

webMethods BPM Task Development Help

Version 10.15

October 2022

This document applies to Task Development 10.15 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2007-2022 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <https://softwareag.com/licenses/>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <https://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <https://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

Document ID: WTND-OLH-1015-20221015

Table of Contents

About this Guide.....	9
Document Conventions.....	10
Online Information and Support.....	10
Data Protection.....	11
 1 webMethodsTask Development Introduction.....	 13
About Task Development.....	14
Task Development Prerequisites.....	14
About JavaServer Faces.....	14
About Views.....	15
About CAF Controls.....	15
Other Resources for Task Development.....	16
Setting Task Development Preferences.....	16
 2 Working with Task Applications.....	 19
About Task Applications.....	20
Task Application Overview.....	20
 3 Working with Task Application Projects.....	 21
About Task Application Projects.....	22
Accessing the Portlet Application Project Wizard.....	22
Creating a Task Application Project.....	22
Deleting a Task Application Project in Designer.....	23
Migrating Task Projects and .view Files to JSF 2.x Format.....	24
 4 Task Overview.....	 27
About Tasks.....	28
Task Key Features.....	28
About Task Characteristics.....	29
About Task Status.....	30
About Task Voting.....	31
About Task Names and Task Type IDs.....	31
About webMethods Business Rules Integration.....	34
About Collaboration Tasks.....	35
About Collaboration Processes.....	36
About Task Comments and Attachments Sharing.....	38
 5 Creating Tasks.....	 39
About Creating Tasks.....	40
Accessing the Task Wizard.....	40
Creating a New Task with the Task Wizard.....	40
Creating a Collaboration Task.....	44

Creating a New Task from an Integration Server Document Type.....	45
Adding a Task to a Process.....	45
6 Working with Task Portlets.....	49
Task Portlet Overview.....	50
Viewing and Opening Task Portlets.....	63
Adding a New Portlet to a Task.....	64
Adding a New View to an Existing Task Portlet.....	71
Previewing a Task Portlet View in the Task Editor.....	73
Deleting Task Portlets and Portlet Views.....	74
Running Task Portlets and Views in My webMethods Server.....	75
7 Working with Security Roles.....	79
Security Roles Overview.....	80
Viewing Task Security Roles.....	80
Adding a Security Role to a Task.....	80
Binding a Security Role to a User Interface Control.....	83
Modifying and Deleting a Task Security Role.....	83
8 Configuring Tasks.....	85
Task Configuration Overview.....	86
Configuring Business Data (Task Inputs and Outputs).....	86
Configuring Votable Tasks.....	106
Adding E-form Support to a Task.....	108
Working with Services in Tasks.....	113
Working with Task Privileges.....	117
Working with Task Assignments.....	121
Working with Task Events.....	136
Configuring Simple Task Conditions.....	145
Working with Task Control Sets.....	148
Changing the Task Type Name.....	151
Using Gadgets to Render the Task Detail Page in webMethods Business Console.....	151
9 Updating Tasks.....	153
Task Update Overview.....	154
Updating Tasks Applying Optional Settings.....	155
10 Working with Task Searches.....	157
Task Search Overview.....	158
Implementing Case-Insensitive Searching.....	158
About Indexed Searches.....	159
About HPSTRA Searches.....	160
Search Considerations with Task Type Inboxes.....	161
Implementing an Indexed Search Content Provider for Existing Task Types.....	161
Searching Indexed Data in a Custom Task Inbox.....	165
Implementing a Standard Search Provider for Existing Task Types.....	167
Considerations When Publishing Tasks with Indexed Data.....	168

11 Working with Task Client Options.....	169
About Task Client Options.....	170
Task Client Option Definitions.....	170
Modifying Client Options for a Task.....	171
12 Working with Tasks in the Bindings View.....	173
About Tasks in the Bindings View.....	174
About Available Task Binding Properties.....	174
About Available Collaboration Task Bindings.....	175
About Task Indexed Search Content Provider Binding Properties.....	175
Modifying Task Binding Properties.....	176
13 Working with Task Pages.....	179
Task Pages Overview.....	180
Accessing a Task Page.....	180
Renaming a Task Root Page.....	181
Renaming a Task Page.....	181
Hiding and Displaying a Task Page.....	182
14 Working with Task Notifications.....	185
About Task Notifications.....	186
Task Notification Overview.....	186
Understanding Task Notification Behavior.....	187
Limitations to Subscription Access.....	188
About Task E-mail Notification Replies.....	189
About the Task E-mail Listener.....	190
Accessing the Task Notification Wizard.....	191
Creating a Task Notification.....	191
Adding a Task Action Link to a Notification.....	192
Extracting Data from a Task Notification Return E-mail.....	193
Modifying and Deleting a Task Notification.....	194
Sending Task Notifications to Non-Assigned Users.....	194
15 Working with Task Delegation.....	197
Task Delegation Overview.....	198
Limitations to Scheduled Delegation Access.....	198
16 Publishing Tasks.....	201
About Publishing Tasks.....	202
Publishing a Task.....	202
17 Deleting Tasks.....	205
Deleting a Task.....	206
Deleting a Task in My webMethods.....	206
Removing Tasks from Integration Server.....	206

18 Working with Task Attachments.....	209
Task Attachment Overview.....	210
Adding Attachment Capability to a Task.....	210
Sharing Attachments by Setting the Attachments Scope.....	211
19 Working with Task Comments.....	213
Task Comments Overview.....	214
Setting a Scope for Task Comments.....	214
20 Working with Custom Task Inboxes.....	215
Custom Task Inbox Overview.....	216
About Creating a Custom Task Inbox.....	216
Modifying the Results List for a Custom Task Inbox.....	217
Example Modifying the Default Search Fields and Results.....	220
Example Adding an Indexed Business Data Search Field.....	221
21 Working with Task Workflows.....	223
Task Workflow Overview.....	224
Code Samples.....	225
Implementation Components.....	226
Working with the Sample Applications.....	226
Key Implementation Points.....	228
22 Working with Tasks in the Faces Configuration Editor.....	231
About Tasks and the Faces Configuration Editor.....	232
23 Finding Task Applications in Software AG Designer.....	233
Searching for Task Applications.....	234
24 Working with Task Metadata.....	235
Task Metadata Overview.....	236
25 Working with User and Business Calendars.....	237
About Tasks and User and Business Calendars.....	238
26 Importing Task Applications from Earlier Versions.....	239
Importing Task Applications Overview.....	240
Indexed Field Considerations When Importing into Version 8.2 or Later.....	240
Moving from JSF 1.x to JSF 2.x Support.....	240
27 Working with Task Expressions.....	241
About Task Expressions.....	242
Task Expression Overview.....	242
Configuring Advanced Task Expressions.....	243

Building a Sample Task Expression.....	244
Viