile application reside on the server and the user must be connected to the server at all times using their standard browser to access the mobile application. This should not be confused with the server application. This is the mobile application presented in a regular internet browser.

The actual physical device used to access the mobile application in a connected mode is irrelevant since the application is not run locally and therefore registering the specific device in the system is not needed. The system however, still needs an MDT record for the mobile application to work properly, but it is not related to the physical accessing device.

To access the mobile application in connected mode, the user needs to reference your organization's mobility server URL in a browser, as they would any website, and enter their username and password. After the user is authenticated, the system allocates an MDT record of a connected platform MDT Type from a pool of available records to the user. If all connected mobile devices are already in use by other users, the system duplicates an existing connected MDT record to create a new MDT record and allocates it to the user.

It is possible to log on with the same user from multiple devices. The first login is considered the "primary" user, has read/write abilities and receives any alerts or notifications delivered during login. All other logins with that same user can also edit and update data but do not receive notifications and alerts. Since all data is shared on the server in connected mode, all users have access to the same data. An icon on the navigation bar allows users to take over as the primary user as needed on a given device.

In connected mode, the system ignores the display option value configured on the MDT Type for the mobile device and instead dynamically changes the display option based on the available screen resolution (once per login). This means that for iPads and newer Android devices, which typically use 1024x768, the system recognizes them as X VGA (laptop). This is important when you are selecting a deployment; for these devices, they should select a laptop deployment.

Once allocated to a user, the mobile device is stamped with that user and is not available to another user. The system generates new MDT records upon request but it requires at least one to be defined manually. To speed up the mobile device allocation process it is recommended to define a pool of connected MDT records in advance for as many crews that access in that mode daily.

Regardless of the connection mode, when a mobile worker starts a shift from the mobile application, the mobile device they have used is stamped on the started shift. If connected mode was used and the user closes the browser and logs in again, the system re-assigns the same mobile device stamped on the started shift to which the user belongs. Moreover, any user that is part of that started shift and uses connected mode to access the mobile application would be assigned the same mobile device so they can all share data associated with working their shift.

When the shift is completed and all messages are processed, transactional data is cleared from the connected mobile device data store on the server and the mobile device it is made available to all.

As with all mobile device specific data, log files are also stored on the server and uniquely identified by mobile device.

The following functionality is not supported when accessing the mobile application in connected mode: GPS data is not captured or sent to the server, encryption, attachments, capture picture and capture sound functionality.

NOTE: Logic to generate and assign a virtual MDT when using the mobile application resides in a "Determine Connected Device" algorithm on the installation options record. This algorithm is only required if your organization uses the connected mode. Click [here](#) to see the algorithm types available for this system event.

Configuration Tools

Standard framework configuration tools, such as business objects, scripts, UI maps, algorithms, etc., are used to create, extend and customize the mobile application.

It is important to note that the mobile framework is a slightly scaled-down version of the Oracle Utilities Application Framework.

Business Objects

Entity maintenance is only performed via interaction with business objects. The business object layer does not interact with an underlying maintenance object layer, but rather is based on a completely different data model architecture. This means that all business rules to be performed on the mobile application need to be defined as business object rules.

The business object framework is limited as follows:

- Automatic state transition is not supported.
- Transitory states are not supported.
- Only two plug-in spots are supported.
- MCP Enter
- MCP Post Processing
- BO invocation for read, add, replace, and delete actions is supported.

The same business object definition that governs an entity on the server application also governs it on the mobile application; however, different business rules apply. On the mobile, entity-specific business logic must be triggered by these MCP-specific business object plug-in spots only.

These MCP Business Object algorithms are modeled after the server concept of the corresponding algorithms, but apply only to the mobile device and have a different hard parameter interface. Unlike the server framework, when these MCP algorithms are called the entity in the local database still reflects the unchanged version, while the changed entity is provided in memory as an input hard parameter. Further changes to the entity should be made to the input copy, not directly.

Scripting is the only supported programming language for these plug-in spots and java scripting for user interface rules. Service scripts and plug-in scripts are limited to XPATH version 1.0.

As mentioned earlier, the data model architecture is completely different from the server framework. Instead of individual maintenance objects, the mobile framework is based on a single XML data storage for business object instances. This also means direct access to tables for query purposes is not possible. All queries need to be performed by calling the MCP Query mobile framework business service. Refer to that business service for more information.

User Interface

There is no notion of portals, zones or BPA scripts on the mobile device. Service scripting syntax in the mobile framework supports a "Terminate with Map" command that allows user interface interaction. This is the only means of presenting data to and obtaining data from a user.

Different devices may call for a different UI map to maintain and display an entity's details. For example, a handheld device has less real estate than a laptop. To accommodate differences in screen size, the MDT type record provides a display option attribute, which specifies the type of display (such as Windows Mobile VGA or Windows XVGA). Further, each business object that is maintained by the mobile application, like task and shift business objects, has an option where you can specify the mobile script to use for each supported device display option.

Navigating Back to a Previous Map

When navigating to a UI Map with this tag, the tag tells the MCP to push the previous UI Map name and data onto the stack of backable maps. This is in case users execute a "back" at some point. In this case the MCP displays the UI Map name and data from the stack and navigates to it.

If this tag is not on the UI Map being navigated to, then the backable map stack is cleared.

For example:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html >
<body oraMCPsupportsBack="true">
...
...
</body>
</html>
```

Clickable Phone Numbers

You can use the oraPhone and oraPhoneContact attributes to identify phone numbers on UI Maps.

For example:

```
<td oraPhone="true" oraPhoneContact="boGroup/customerInformation/contactName" oraField="boGroup/customerInformation/mainPhone"></td>
```

Device Vs. Server Time Zones

The mobile framework supports the user working in the local time zone for their device while at the same time managing the conversion to and from the base time zone when communicating data to and from the server.

Mobile Application Testing

The system provides tools to help implementations test their mobile application. This provides improved efficiency in building mobile UI maps and other mobile application components to help reduce implementation time lines.

The **Base Quick UI Map (M1-MCPDevTools)** deployment part has to be included in the deployment type being tested for the tools to be enabled. During implementation, the development tools deployment part can be included in any deployment type to enable testing.

NOTE: When ready for production, the deployment part should be removed from the deployment types to disable the developer features.

UI Map Testing

This tool allows a user to quickly test their mobile UI map on the device without going through the entire mobile application data setup.

For the purpose of this tool you need to launch a mobile application on any MDT. The deployment may be based on your actual deployment type or it can be a standalone deployment type that only includes the **Base Quick UI Map (M1-MCPDevTools)** deployment part. If using a testing deployment type you need to use the service script **Developer App Initial Service Script (M1-MCPIUITST)** as its Initial Service Script.

Follow these steps to test a UI map:

- Launch your mobile application.
- Access the UI Map portal and make the necessary edits. You may also provide test data as part of the UI map XML element, just like you do when testing server UI maps.
- Enter your MDT Tag in the **Mobile Map** zone on the dashboard, and click the **Test** button. If you typically use the same MDT ID for testing you can reference it as a characteristic on your user record and the zone would default it when displayed .

A new window opens to display the changes on the MDT using test data if defined on the UI map. When this is initially launched, the MDT needs to have already downloaded a deployment and be displaying an application screen (not one of the initial MCP screens such as Register, Logon, or Deployment). This will allow you to logon to the MDT and the device will be ready to receive RSI messages from the server to test UI Maps.

This feature is not supported for the Connected MCP platform.

Refresh the Mobile Application

This tool allows a user to refresh all items in the deployment currently used by their MDT.

Follow these steps to test a UI map:

- Launch your mobile application.
- Change a mobile application item, for example a UI map.
- Click the **Refresh** button on the mobile application. This process rebuilds the deployment and sends it to the mobile device. It does not create a new deployment but rather update an existing deployment capturing the time it was updated.

This one click approach can be used instead of re-creating the deployment on the server each time there are metadata changes done. This will re-generate the deployment on the server and replace the deployment already present on the MDT.

Navigation and Display

This section describes configurable aspects of the mobile application work flow and display.

Everything Starts with the Initial Script

When a user logs on to the mobile application, the mobile framework invokes the Initial Script defined for the specific deployment on the device.

The base package Initial Script demonstrates how a crew obtains their shift for the day, starts the shift, and then is presented with their tasks for the day. The crew can work each task and use other actions to manage their shift.

As mentioned earlier, the maintenance user interface associated with each task and shift business object can easily be replaced with your custom user interface, as these are defined as business object options. If you need to customize common parts of the mobile application that are not business object specific, you can do that by cloning the Initial Script and adjusting it as needed.

Indicator Bar

The user interface on mobile devices includes an indicator bar which shows various indicators such as new email and connectivity. You can introduce your own indications by doing the following::

- Add the special icons to the **Mobile Device Indicator Bar (M1-MCPIndicator)** extendable lookup.
- Add custom logic to the mobile application to properly add or remove the indication from the bar as needed. To do this you will need to call the "MCP Indicator Bar" Manager business service.

The Tool Bar

The mobile application displays a toolbar below the application indicator bar. The toolbar is available from every application screen, but the buttons that are displayed at any time vary depending on the current application screen you are viewing.

General Actions Menu

The Task List user interface supports an actions icon that when clicked presents the user with a list of actions a crew can take during their shift, for example changing their primary function, changing crew allocation, going offline, completing their shift etc.

To add more custom actions

- Copy the base actions script and UI map to create your custom script and map. Make sure that you to include the base script as your custom script's schema for upgradability reasons.
- Enhance them to support your custom actions.
- Copy the base initial script and set it to call your custom actions script.
- Use your custom initial script in your deployments.

Task Actions Menu

When on a task related user interface a menu actions icon appears if configured on the task business object. The base activity business objects are configured with a BO option to manage activity related actions such as creating a new activity for the same location, capturing contents for this activity etc.

To add more task related custom actions

- Copy the base actions script and UI map to create your custom script and map. Make sure that you to include the base script as your custom script's schema for upgradability reasons.
- Specify your actions script as a BO option with a higher sequence number than the base actions script.

Remote Script Invocation

Data and updates sent to the crew, as well as updates and completion data received from the crew, rely on the Remote Script Invocation (RSI) mechanism supported by the mobile framework.

Data exchange is achieved by calling a framework business service to place an asynchronous request to invoke a service script and its corresponding data on a remote application. The system provides both server and mobile device versions of this service. The server version places the request to be performed on a specific MDT, whereas the mobile device version places the request on the server. A synchronous RSI call to the server is also supported.

NOTE: Logic to process remote messages sent from the mobile device resides in a "Remote Script Invocation" algorithm on the installation options record. Click [here](#) to see the algorithm types available for this system event.

Messages Sent to the Crew

Various algorithms implement logic to send messages to the crew's mobile device. For example, the Task Synchronization algorithm is responsible to dispatch tasks and send updates to dispatched tasks.

A call back notification functionality is supported for RSI messages issued from the server only.

With this functionality, the process posting the RSI message to the crew may request to be notified if delivery of the message to the crew takes too long. For example, this may be used to raise an alert when an activity takes too long to dispatch to a crew.

Messages Received from the Crew

A synchronous form of an RSI call may be issued from the mobile device only, such as when the crew request and start their shift for the day.

On the server, messages sent from the crew are processed by the Remote Script Invocation installation plug-in. This allows a better handling when an application error occurs. The base algorithm persists the message as a crew message business object and if an error is encountered while processing the message a To Do Entry is raised for a user to correct the issue and reprocess the message.

Remote messages in an error state are periodically retried by the remote message monitor batch process.

Device Capabilities

The capabilities section on the MDT Type allows you to define the various features or functions supported by mobile devices of this type, such as capturing photo or sound, attaching supporting documents or launching a geographic map.

If a capability is not explicitly enabled on the MDT Type, the corresponding function is not available on the device. Review your device types to ensure their platform is supported and enable functions to represent true device capabilities.

Mobile Data Cleanup

The system uses a background process to purge old RSI records as well as cleanup unused data related to Connected MCP transactions.

NOTE: Set the parameters of the **Stale MDT and RSI Cleanup** batch process to clean up this data as required by your business practices. Refer to [MI-CLMDT](#) batch control for more information.

Configuring Mobile Devices

The following section describes mobile device configuration options.

Defining MDT Types

This portal is used to maintain MDT types.

Refer to [Understanding Mobile Devices](#) for more information.

You can access the portal from the **Admin Menu > Device Management > MDT Type**.

The following zones may appear as part of the portal's **Main** tab page

- **MDT Type List.** This zone lists all MDT types. Broadcast a record to display the details of the selected deployment part.
- **MDT Type.** This zone provides information about the selected MDT type.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_MDT_TYPE](#).

Defining MDTs

This portal is used to display and maintain a mobile data terminal (MDT). You need to define a mobile data terminal (MDT) record for each mobile device that needs to communicate with the system to send and receive work using the mobile application.

You can access the portal from the **Main Menu > Device Management > MDT**. You are brought to a query portal with options for searching for a specific MDT. Once an MDT has been selected you are brought to the maintenance portal to view and maintain the selected record.

Refer to [Understanding Mobile Devices](#) for more information.

The **MDT** zone on the portal's **Main** tab page provides information about the mobile device.

The following zones may appear as part of the portal's **Log Files** tab page:

- **Mobile Log Files Search.** This zone allows you to search for log files that may have been sent in from the mobile device to a designated location on the sever.
- **Mobile Log.** This zone shows server logging information related to communications made using this mobile device, if this feature is enabled from the mobile device. The zone is only applicable to devices using the **Connected MCP** or the **RESTful Services** mobile platforms.

Replicate MDTs

You may use the **Replicate** button set up a bulk of MDT records based on the details of the current MDT. This function sets the device tag for each new MDT to the prefix text you provide and a running sequence number.

Getting the MDT Log Files

Log files are not sent to the server automatically; they are only sent on demand, usually for debugging or tracing purposes. Use the **Get Log Files** button to send a message to the device to send log files to the server. When the device sends the files, you may use the **Log Files** tab to view their content.

Changing the Mobile Device Log Level

Click the **Set Log Level** button to send a request to the device to change the logging level used by the mobile application. Once the mobile device is updated, it sends a response back to the server to update the current log level on the MDT record to reflect the change.

Reset Registration

If this MDT is used by an iOS device and the mobile application was reinstalled on the device, the unique ID has to be reset before the MDT can be used again. Click the Reset Registration button to do so.

Refer to [Registration](#) for more information.

Configuring Deployment Types

The following section describes mobile device configuration options.

Defining Deployment Parts

This portal is used to maintain deployment parts. You may upload a predefined bundle when adding or updating a deployment part.

Refer to [Understanding Deployments](#) for more information.

You can access the portal from the **Admin Menu > Device Management > Deployment Part**.

The following zones may appear as part of the portal's **Main** tab page

- **Deployment Part List.** This zone lists all deployment parts. Broadcast a record to display the details of the selected deployment part.
- **Deployment Part.** This zone provides information about the selected deployment part.
- **Deployment Items.** This zone lists the application items included in this deployment part.
- **Referenced By Deployment Types.** This zone lists the deployment types referencing this deployment part.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_DEPLOYMENT_PART](#).

Defining Deployment Types

A deployment type represents a specific type of mobile application. This portal is used to maintain deployment types.

Refer to [Understanding Deployments](#) for more information.

You can access the portal from the **Admin Menu > Device Management > Deployment Type**.

The following zones may appear as part of the portal's **Main** tab page

- **Deployment Type List.** This zone lists all deployment types. Broadcast a record to display the details of the selected deployment type.
- **Deployment Type.** This zone provides information about the selected deployment type. You may use the **New Deployment** link to submit the batch process to create a new deployment.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_DEPLOYMENT_TYPE](#).

Maintaining Deployments

This portal is used to view existing deployments and change their status.

NOTE: A deployment has to be activated before it can be downloaded to mobile devices. You can also inactivate a deployment so that it can no longer be deployed.

Refer to [Understanding Deployments](#) for more information.

You can access the portal from the **Main Menu > Device Management > Deployment**. You are brought to a query portal with options for searching for a specific deployment. Once a deployment has been selected you are brought to the maintenance portal to view and maintain the selected record.

The **Deployment** zone on the portal's **Main** tab page provides information about the mobile device.

The following zones may appear as part of the portal's **Main** tab page

- **Deployment.** This zone provides information about the selected deployment.

- **Out of Date Items.** This zone lists items that have changed and caused the deployment to become out of date. Expand the zone to request the system to evaluate the deployment now. If the deployment is out of date, it is marked as such and the zone lists the out of date items information.

Schedulers

Understanding Schedulers

The scheduling process automatically assigns tasks to crew shifts, matching field work requirements to crew capabilities and optimizing the routes traveled by crews. The goal of the scheduling process is to produce the most cost-effective schedule.

The following sections provide a high level understanding on how the scheduler works and what components requires configuration.

The scheduling Process

The scheduling process has the following basic inputs:

- **Tasks.** Tasks represent the types of "things" the crew might have to do during their shift. They can be any one of the following, all of which are incorporated into the schedule as needed.
- Activities indicate the type of work, required skills, task duration, service location, time windows, and task-based cost factors.
- If using depot functionality, the scheduler inserts visits to the depot as depot tasks.
- Periods of Unavailability tasks (POUs) indicate a fixed period of time as well as a location where the crew must to attend to some task unrelated to shift work, such as a meeting.
- Non productive tasks (NPTs) indicate a window of time after the shift has started as well as location where the crew must attend to some task unrelated to shift work.
- Breaks indicate a broad window during which the crew may take their break, as well as the allowed break duration.
- **Crew Shifts.** Shift data includes crew capabilities, covered service areas and service classes, logon and logoff location, and shift-based cost factors.
- **Contractor Capacities.** Capacity data includes covered service areas and service classes, as well as a capacity amount which may be defined in number of activities or number of hours. This is only applicable if you are using capacity contractors. Refer to [About Capacities](#) for more information.
- **Depots.** If you are using depot functionality, the scheduler can manage shifts which include the location where the crew picks up and drops off goods.
- **Map Data:** All locations within the service territory are represented in the map data.
- **Static Control Data:** This includes scheduler parameters and appointment booking groups information.

The scheduling process assigns each task a scheduled date/time on a particular shift of a crew, or a contractor capacity if using the contractor functionality. As part of the optimization process, a task may be assigned, unassigned, and reassigned any number of times until the task is dispatched.

Refer to [Scheduling An Activity](#) for more information.

Multiple Schedulers

The term scheduler refers to the system component that manages a single scheduling process. An implementation typically runs at least two schedulers to provide for backup, but may run additional sets of schedulers if required for scalability.

Refer to [Using Multiple Schedulers to Scale the Scheduling Process](#) for more information.

Each uniquely named scheduler record is associated with

- A single scheduling process it manages. Refer to [Schedules Are Continuously Running Batch Processes](#) for more information.
- A scheduler configuration that defines the scheduling parameters to be used during the scheduling process. Refer to [Understanding Scheduler Parameters](#) for more information.
- A specific geographic area called a scheduling area it covers. Refer to [Understanding Scheduler Areas](#) for more information.

Scheduler Process Types

A scheduling process may have the following responsibilities:

- Optimization
- Request Handler Only
- Optimization and Request Handler
- Holding

Optimization refers to the main responsibility of the scheduling process, i.e. to schedule tasks to shifts (or capacities).

A request handler scheduler responds to various types of ad-hoc requests, such as an appointment booking request etc. Refer to [Request Types](#) for more information.

A scheduler can be configured to handle optimizations and requests, or just one of these roles.

A holding scheduler does not schedule shifts or tasks. It is designed to capture tasks and shifts that cannot be assigned to any online scheduler. One and only one holding scheduler should exist. Any task or shift attached to a holding scheduler is evidence of a scheduling error.

Request Types

A scheduler that handles requests can handle the following request types:

- **Chooser.** Returns a list of recommended shifts or contractor capacities for an activity.
- **Appointment booking.** Returns a list of valid possible appointments for an activity. Appointment booking refers to how the system handles requests from the host system for appointments and how it responds with available appointment windows. Refer to the [Appointment Booking Groups](#) for more information.
- **Conditional booking.** Confirms the schedule of an activity when it is interfaced to the system. Refer to [Conditional Booking](#) for more information.
- **Route and Directions.** Returns the street level route as well as driving directions for an ordered list of locations.

Schedules Are Continuously Running Batch Processes

Each scheduler record must reference a unique batch control that manages and handles the execution of the scheduler and is designed to run continuously without interruption.

At run time, the batch process determines the scheduler record associated with its batch control and handles operations requested on behalf of the scheduler. If the record is marked **Online**, the batch process starts the corresponding scheduler, and if requested to be **Offline**, it instructs the scheduler to properly terminate. If necessary, a scheduler can be brought online or offline by using the **Boot** or **Shut Down** buttons, respectively, on the [Scheduler](#) portal.

As standard batch processes, scheduler batch processes may be visible by an enterprise management console as well as started and monitored by an enterprise batch scheduler.

Refer to [Monitoring Schedulers](#) for more information.

Auto-Dispatching

Dispatching is an action that locks a task's schedule so no further optimization takes place and marks it as ready to be sent to the assigned resource (crew on a specific shift or a contracting company). If it is currently assigned to a crew's shift, dispatching locks the task's sequence in the shift schedule.

Automatic dispatching refers to the process in which the scheduler initiates the dispatching of tasks.

You can enable or disable automatic dispatching for activities as follows:

- **Globally.** Use the "Enable Auto Dispatch" scheduler parameter to enable or disable auto dispatching for activities globally. If enabled, it can still be disabled at the shift or activity level.
- **By Shift.** Refer to [Controlling Auto-Dispatching](#) for more information.
- **By Activity.** Refer to [Disabling Auto Dispatch](#) for more information.

The following are additional [scheduler configuration](#) parameters that affect auto dispatch:

- Break Dispatch Mode
- POU Dispatch Mode
- Non Productive Task Dispatch Mode
- Auto Dispatch Stability Period
- Auto Dispatch Interval
- Auto Dispatch On Completion
- Auto Dispatch Time Horizon
- Emergency Dispatch Mode
-

The Scheduler Registry

The scheduler registry contains a record for each task, shift, capacity, appointment booking group, and scheduler configuration managed by a scheduler. It is via the registry mechanism that entity updates are sent to the scheduling process.

For example, when the scheduler acquires an activity, it adds it to its registry and gets its scheduling related information. If the activity is later on changed, the registry record for the activity is marked as being changed which triggers the scheduler to read the activity's latest information.

This section describes topics related to the scheduler registry.

Registry Objects

The scheduler acquires entities within its short and long scheduling horizons as follows:

- An activity is registered with all schedulers that cover its geographic location or at least its service area. A scheduler that uses optimization areas acquires activities that their location's geocode is within its optimization areas. If the scheduler's area is defined using service areas only, it registers all activities within these service areas.
- A shift is registered with all schedulers that cover any of its service areas.
- A break is registered with all schedulers that register its shift.
- A POU task is registered with all schedulers that register any shifts for the invited crews, and cover any of the POU services areas, if any.
- Every appointment booking group is registered with all schedulers.
- A scheduler configuration and its cost profile are registered with all schedulers using them.

Details about each entity are kept within the in memory data store of the scheduler. The scheduler manager batch process reads the scheduling details of a registered object and forwards the information to the scheduler.

NOTE: Logic to prepare details to be sent to the scheduler resides in a "Scheduler Read" algorithm on the entity's business object.

NOTE: Click [here](#) to see the task related algorithm types available for this system event.

NOTE: Click [here](#) to see the shift / capacity related algorithm types available for this system event.

NOTE: Click [here](#) to see the depot related algorithm types available for this system event.

NOTE: Click [here](#) to see the scheduler configuration related algorithm types available for this system event.

NOTE: Click [here](#) to see the appointment booking group related algorithm types available for this system event.

The scheduler manager batch process keeps the scheduler's registry up to date, removing entities that should no longer be managed by the scheduler, and acquiring new entities as needed.

You can configure the system to provide advanced users with the ability to search for and locate schedulers that are handling a particular task or shift. The **Navigate To Scheduler Registry(M1-NavScReg)** BPA script can be added to your user's favorites through preferences. This script locates the related scheduler registry, if any, for the task or shift being displayed and navigates to the scheduler registry portal.

Notifying The Scheduler Of Updates To Entities

The system notifies scheduler managers about a change that was made to an entity they have registered by posting a change action onto each scheduler's registry record for the entity. Each scheduler manager then communicates this information to its scheduler process.

Posting actions in the registry include:

- **Send.** An entity has changed and the entire entity needs to be sent to the scheduler.
- **Send Status.** An entity's state has changed and state-related details for the entity, rather than the entire entity, need to be sent to the scheduler.

- **Send Schedule.** Another scheduler manager has made a schedule change to an entity and schedule-related details need to be sent to the scheduler.
- **Processed.** The update conveyed via the registry posting action has been processed by the scheduler.
- **Remove.** The system has deleted or cancelled an entity (or changed it so that it is no longer relevant to the scheduler) and the entity needs to be removed from the scheduling process (that is, it should no longer be considered for scheduling). After removing the object, the scheduler deletes the registry entry.

NOTE: It is the responsibility of an audit algorithm associated with the entity's business object to post the scheduler registry update when changes are made to that entity.

Notifying The Entity Of A Schedule Change

The scheduler manager notifies the system when it automatically dispatches a task as well as when key information about a dispatched task's schedule has changed.

This allows the system to perform business rules related to dispatching as well as communicate schedule changes to crews for tasks already dispatched to them.

NOTE: Logic to process changes in scheduler post dispatch resides a "Post Dispatch" algorithm on the task business object. Click [here](#) to see the algorithm types available for this system event.

Errors

If any errors or warnings resulted from the posting action, they appear in the registry record.

In addition, the registry records for a **Holding** scheduler represent entities that should have been covered by online schedulers, but are not.

Some scheduling errors may resolve on their own when the problem is temporary, while other errors may require user intervention. The system generates To Do entries for all registry related errors in batch and closes these To Do entries when the issue is resolved.

NOTE: To Do entries for scheduling errors are created by the **Scheduler Monitor** batch process. You need to ensure the batch runs frequently enough to match your business requirements of reporting these errors. Refer to [M1-SCHED](#) batch control for more information.

Current Crew Location

The current location of the crew at any point in time is key information to the scheduling process. The scheduler estimates the crew's current location based on its last known location. GPS data is not required, but when it is available, estimating the crew's current location is more accurate.

The mobile application may be enabled to frequently send GPS reports of the crew's location. Refer to [Sending Current Location](#) for more information.

Vehicles equipped with an external GPS system may also be configured to transmit GPS data independently. Refer to [Add GPS Coordinates For Resource](#) for more information.

When determining the location of a crew working on a specific shift, the system first looks for GPS records associated with the crew and if none are found, it looks for GPS data associated with any of the vehicles the crew utilizes on that shift.

Monitoring Schedulers

monitor and troubleshoot schedulers

The application provides the ability to monitor schedulers using standard batch controls. This allows you to use third party enterprise management tools, if desired, to monitor schedulers as you would with any other batch process.

You can monitor schedulers using standard processing:

- Use an enterprise scheduler which shows if any scheduler submissions have failed. These can typically restart the scheduler using the tool's console.
- JMX monitoring is also available for enterprise management consoles such as Oracle Enterprise Manager or via simple JMX clients.
- View diagnostics in the [Batch Control Global View](#) portal. The scheduler periodically reports various statistical measurements related to its current operation such as total number of activities and shifts it manages, average cycle time, etc. You may view these statistics on the [Scheduler](#) portal. If the **Level Of Service** algorithm is enabled on the scheduler's batch control, the system uses this information to evaluate how well the scheduler currently works and rate its level of service accordingly.

Understanding Scheduler Areas

A scheduler area defines a geographic area to be scheduled by a number of cooperating schedulers, where each scheduler covers one scheduler area.

Refer to the inline help for instructions on creating and maintaining scheduler area records. The following sections provide conceptual information to help you design your scheduler areas before entering them into the system.

Multiple Service Areas within a Scheduler Area

A scheduler area may comprise one or more service areas.

A scheduler is limited to scheduling tasks within its scheduler area. Thus, the set of service areas comprising a scheduler area must at least match the most complex set of service areas covered by any given shift.

Consider a territory with eight service areas: N, E, S, W, NE, SE, NW, SW. Shifts scheduled during the day typically operate in just two service areas, while shifts scheduled during the night can be routed to any of the eight service areas:

- Day Shift (N): N, NE
- Day Shift (E): E, SE
- Day Shift (S): S, SW
- Day Shift (W): W, NW
- Night Shift: N, E, S, W, NE, SE, NW, SW

Since the scheduler area must cover the most complex set of shift service areas, the Night shift is used to model the scheduler area: N, E, S, W, NE, SE, NW, SW.

Using Optimization Areas

If a scheduler area has too many tasks for a single scheduler to manage (typically over 20,000), the system should be configured to run multiple schedulers in parallel and each scheduler should handle a segment of the total scheduling area. The segments, called optimization areas, should overlap each other. The overlap promotes task/shift migration between schedulers and thus achieves optimal shift routing.

Determining the Best Optimization Area Shape

Every route that crosses an optimization area boundary represents additional workload for the scheduler. The scheduler must acquire all tasks of every shift it schedules so it is most efficient if every scheduler has most of its shifts wholly within

its own optimization area. Thus, the best shape for an optimization area is the one least likely to have routes exiting and entering the area.

Optimization area shapes must be complementary. Optimization areas of different sets should not share common boundaries and should overlap as much as possible to promote optimum routing.

The diagram below illustrates how a pair of complementary optimization areas may be insufficient for optimal routing at specific points on the map. This simple diagram shows two optimization sets, each with two optimization areas. Each of the four optimization areas shown has its own scheduler - for simplicity, only partial sets are illustrated. The black and orange lines represent four different shifts.

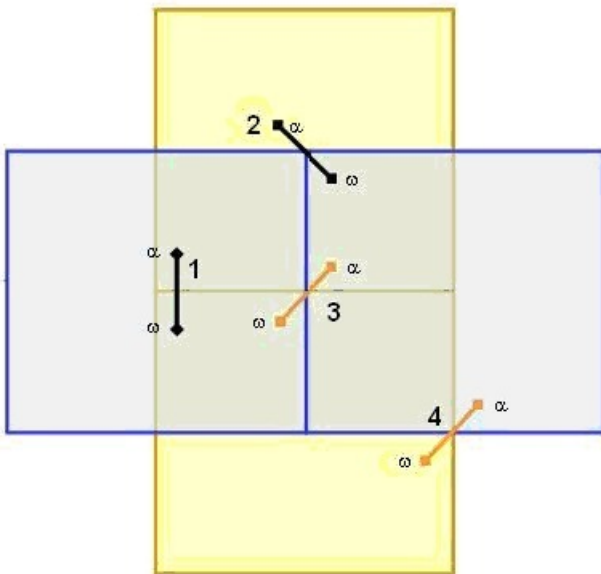


Figure 12: Inefficient Optimal Routing

Shifts 1 and 2 can be adequately covered in this configuration, because each one lies entirely within at least one scheduler. Shifts 3 and 4, however, cannot, because no single scheduler contains all the tasks of the shift. This is referred to as the corner problem, which occurs wherever the boundaries of two different optimization areas intersect and the two areas belong to different optimization sets.

Adding a third optimization set, as shown in the following diagram, solves the corner problem. All shifts can be covered as shown, because each one lies entirely within at least one scheduler.

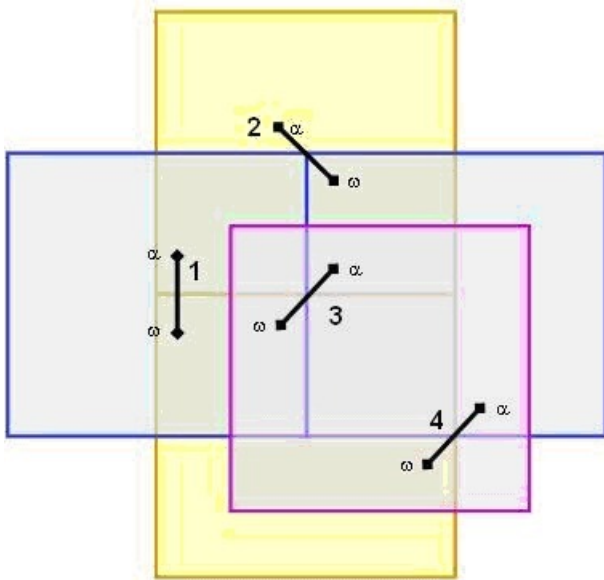


Figure 13: Efficient Optimal Routing

In general, corner problems caused by optimization area overlap must be covered by a third set of optimization areas. If no corner problems exist, only two optimization sets are required.

Follow these general steps when designing optimization sets:

- Design a set of optimization areas to cover the scheduler area, where each optimization area is as large as possible (the number of tasks bounded by any area should not exceed the Scheduler Areas maximum scheduling load plus a growth factor) but still fits into a single scheduler. For best results, always try to use the least number of optimization areas possible and assign only one optimization area to a scheduler (though multiple optimization areas can be assigned to a single scheduler).
- Design a second set of optimization areas where the optimization area boundaries are as far as possible from the first set's optimization area boundaries.
- Design a third set of optimization areas to cover any 'corner problems' caused by the overlap of the first two sets.

Example

The following image illustrates a scheduler area, named 'City', that contains more tasks than the physical limit for a single scheduler. Therefore, the scheduler area will require three sets of optimization areas.

City

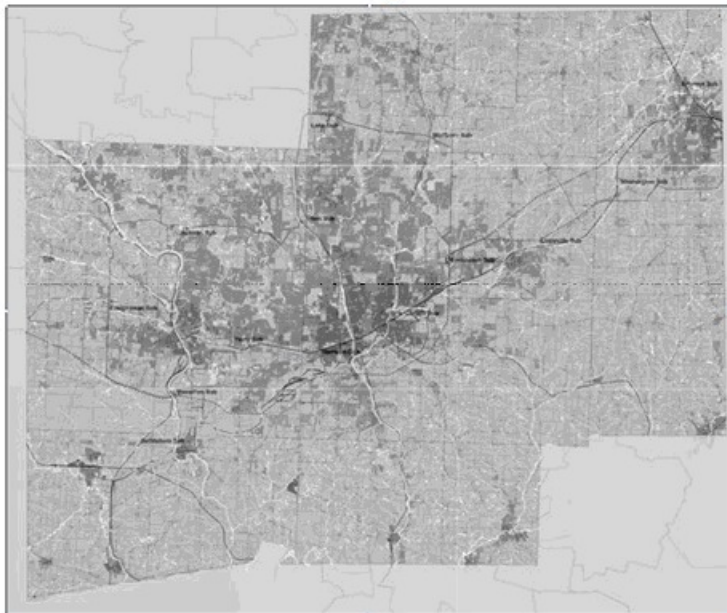


Figure 14: City Geographic Area

Define the first optimization set with coordinates: (40.9, -75.1 NW) to (40.2, -74.0 SE). The optimization set will have two optimization areas:

- Optimization area 1: (40.9, -75.1 NW) to (40.6, -74.0 SE).
- Optimization area 2: (40.6, -75.1 NW) to (40.2, -74.0 SE).



Figure 15: Optimization Set 1

Define the second optimization set with coordinates: (40.9, -75.1 NW) to (40.2, -74.0 SE). The optimization set will have two optimization areas:

- Optimization area 1: (40.9, -75.1 NW) to (40.2, -74.5 SE).

- Optimization area 2: (40.9, -74.5 NW) to (40.2, -74.0 SE).

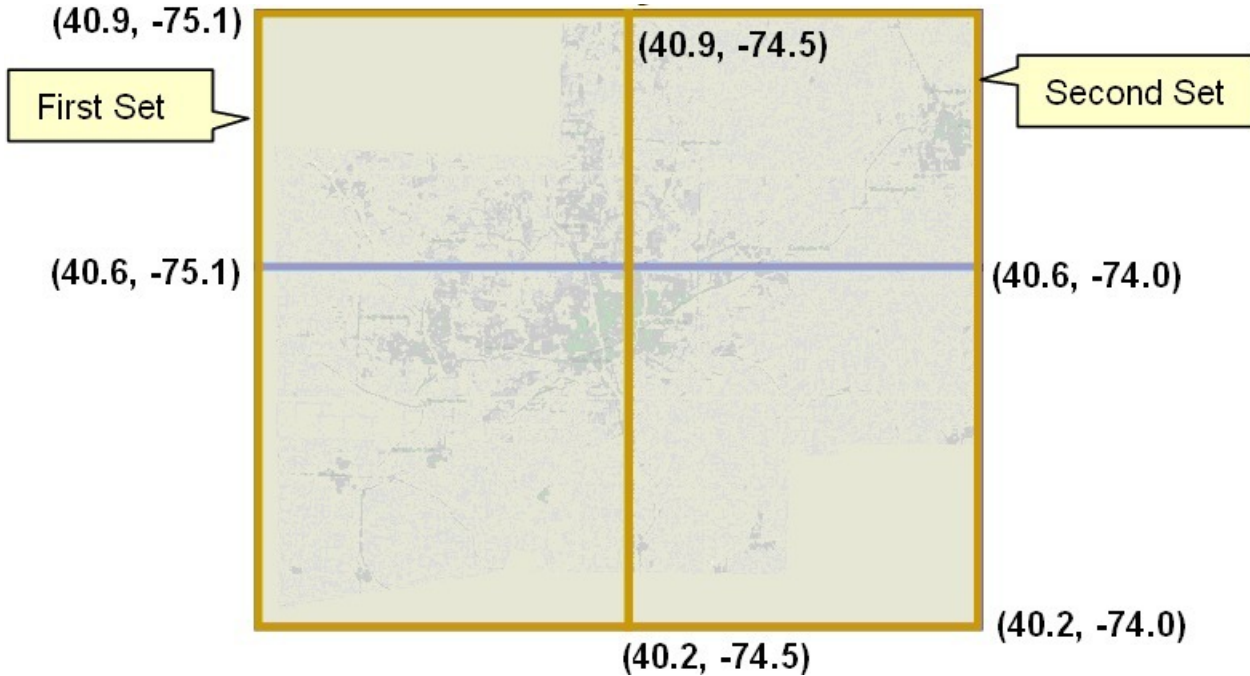


Figure 16: Optimization Set 2

Define the third optimization set with coordinates: (40.7, -74.7 NW) to (40.5, -74.4 SE). The optimization set will have one optimization area:

- Optimization area 1: (40.7, -74.7 NW) to (40.5, -74.4 SE).

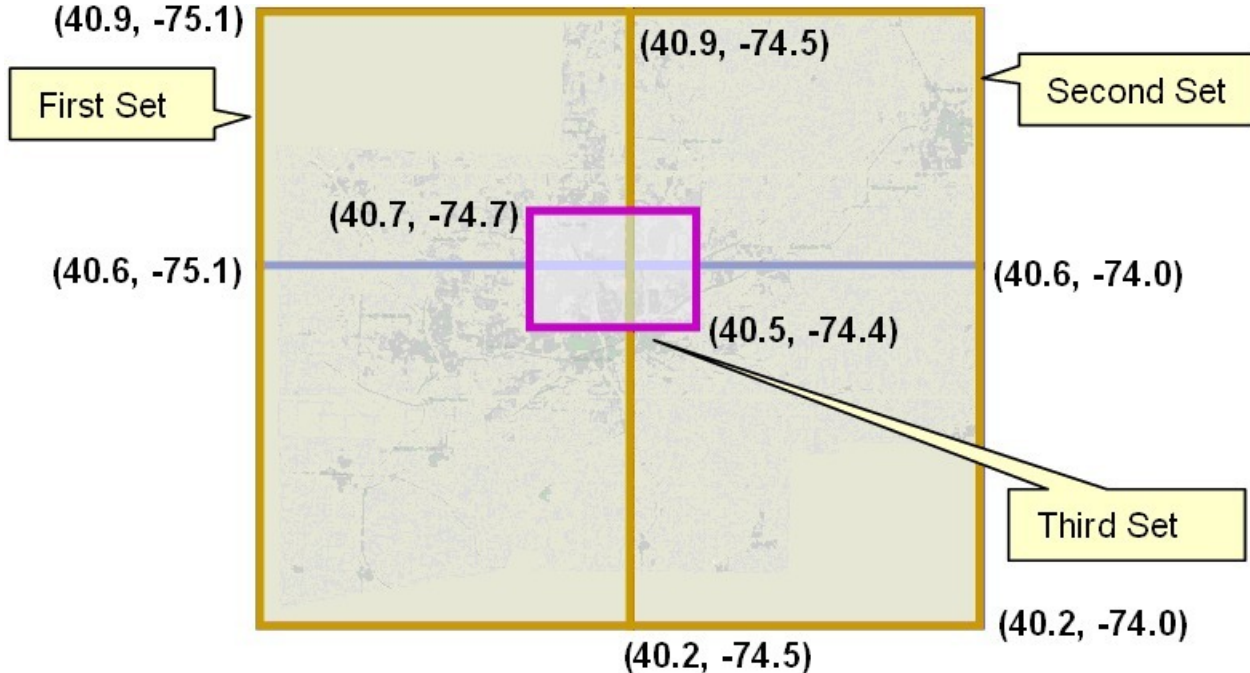


Figure 17: Optimization Set 3

Scheduler Map Files

A scheduler uses a map file that consists of geographic locations, referred to as nodes, and inter-node links for which the direction, distance, and travel time are known. The MapEditor tool allows you to create map files used by the scheduler to obtain travel information and to display places and travel paths in a particular geographical region.

When new map updates are sent from HERE maps (formerly Nokia / NAVTEQ), they can sometimes overlay the existing maps and overwrite any edits made by your implementation via the Map Editor tool. If your system is properly configured, edited maps can be reapplied onto an updated version of HERE maps.

Documentation for this tool is provided in a separate guide. Refer to the Oracle MapEditor User Guide.

Understanding Scheduler Parameters

The goal of the scheduling process is to produce the most cost-effective schedule or the most efficient schedule, while considering a large number of configurable factors to arrive at the optimum schedule.

The scheduler is heavily parameterized with the ability to support a myriad of operational use cases and scenarios. Every individual implementation will utilize a unique combination of these parameters to deliver the final operational solution.

Scheduler parameters control how a particular scheduler will schedule tasks, optimize routes, and dispatch activities to crews to achieve this goal. Descriptions of the scheduler parameters are provided in [Parameter Definition](#) portal. In general these parameters should only be modified with the guidance of your Oracle representative.

This section describes in a high level the various types of parameters used by the scheduler.

Parameter Types

Scheduler parameters can be broadly categorized as follows:

- **Cost Control.** These parameters provide detailed scheduling goals and policies. They control the day today flexibility and "tuning" of the scheduler solution.
- **Entity.** These parameters are defined as part of other entities in the system, such as shifts or activity types, and serve as factors over corresponding cost control parameters.
- All other types of parameters control general operating and administrative behavior of the scheduler.

Values for all but entity parameters are associated with a [scheduler configuration](#) record that may be shared by multiple schedulers.

Refer to the [Parameter Definition](#) portal for the complete list of scheduler parameters, their description and default values. .

Scheduler Configurations

Each scheduler is associated with a scheduler configuration. A scheduler configuration is a predefined set of parameters and settings that control scheduler behavior; a scheduler configuration may be used by multiple schedulers.

Scheduler configuration settings differ for each implementation depending upon the type of key operations managed by an organization.

This section describes topics related to scheduler configurations.

Scheduling Horizon

The scheduler uses a scheduling horizon to determine the number of days into the future to consider when scheduling work.

Appointments set outside of the scheduling horizon are considered by the scheduler as "long term" appointments. A scheduler configured to handle requests may be restricted to handle short term requests only (i.e. those set within the scheduling horizon), long term requests only, or both.

The scheduler uses a long term horizon to determine the number of days into the future to consider for long term requests. The long term horizon must be the same or larger than the scheduling horizon.

A scheduler configured to handle requests can also be restricted to handle short term requests only (i.e. those set within the scheduling horizon), long term requests only, or both.

Plan Consistency

To ensure consistency in decision making between schedulers, all schedulers that their scheduler areas cover the same service area must use the same scheduler configuration. If they manage the same entities, they must share the same scheduling goals and restrictions.

Cost Profiles

A cost profile is a type of scheduler configuration that establishes a pre-defined set of cost control parameters which can be referenced on multiple scheduler configurations.

Refer to [Cost Control Parameters](#) for more information.

Cost profiles can be defined to handle different situations, users can actively switch between cost profiles without shutting down the scheduler. Example situations which may cause you to switch cost profiles include:

- Time of year - winter work, summer work, etc.
- Strong thunderstorms - where different types of work may be preferred
- Sudden drop in temperature - where regulations prohibit turning off power to customers for non-pay
- Sudden high temperatures - where regulations prohibit turning off water to customers for non-pay
- A large backlog of certain types of work, where it has been determined it is ok to work more overtime
- and so on.

Switching cost profiles is done by changing the cost profile referenced on the scheduler configuration. When the cost profile is changed, the scheduler responds immediately by optimizing work according to the changed cost profile. The same applies to changing the parameters defined on a cost profile that is currently used by online schedulers; the schedulers are notified and acquire the new cost profile values automatically.

NOTE: There is no need to restart the schedulers to reflect cost control changes.

Working With Scheduler Configurations

Only complete configurations can be associated with online schedulers. If you are setting up a new scheduler configuration, mark it complete before you reference it on scheduler records.

Changing cost profile parameters or switching a scheduler configuration to use another cost profile takes effect automatically without the need to stop and restart the scheduler.

For all other scheduler configuration changes, you would need to stop and restart the schedulers for the changes to take effect.

Cost Control Parameters

Applying cost controls allows the system to produce the most cost-effective schedule while adhering to your business rules and processes. A scheduler is associated with cost control parameters via the cost profile referenced on its scheduler configuration.

Refer to [Business Drivers Related to Cost Control Settings](#) for high level guidelines on how cost control parameters are used to achieved different business goals.

This section discusses cost control and entity parameters which can be modified to best suit your business practices when configuring the system.

Cost Parameter Types

Cost parameters are of two types.

Relative

Cost factors that work as multipliers against global costs. For example, consider that an organization has fulltime crews and contractors that may be called on to handle excess work. In order to encourage the system to schedule work to contractors only when no full-time crew is available, you could apply a relative cost of 10 to contractor shifts, indicating that these shifts would cost ten times as much to use in the schedule as a full-time crew shift.

Examples of relative cost parameters are Relative Shift Cost, Relative Late Cost. Relative costs factors default to a value of 1.0.

Global

Costs that are applied globally at a base level within the schedule.

Examples of global cost settings are Late Cost, Shift Cost, Travel Time Cost, and Travel Distance Cost.

Global Cost Parameter Types

Global costs can be of various types.

Flat

Costs that are applied once when the object is activated, and removed when the object is no longer being used. These are also known as "Standing Costs". The cost for using this object does not go up over time but stays at a standard or flat cost.

For example, the global Shift Cost is the flat cost of utilizing a shift. The Relative Shift Cost parameter can be used as a multiplier against the flat cost. Other examples of flat costs include: Shift cost, cost of a run, cost of goods, hauling costs, site costs, etc.

Variable

Costs that increase linearly based on a measurable variable. These are also known as "Running Costs" or "Variable Costs". For example, travel time costs may be expressed as dollars per minute or dollars per hour, and travel distance costs expressed as dollars per mile or dollars per kilometer. Other examples of rate costs include: idle time costs, access window costs, pick up or drop off services, allocation costs, site separation costs, product depot costs, etc.

Complex

Costs that increase exponentially and are nonlinear in nature. The Late Cost parameter, which defines the global cost of late arrival to an activity, is a function of the number of seconds late for arrival. For example, late cost is incurred beginning with the first second the crew is late for arrival. However, the exponential function can be defined so that the late cost is insignificant for the first ten minutes, but starts to ramp up quickly thereafter. This cost pattern models the idea that the customer will accept a crew being 5-10 minutes late, but will be increasingly displeased with any further delays.

Examples of complex costs include: resource arriving late at a task, overtime, volume, weight, reserve capacity, etc.

The following figure illustrates the type of curve represented by a complex cost:

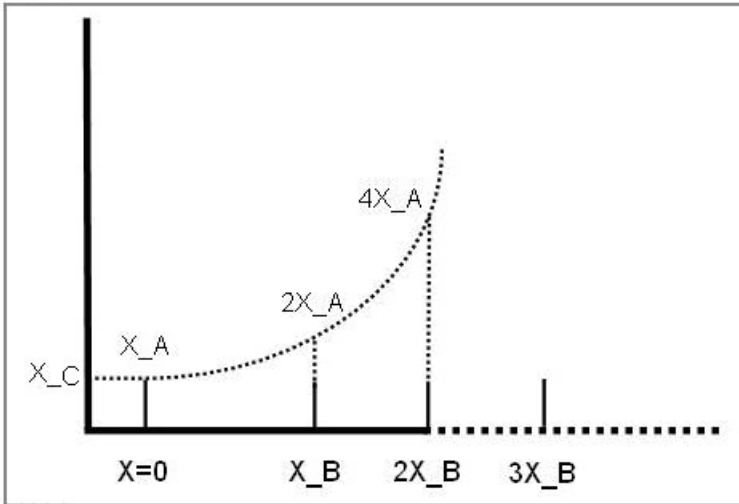


Figure 18: Complex Cost Curve

Complex costs are determined by mathematical functions controlled by a group of three cost control parameters:

- **Cost Value A** represents the initial slope for $x \geq 0$ (at $x=0$).
- **Cost Value B** represents the doubling rate (the slope of the function doubles every x_B).
- **Cost Value C** represents a flat additional cost, creating an initial step.

Time-Dependent

Factors that adjust shift-based costs based on when a shift begins. For example, when the system is scheduling activities several days into the future, the goal is to combine activities to produce the best route for each day. As a result, some shifts may appear under-utilized or even empty. However, when scheduling activities for the current day, the main objective is to add as many activities as possible to fully utilize the crew's time. Time-dependent cost (TDC) factors allow the scheduler to produce the most cost-effective schedule considering these time factors.

To configure time-dependent costs, you must define three distinct curves.

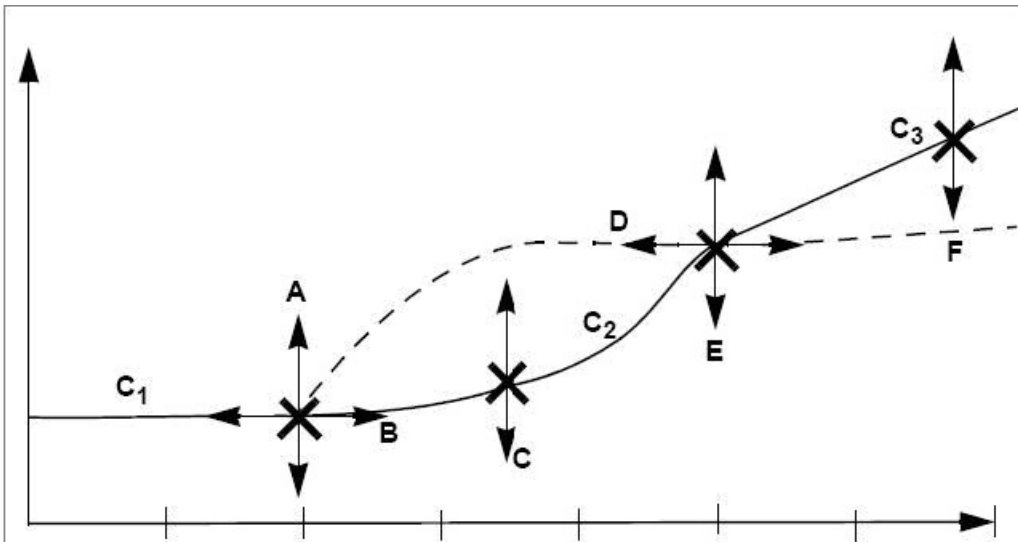


Figure 19: Time Dependent Curve

In the diagram above, the horizontal axis is time, in day increments (Day 0, Day 1, Day 2, etc.). The initial curve is always flat. The second curve is either convex or concave. The final curve is always a straight line into infinity.

Six independent variables (TDC Values A-F) are required to define this shape:

- **TDC Value A.** Initial value for the time-dependent cost factor, which is flat from day 0 to TDC Value B. This value must be greater than or equal to zero.
- **TDC Value B.** Controls how long the initial TDC Value A applies, in hours. For example, if the initial factor should apply for all of the first day, then this should be set to 24 (hours).
- **TDC Value C.** Controls the shape of the curve between the initial value and the final slope, referred to as the transition curve. The valid range is 0.1 to 10. A value of 4 produces a curve similar to the solid line in the previous diagram. A value of 0.2 produces a curve similar to the dashed line in the previous diagram. A value of 1 produces a straight, linear gradient between the initial and final slope, and is a good starting point.
- **TDC Value D.** Number of hours after which the function changes from curved to linear. This value controls the width of the transition curve. This value must be greater than TDC Value B.
- **TDC Value E.** Factor at the point at which the function changes from curved to linear. (This the final factor for the transition curve and the initial factor for the final curve.) This value must be greater than or equal to zero.
- **TDC Value F.** Controls the gradient of the final slope (units/hour).

Thus, the three curves are defined as follows:

- **Initial curve:** $C1 (x:0..B) = A$
- **Second curve:** $C2 (x:B..D) = ((x - B) / (D - B))C * (E - A) + A$
- **Final curve:** $C3 (x:D..) = (x - D) * F + E$

Example 1: TDC Applied to Late Cost

Time-dependent costs can be applied to the **Late Cost** parameter to address a situation where cost should remain the same for the first two days, then rise rapidly 1 cost unit per day over the next two days and then increase by 0.2 units for all the remaining days. In this example, TDC variables would be set as follows:

$$A = 1$$

$$B = 48 \text{ (hours)}$$

$$C = 1$$

$$D = 48 \text{ (hours)}$$

$$E = 3$$

$$F = 0.0083 \text{ (} 0.2 / 24 \text{)}$$

Assuming that 5 minutes of late time is costed at 100 units to begin with, applying these time dependent costs would result in the following:

For a shift on day 1, the cost is 100

For a shift on day 2, the cost is 100

For a shift on day 3, the cost is 200

For a shift on day 4, the cost is 300

For a shift on day 5, the cost is 320

For a shift on day 6, the cost is 340

For a shift on day 7, the cost is 360

Example 2: TDC Applied to Shift Promotion Cost

Time-dependent costs can be applied to the **Shift Promotion Cost** parameter to promote early completion of activities. The scheduler has no innate preference for activities to be completed as early as possible. Until other constraints are breached, the scheduler is mainly distance driven.

Thus, it may schedule an activity for day 5, even though there is capacity on day 1, if it results in less travel. Time-dependent costing can be applied to the Shift Promotion Cost to bias against distance in favor of ensuring that no transport capacity is wasted for day 1. Ideally, costing is set so that day 1 fills up just before it becomes day 0. If costing is too weak, capacity is wasted; if too strong, days fill up well in advance, and the quality of routes suffers.

Example 3: TDC Applied to Reserve Capacity Cost

Time-dependent costs can also be applied to the **Reserve Capacity Cost** parameter to limit the area covered for individual routes when providing appointments. This is done by invoking an additional cost when a vehicle exceeds a certain percentage (such as 50%) of its available time.

Thus, the first appointment determines the general direction of that day's run, with the distance travelled for subsequent appointments being restricted through the reserve capacity cost. The more distance is being covered, the stronger the restriction, causing activities that introduce excessive travel to be deferred to another day. Time-dependent costing is required to gradually relax the Reserve Capacity Cost, enabling the system to fill day 0, day 1 and so on, in case the reserve capacity cost turns out to be too restrictive.

Understanding Speed Profiles

In regions where traffic issues have a significant impact on crew travel times and schedules, it is beneficial for the scheduler to be able to factor in these traffic patterns when scheduling shifts and calculating travel.

Your implementation can set up speed profile templates that define specific days and times where travel speeds are impacted. For example, the speed profile for weekday rush hour would indicate slower speeds than the profile for a week day at mid day or on the weekends.

This section describes topics related to speed profile configuration.

Speed Profile Is Made Of Daily Templates

Speed profiles define a collection of daily templates which delineate 24 hour speed factors that are associated with a type of day (default, rush hour on a weekday, holiday, weekend, and so on). Note that these are indications of relative speeds, not time. This information improves the scheduler's travel time estimates by taking into account traffic patterns based on day of the week and time of day.

Once the speed profiles are defined, you can draw the related geographic areas to show the specific regions where the speed profiles apply.

Speed Profile Area

Speed profile geographic areas represent the specific region on the map that is impacted by a speed profile.

Configuring these geographic areas allows the scheduler to calculate more accurate travel times by adjusting the average travel speeds within the map area. The scheduler factors in the speed areas that overlap the scheduler area.

The speed profile does not impact which road or route the scheduler selects or adjust the travel speeds in real time based on the geographic area of the destination. Rather, it allows the scheduler to make a more precise travel time calculation with the expected traffic conditions taken into account.

Speed Profile Overrides

On any speed profile geographic area, you have the option to override the speed profile on a specific date. For example, if you know that there is a major event which will slow down traffic during a time frame which would normally be considered off hours, you can apply an override and either enter the speed factors manually or use a daily template to define the override.

Designing Your Schedulers

This section provides high level guidelines and recommendation around scheduler administration.

Using Multiple Schedulers to Scale the Scheduling Process

The scheduling process is very complex and has high memory requirements that increase dramatically as the number of tasks being scheduled increases. Memory requirements also increase as the scheduling horizon (the number of days into the future that must be scheduled) increases.

To achieve optimum performance without sacrificing scheduling efficiency, the system can be configured to use multiple schedulers at once and limit the number of tasks and shifts for which each scheduler is responsible.

Oracle provides general guidelines for calculating the optimum number of schedulers required for a particular installation:

- Define a variable, **L**, to be the maximum task load that can be scheduled by a single scheduler. 20,000 is the typical maximum.
- Determine the total number of tasks, of unique location, to be scheduled in a given scheduler area within the scheduling horizon.
- If the task count is less than L, then a single scheduler can cover the entire area. One extra scheduler will be required for backup.
- If the task count is greater than L, then the scheduler area must be covered by multiple schedulers working in parallel, and the area must be divided into smaller segments called optimization areas. Each scheduler is typically configured with a single optimization area.
- To calculate the number of optimization areas (schedulers) required for a large task count, use the following guidelines:
- Though the typical maximum load (L) for a scheduler is about 20,000, schedulers configured with an optimization area should be designed for a smaller average load, typically 62.5% of L, which we call *oL*.
- The total number of schedulers required for the scheduler area can be estimated as follows, where *n* is equal to the expected maximum task count: $((n/oL \text{ rounded up to the next whole number}) * 3) - 1$

The result is multiplied by 3 because three sets of optimization areas are typically required to manage a scheduler area with a large task count. Subtract 1 because the third optimization set can typically have fewer optimization areas than the first two sets.

Please note that these rules are guidelines only; greater or fewer schedulers may be required depending on local geographies and particular scheduling requirements.

The table below shows an approximate number of schedulers required for various task counts, assuming the following base values:

Maximum task count for a single scheduler: $L = 20,000$

Maximum task count for a single scheduler linked to an optimization area: $oL = 12,500$ (62.5% of L)

$n = \text{Task Count}$

Task Count (n) Low	Task Count (n) High	Scheduler Count	Explanation
0	20,000	2	If the load is less than the maximum task count, then one scheduler can handle the entire load, with an additional scheduler for backup
20,000	25,000	5	n = 25,000 n/oL (25,000 / 12,500) rounded up = 2 $2 * 3 - 1 = 5$
25,000	37,000	8	n = 37,500 n/oL rounded up = 3 $3 * 3 - 1 = 8$
37,500	50,000	11	n = 50,000 n/oL rounded up = 4 $4 * 3 - 1 = 11$

Thus, every physical location of an area must be covered by at least two schedulers for small task counts (typically under 20,000) or three schedulers for large task counts (typically over 20,000).

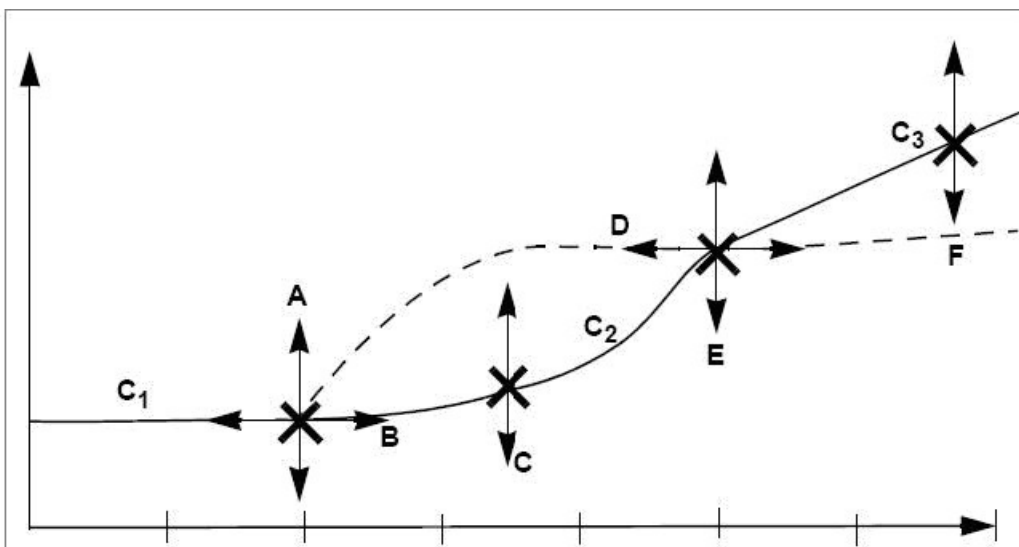
NOTE: The actual number of schedulers necessary may vary due to geographic constraints or particular scheduling requirements. Some geographic areas may be too large for a single scheduler to process. Determining the appropriate number of schedulers and defining optimum scheduling areas are essential to achieve maximum scheduling efficiency.

Failover and Load Balancing

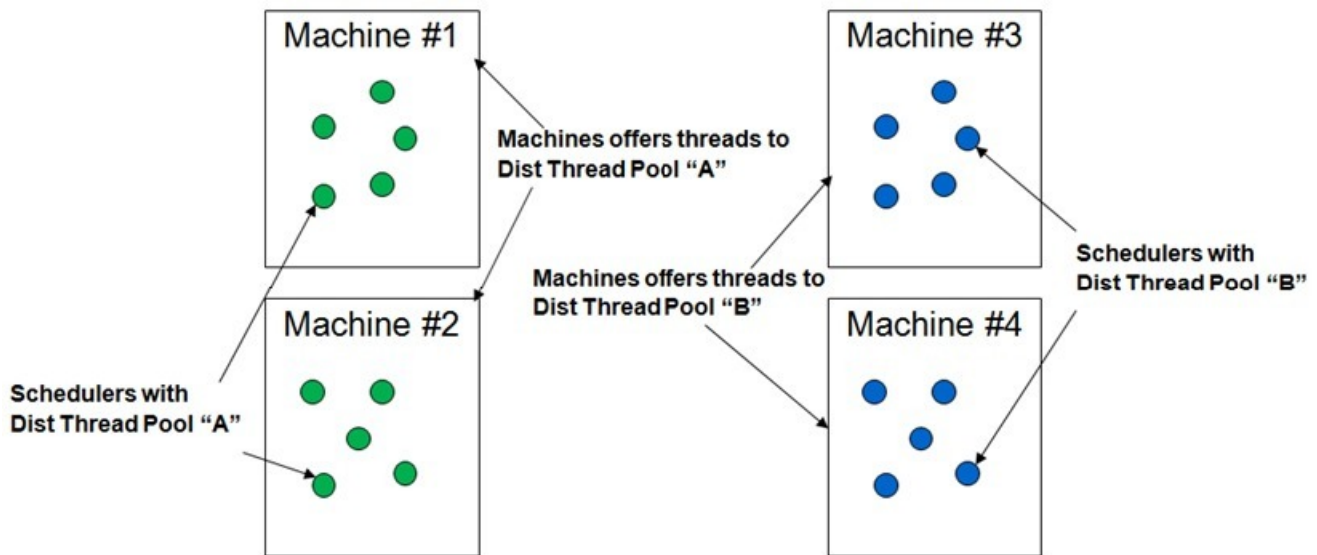
Each scheduler is controlled by a designated batch process. The batch infrastructure provides flexibility around failover and load balancing. Distributed Thread Pools allow a pool of schedulers to be associated to a pool of threads across multiple JVMs on multiple machines.

Load balancing among batch nodes assigns a thread to a JVM with the lowest percentage thread utilization for the requested thread pool. This enables active load balancing where schedulers are spread out among servers.

The following diagram illustrates scheduler distribution between thread pools and machines:



The following diagram illustrates how machine failure causes automatic migration to other machines offering threads to the same thread pool:



Business Drivers Related to Cost Control Settings

Typically, two major business drivers will guide your configuration of cost control parameters. These business drivers are generally tied to the areas of field service vs. transportation operations. Of course, any implementation may mix and match these business goals.

NOTE: It is best not to change the Travel Distance and Travel Time costs from their default settings. These act as a reference for all other cost-controls.

Field Service Based Operations

Businesses with field service based operations typically require that a person in the field make a visit to the customer site. Electric, gas, and water utility companies usually fall into this categorization.

Organizations with operations based mainly in the field might focus on the following considerations:

- More responsive to customer requests
- More assurance of timely / complete response

SLA Adherence

Operations that have service level contracts to customers, generally known as service level agreements (SLA), must adhere to certain standards when making service calls. Failure to meet these standards can result in costly penalties and legal consequences.

Cost controls to configure include any lateness and time cost rules such as:

- Arrival Costs
- Late Cost
- Relative Late Cost (Entity Parameter)
- Resource Attribute Cost
- SLA
- SLA Flexibility (Entity Parameter)

- Preferred Time Factor (Entity Parameter)

Days to Serve

Organizations that do not have formal customer service level contracts but that maintain internal targets to offer service within a certain number of days of a customer placing a work order, offer incentives to meet these service goals.

Cost controls to configure include any time dependent cost rules, resource, shift, travel, time and shift promotion rules such as:

- Late Cost
- Shift Promotion Cost
- Preferred Time Factor
- Complex Activity Span Cost

Balance Planned and Reactive Work Orders

Some scheduler rules can help to balance situations where pre-existing planned work (such as preventative maintenance) must be superseded by higher priority reactive work order types (emergency or reconnects or credit restores). The planned work must still be completed to meet service level commitments, while the organization completes the reactive work tasks.

When planned work orders come into the system, they usually come in bulk. Reactive work only comes when it occurs, so the scheduler needs to be automated to factor in reactive work. It moves planned work out to make space for reactive work while also factoring in the cost for these changes.

Cost controls to configure include any priority allocation settings, cost rules and time window automatic extension settings, such as:

- Shift Promotion Cost
- Stop Promotion Factor
- Preferred Time Factor
- Late Cost

Lowest Resource Schedule

In organizations where the efficiency of the operation is measured primarily on the utilization of the fewest number of available resources, shift cost and resource rules become very important. When the scheduler is set to consider the lowest number of resources, it attempts to use fewer crews and better resources (such as varying skill levels).

Cost controls to configure include any resource (including contractors), overtime or shift cost rules, such as:

- Resource Attribute Cost
- Overtime Cost
- Shift Cost
- Cost for a Non-Preferred Contractor. This mainly applies for shift based contractors. Organizations employing a number of sub-contractors with different agreements may want the scheduler to prioritize one sub-contractor over another, depending on the activity. A ranked list of preferred contractors is specified on the activity; any contractor company not specified is deemed to be ranked last.
- Cost for a Non-Preferred Shift Class. This only applies for organizations employing capacity based contractors. The same considerations apply as for shift based contractors (as described above); however, organizations employing both types of contractors must also configure a "Non Preferred Shift Class" cost.

Complex activities require additional configuration to control the elapsed time between the first and last visit, and the duration of visits themselves.

- Complex Activity Span Cost
- Under Scheduled Visit Cost
- Maximum Overlap Between Visits

Transportation Based Operations

Business focused on transportation based operations tend to place a high value on making appointment times and minimizing traveling distances.

Organizations with operations mainly based around transportation might focus on the following considerations:

- Better Leverage of workforce skills
- Reduction of time lags / travel waste

Appointment Compliance

In organizations where appointments are offered to ensure that deliveries and collections are made according to the appointment commitments, lateness time and distance become important factors in the scheduler configuration.

Cost controls to configure include any lateness, time or distance cost rules, such as:

- Late Cost

Maximum Density of Appointments

Most organizations will need to configure the scheduler to ensure that appointments are not scheduled too close together and that there is enough time to complete work in between appointments.

Cost controls to configure include any time or distance cost rules, such as:

- Late Cost
- Apply Minimum Travel Time to All Tasks (Optimizer Behavior Parameter)
- Task Grouping Travel Time (Optimizer Performance)
- Shift Promotion Cost (to promote appointments/fill gaps in earlier shifts)

Zero Over Capacity Schedule

In situations where the work load puts resources over capacity, parameters must be in place to enforce alternate scheduling policies and contingent plans to reallocate priorities.

Cost controls to configure include any weight, volume, capacity, lateness or overtime cost rules, such as:

- Cumulative Capacity 1- 5 Cost
- Depot SLA Window
- Depot Late
- Depot Window Capacity 1 - 5
- Late Cost
- Overtime Cost

Lowest Mileage Schedule

Some organizations place the most value on achieving the highest efficiency in their operations by measuring the economy of scheduled routes. The scheduler should be configured to plot the shortest route possible and within the shortest amount of time.

Cost controls to configure include any time or distance cost rules, such as:

- Late Cost
- Task Grouping Travel Time (Optimizer Performance)

Lowest Resource Schedule

Some organizations place the most value on achieving the highest efficiency in their operations by measuring the utilization of resources. The scheduler should be configured to use the fewest number and least expensive resources.

Cost controls to configure include any resource, shift or overtime cost rules, such as:

- Resource Attribute Cost
- Shift Costs. Any parameters related to maximizing shift efficiency.
- Overtime Cost
- Service Class Cost
- Shift Promotion Cost

Configuring Schedulers

This section provides a high-level summary of the steps required to set up schedulers and scheduling entities for your organization.

- Use the MapEditor to build map files used by the scheduler to create and optimize routes, and calculate travel times and distances. Refer to the MapEditor User's Guide for details.
- Analyze the geographic areas requiring scheduling. Refer to [Understanding Scheduler Areas](#) for more information.
- Review the scheduler parameter definitions and adjust the global default values as necessary.
- Define various cost profiles and scheduler configurations to support various circumstances as needed.
- Identify each scheduler you will be using.
- Analyze scheduler distribution over servers, as well as the necessary thread pools you would need to support load balancing and failover.
- Set up an appropriate batch control record for each scheduler and schedule it to run continuously.
- Create a scheduler record for each scheduler and associate it with its batch control, a scheduler configuration, a scheduler area, and an optimization area as needed.

Configuring Scheduler Options

Follow the steps described in this section to configure scheduler options.

Defining Parameter Definitions

This portal allows you to adjust the minimum, maximum, and default values for each scheduler parameter as needed.

Refer to [Understanding Scheduler Parameters](#) for more information.

You can access this portal from the **Admin Menu > Scheduler > Parameter Definition**.

The following zones may appear as part of the portal's **Main** tab page

- **Parameter Definition List.** This zone lists all parameter definitions. Broadcast a record to display the details of the selected parameter definition.
- **Parameter Definition.** This zone provides information about the selected parameter definition.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_PARM_DEF](#).

Defining Scheduler Areas

This portal is used to maintain scheduler areas. A scheduler area defines a geographic areas to be scheduled by one or more cooperating schedulers.

Refer to [Understanding Scheduler Areas](#) for more information.

You can access this portal from the **Admin Menu > Scheduler > Scheduler Area**.

The following zones may appear as part of the portal's **Main** tab page

- **Scheduler AreaList.** This zone lists all scheduler areas. Broadcast a record to display the details of the selected record.
- **Scheduler Area.** This zone provides information about the selected scheduler area.
- **Scheduler Area Tree.** This zone displays a tree view the service areas included in the scheduler area.

The **Scheduler Area Map** zone on the portal's **Map** tab page displays the territory covered by the scheduler area as well as all the optimization sets if any are associated with area.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_SCHED_AREA](#).

Defining Scheduler Configurations

This portal is used to maintain scheduler configurations as well as cost profiles.

Refer to [Scheduler Configurations](#) for more information.

You can access this portal from the **Admin Menu > Scheduler > Scheduler Configuration**.

The following zones may appear as part of the portal's **Main** tab page

- **Scheduler Configuration List.** This zone lists all scheduler configurations. Broadcast a record to display the details of the selected record.
- **Scheduler Configuration.** This zone provides information about the selected scheduler configuration.
- **Schedulers Using This Configuration.** This zone lists all schedulers using this scheduler configuration.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_SCHED_CONFIG](#).

Cost Control Changes

If you need to change cost control parameters for an online scheduler you may do so in either of these ways:

- Directly change the parameters on the cost profile being used
- Switch the cost profile currently used on the scheduler's scheduler configuration with another cost profile.

NOTE: There is no need to restart the schedulers to reflect cost control changes. All impacted schedulers are notified of the change and acquire the new cost control values automatically.

Scheduler Configuration Changes

If you need to change scheduler configuration parameters for an online scheduler you may do so in either of these ways:

- Directly change the parameters on the scheduler configuration being used. Next, shutdown and boot all online schedulers for the changes to take effect.
- If you wish to keep the current configuration intact, create and adjust a copy of it and publish the new configuration to all schedulers using the original scheduler configuration record, such that they use the new configuration instead of the original. Use these steps to publish a scheduler configuration:
 - Make a copy of the configuration you want to replace using the **Duplicate** action. When prompted, enter a name for the new configuration.
 - Enter your changes to the new configuration, save and complete it.
 - Broadcast the original configuration (the one you are replacing).
 - Shutdown any online scheduler using the original configuration.
 - Broadcast the new configuration.
 - Click **Publish**. When prompted, select the original configuration you want to replace. The new configuration will now be linked to every offline scheduler the old scheduler configuration was previously linked to.
- Boot the schedulers back online.

Setting Up Scheduler Batch Controls

For each scheduler, create a new batch control by duplicating the **Scheduler Manager Batch Job (M1-SM)** sample batch control. Refer to [M1-SM](#) batch control for more information. This batch control should be referenced on the corresponding scheduler record.

Scheduler manager batch processes must be run continuously for the schedulers to work properly. As standard batch processes, scheduler batch processes may be visible by an enterprise management console as well as started and monitored by an enterprise batch scheduler.

If your organization does not use a third party software to trigger and monitor batches processes periodically, you may configure these batch controls to be timed with a 60 seconds interval.

You may also want configure these batch controls to run on designated thread pool workers for load balancing purposes. Refer to [Failover and Load Balancing](#) for more information.

Defining Schedulers

This portal is used to manage a scheduler's state and configuration.

If necessary, a scheduler can be brought on or offline by using the **Boot** or **Shut Down** buttons, respectively.

NOTE: This does not cancel or start the scheduler's batch process but rather guides the batch process to start or terminate its associated scheduler process. If you no longer need to use a certain scheduler, you need to shut it down and contact your system administrator to terminate the corresponding batch process and ensure it is not automatically started again.

Refer to [Understanding Schedulers](#) for more information.

You can access this portal from the **Admin Menu > Scheduler > Scheduler**.

The following zones may appear as part of the portal's **Main** tab page

- **Scheduler List.** This zone lists all scheduler definitions. Broadcast a record to display the details of the selected scheduler definition.
- **Scheduler.** This zone provides information about the selected scheduler definition.
- **Scheduler Runtime Details.** This zone provides run time information about the scheduling process managed by this scheduler as well as any statistics it has recently posted.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_SCHEDULER](#).

Defining Speed Profiles

This portal is used to maintain daily and template speed profiles.

You need to create daily speed templates first, then these can be referenced on your speed profile templates. Next, you can navigate to the [Geographic Area](#) portal to define the region on the map to which the speed profile template applies.

Refer to [Understanding Speed Profiles](#) for more information.

You can access this portal from the **Admin Menu > Scheduler > Speed Profile**.

The following zones may appear as part of the portal's **Main** tab page

- **Speed Profile List.** This zone lists all daily and template speed profiles. Broadcast a record to display the details of the selected record.
- **Speed Profile.** This zone provides information about the selected speed profile.

Where Used

Follow this link to open the data dictionary where you can view the tables that reference [M1_SPEED_PROFILE](#).

Viewing the Scheduler Registry

This portal is used to display the scheduler registry record for an entity managed by a specific scheduler.

Refer to [The Scheduler Registry](#) for more information.

You can access the portal from the **Main Menu > Scheduler > Scheduler Registry**. You are brought to a query portal with options for searching for a specific scheduler registry records. Once a scheduler registry record has been selected, you are brought to this portal to view the selected record.

The following zones may appear as part of the portal's **Main** tab page

- **Scheduler Registry.** This zone provides information about the registered entity.
- **Scheduler Read.** This zone displays the details that would be sent to the scheduler for this entity. This information varies depending on the type of registry object.

Integrations

Understanding Web Services

This section provides a high level description of web services and outbound messages provided by the product to support various types of integrations.

Refer to the Inbound Web Service portal for more information about each inbound service.

Refer to the appropriate Oracle Utilities Integration documentation for information about specific integrations.

Appointment Booking

The host system uses this service to request a list of available appointment time windows for an activity prior to sending it to the application for scheduling. Requests from the host specify key activity details and the time periods for which the scheduler should return a list of available appointment windows.

The system processes an appointment booking request from the host as follows:

- Using the appointment booking group specified on the request, identify the actual time windows during which appointments can be scheduled. If no appointment booking group is supplied by the host, the system uses the activity type's default appointment booking group.
- Geocode the proposed activity's site address.
- Determine the activity service area. This is done by invoking the Determine Service Area algorithm associated with the activity business object defined on the proposed activity's type.
- Calculate its arrival time window (ATW) and service time windows (STW) based on the requested Effective Time Windows (ETW) and preferred time windows (PTW). This is Activity Maintenance By Host done by invoking the Update Time Windows algorithm associated with the activity business object defined on the proposed activity's type.
- If scheduling priority is not provided, default it from activity type and business priority number if any.
- Default contractor eligibility and preferences based on activity type and service area.
- The information gathered above is sent to scheduler(s) responsible for appointment booking.
- Appointment information is returned. If there are multiple schedulers handling appointment booking requests, the first response is used.

Refer to [Appointments](#) for more information on how appointments are managed.

Activity Maintenance By Host

The host system uses the following inbound services to interface activities to be scheduled, sending updates to activity information, as well as inquiring as to the current status of an activity.

Maintain Activity By Host

The host system uses this service to interface a single activity and to update the activity information as needed. This service does not accept update requests to an activity that is already completed or canceled.

The following are important notes about this service:

- The system cannot evaluate which change warrants a recall and which does not. The host system should explicitly cancel an activity to request a recall. Refer to [External System Activity Cancellation](#) service for more information on the cancellation process.
- The host may indicate that the new activity, or changes to an existing activity, should only be accepted if it can be scheduled as requested by sending the request as a **conditional booking** request.
- If the activity is already dispatched and a change is requested as a conditional booking, the system is requested to ensure the "new" booking details can be scheduled as requested, if and when the crew returns their assignment. Depending on the type of work requested, it may be important to only accept the "new" booking if the "existing" booking can be guaranteed to be canceled. In other words, only accept the new if it can replace the existing.
- If an activity is already dispatched, the system cannot guarantee that the work will not be performed because the crew may not be connected to receive the latest status of the activity. If it is important to only accept a new booking as a

replacement of an existing booking, the host should also indicate that the request is a **guaranteed rebooking**, in which case the system will error and not accept the change.

- If it is acceptable that rebooking is not guaranteed, the host system should send in a request to cancel the existing activity and add a new activity with the new details using the [Maintain Activities \(Job\) By Host](#) service. This way the two bookings are kept independent.

Maintain Activities (Job) By Host

The host system uses this service to interface multiple activity as well as update their information once interfaced.

This service is useful in the following situations:

- When interfacing a set of activities that should be scheduled as a job
- When interfacing a set of activities that should be scheduled as a chain
- When sending in a rebooking request that cancels one activity and adds another to replace it as a single transaction.

Note that you can still add each activity, even if it is part of a job or a chain dependency, using the single activity service. Using this service though enables the system to validate the set of activities and either accept or reject all in a single transaction.

If conditional booking is requested but it should not be a guaranteed rebooking the host should send the request as a cancellation of the existing activity and a request to create a new activity with the new booking details. The system will confirm the new details with the scheduler and if can be scheduled the system will attempt to recall the activity from its currently assigned crew by calling the [External System Activity Cancellation](#) service. Refer to this service for more information.

If the crew has not performed the work and is connected then the recall would be successful and the existing activity would eventually be canceled for which a cancellation message is sent to the host as usual. If however the crew has completed the work the completion information would be sent to the host and it is the host's responsibility to cancel the new activity.

External System Activity Cancellation

The host system uses this service to cancel an activity. If the activity is not yet dispatched, it is canceled right away and a successful response is sent to the host system. If the activity is already canceled or completed, an error is issued.

Once an activity is dispatched, the system can only request a recall, but cannot guarantee the work would not be performed as the crew may not be connected to get notification of the latest status of the activity. The system responds with an indication that it will attempt to cancel the activity and has alerted a dispatcher of the situation.

Refer to [Activity Cancellation By Host](#) for more information.

Retrieve Activity and Assignment Details

The host system uses this service to inquire upon the current status and details of an activity.

Activity Completion To Host

In an integration where activity completion information is entered into the system by crews or external capacity contractors, the system forwards this information to the host system requesting the work.

The system initiates an **Activity Completion** synchronization request to send activity completion information to the host system when an activity is canceled or completed. If the host has initiated the cancellation or completion, no synchronization request is sent back to the host system.

Activity Status Update To Host

In an integration where crews utilize the application to work their schedule and provide status updates about activities they perform, the system may be configured to forward this information to the host system requesting the work.

This is not necessary if the host system is designed to request a status update on demand using the [Retrieve Activity and Assignment Details](#) inbound service.

If enabled on the external system's messages extendable lookup, the system initiates an **Activity Status Update** synchronization request to notify the host system of interim activity status updates once the activity is dispatched.

New Activities to Host

Any activity created in the system has to be communicated back to the host system associated with the activity type. This typically occurs when a mobile worker discovers an issue in the field that needs to be acted upon. It may also occur when a dispatcher creates an activity for a crew to assist another crew.

The system initiates a **New Activity** synchronization request to send the details of a new activity to the host system when it is created.

Contractor Management

The following section describes message exchanges with a capacity contractor company's system.

Activity to Contractor

All outbound messages to the contractor's system are sent to the external system referenced on the contractor's resource group record.

The system initiates an **Activity to Contractor** synchronization request to send activity details to the contractor's system when an activity is committed to that contractor. The same type of message is also sent when activity details change, as well as when the commitment is recalled.

Activity Commitment By Contractor

The contractor system uses this inbound service to approve or reject an activity committed to them.

If the contractor company does not approve an activity committed to them in a timely manner, i.e. within the agreed upon approval time, the system automatically assumes they have rejected the commitment and excludes this contractor from the activity, allowing it to be rescheduled as needed.

Activity Completion By Contractor

The contractor system uses this inbound service to send in completion information for an activity.

Crews Managed By Host

This section is relevant to an integration where crews are managed by the host system, and communicate status and completion information directly to the host system. In this type of integration, the system is mainly used for planning, scheduling, and dispatching purposes, but not to capture completion information.

All resource information contributing to establishing mobile workers, crews and shift templates, and exceptions are defined in the host system and synchronized with the system. This is then used to generate corresponding entities and shifts needed for scheduling.

The system sends generated shifts to the host system as well as any updates made to shift information.

The host system then notifies the system when shifts start which causes work to be dispatched back to the host system. The system notifies the host system of each dispatched task and in return, it is informed of task and shift status changes, but not completion information.

All messages to the host system are sent to the external system referenced on the crew record.

Maintain a Common Location

The host system uses this inbound service to synchronize common location information.

Maintain Service Area

The host system uses this inbound service to synchronize service area information.

Maintain Shift Weekly Template

The host system uses this inbound service to synchronize shift weekly template information.

Maintain Work Calendar

The host system uses this inbound service to synchronize work calendar information.

Process Employee Sync Requests

The host system uses this inbound service to synchronize an employee's information.

Update Shift Status

The host system uses this inbound service to notify the system of shift status changes.

Update Task Status

The host system uses this inbound service to notify the system of task status changes.

Dispatching A Task

The system initiates an **Activity Dispatched** synchronization request to notify the host system that an activity has been dispatched or advance dispatched prior to shift start.

Crew Shift Synchronization

The system initiates a **Crew Shift Synchronization** request to notify the host system of a new shift or a change to an existing shift.

This process relies on the resource monitor batch process to review and detect shift information changes associated with externally managed crews.

NOTE: It is important that you schedule the **Resource** is batch process to run frequently to meet your business requirements. Refer to [M1-RESRC](#) batch control for more information.

Depot Management

Request Depot Cutoff

A depot cutoff request closes all distribution depot runs that are planned to depart from the depot before a cutoff time.

Refer to [About Depot Cutoff](#) for more information.

Miscellaneous

Add GPS Coordinates For Resource

If crews are using the product's mobile application over devices that support GSP data capture, you can configure their shifts to allow the collection and sending of GPS data to the server.

If however the source of your GPS data is different, you may use this inbound service to interface GPS data from an external system. GPS data records have to be associated with either a crew or a vehicle known to the system. Make sure you reference a crew's name or a vehicle's tag respectively in the external identifier of each GPS data record sent in using this service.

Geocode an Address

An external system may use this inbound service to geocode an address. The service returns multiple addresses if more than one is found to match the request.

Configuring Outbound Messages

You would need to configure an [External System](#) record for each system your company sends messages to, as well as set up an outbound message type for each type of message handled by these systems.

Refer to [External System](#) for more information.

The application creates a synchronization request to manage the sending of an outbound message to an external system. The synchronization requests create the appropriate outbound messages based on outbound message types specified on a feature configuration. Find the [Feature Configuration](#) record for the **Schema Constants** feature type and populate the outbound message types you created on the corresponding message type options as needed. (It may need to be defined if it does not exist).

The system determines which synchronization request business object associated to use for each type of outbound message based on an extendable lookup configured for each external system.

For each external system record, add a corresponding record with same name, to the **External System Messages (M1-ExternalSystemMessages)** extendable lookup. Populate the list of messages with the appropriate information to either enable it or mark is as not applicable.

Synchronization request records are processed by the synch request monitor batch process, and it is at that time that the actual outbound message is prepared and sent to the external system.

NOTE: It important that you schedule the **Sync Request** batch process to run as frequently as needed to meet your business requirements. Refer to [F1-SYNRQ](#) batch control for more information.

Configuration Options

Batch Control Global View

This portal provides an "at a glance" view of the current running state of batch processes. Use the portal to find out if a batch process currently runs and how well it runs.

NOTE: Monitor batch processes as well as scheduler batch processes are configured to use a "Level of Service" algorithm defined on their batch control record to assess how well the process currently runs. Click [here](#) to see the algorithm types available for this system event.

When these algorithms are used with batch processes, they can answer the question of whether or not the batch process is running as expected. The diagnostics provide additional supporting information so that users can troubleshoot and otherwise monitor the processes. For example, slow running processes and job failures are highlighted.

You can access the portal from the **Main Menu > Batch > Batch Control Global View**.

Refer to [Batch Control](#) for more information.

The following zones may appear as part of the portal's **Main** tab page

- **Batch Controls.** This zone allows you to filter the list of batch controls to view.
- **Current State.** This zone displays information about the currently running process for a selected batch control.

Configuration Migration Assistant (CMA)

The Configuration Migration Assistant (CMA) provides customers with a flexible, extensible facility for migrating their configuration data from one environment to another e.g., from a development environment to a production environment.

Refer to CMA for more information.

This section describes CMA related topics that are specific to this product.

Executing Wholesale Migrations

Follow these high-level steps to migrate all administrative data for this product:

- Process the **Languages (F1-Languages)** migration request.
- Process the **Framework Schema Admin (F1-SchemaAdmin)** migration request.
- Process the **General System Options (F1-GeneralSystemOptions)** migration request.
- Process the **Framework Admin (F1-FrameworkAdmin)** migration request.
- Process the **Admin Data (M1-Admin)** migration request.

Process any of the other delivered migration requests as needed.

Data that Cannot be Migrated Using CMA

Configuration Migration Assistant does not currently support migration of objects with system generated IDs. The following maintenance objects are therefore not supported by CMA:

- Shift Weekly Template
- POU Template
- Mail Template
- Alert areas that reference excluded resources cannot be migrated because IDs for resources are system generated.

BI Configuration

This portal provides a bird's eye view of the configuration information for Oracle Utilities Analytics. It provides links and guidelines for the areas that need configuration to successfully run the extract transfer and load processes from Oracle Utilities Analytics.

Refer to the Oracle Utilities Analytics Administration Guide for more information.

You can access the portal from the **Admin Menu > BI Configuration > BI Configuration**.

The following zones may appear as part of the portal's **Main** tab page

- **Bucket Configuration List.** This zone lists bucket configurations to set up.
- **BI-Related Business Objects Information.** This zone lists additional control entities to set up.

The following zones may appear as part of the portal's **OWB-Based ETL** tab page

- **Outbound Sync BOs and Algorithm List.** This zone lists business objects and algorithms involved in the extract process.
- **BI-Oriented Extendable Lookup.** This zone lists extendable lookups to set.
- **External Data Source Indicators List.** This zone lists the external identifiers of the various sources for your analytics data.

Table Setup Sequence

Control Data Setup

This section lists the control data entities in the order they should be defined.

Level	Entity	Comment
1	Defining Installation Options	
1	Configuring Mapping Options	
1	Configuring Outbound Messages	This step should be performed for each external system your system is to integrate with.
1	Defining Locations	
1	Configuring Service Areas	
1	Defining Worker Capabilities	
1	Defining Equipment	
1	Defining Procedure Types	
1	Defining Shift Types	
1	Configuring Timesheet Options	
1	Configuring Crew Hierarchy	
1	Defining Crew Types	
1	Defining Dispatcher Types	
1	Configuring Mobile Devices	
1	Defining Mailing Lists	
1	Defining Break Types	
1	Defining Non Productive Task Types	
1	Defining POU Task Types	

2	Defining Period of Unavailability Types	
2	Configuring Activity Options	
2	Defining Mobile Worker Types	
2	Defining Vehicle Types	If your implementation utilizes loading capacities on vehicles refer to the Transfer of Goods Setup instead.
2	Configuring Scheduler Options	
2	Defining Mail Templates	
3	Defining Alert Types	
3	Defining Service Classes	
4	Defining KPIs	
4	Defining Dispatch Areas	
4	Defining Shift Cost Profiles	
5	Setting Up Global Configuration	
6	Configuring Deployment Types	

Resource Setup

This section lists resource related entities in the order they should be defined.

Level	Entity	Comment
1	Maintaining Alert Areas	
1	Maintaining Dispatchers	
1	Maintaining Mobile Workers	You would need to set up the contractor company before you can set up contractor mobile workers,
1	Maintaining Vehicles	You would need to set up the contractor company before you can set up contractor vehicles,
2	Maintaining Crews	You would need to set up the contractor company before you can set up contractor crews,
3	Maintaining Periods Of Unavailability	
3	Maintaining Shift Weekly Templates	

Transfer of Goods Setup

This section lists entities related to transfer of goods functionality in the order they should be defined.

Level	Entity	Comment
1	Configuring Transfer of Goods	

Contractor Management Setup

This section lists entities related to contractor management functionality in the order they should be defined.

Level	Entity	Comment
1	Defining Resource Group Types	
1	Configuring Capacity Model Options	
2	Maintaining Resource Groups	

BI Setup

Refer to the Oracle Utilities Analytics Administration Guide for more information.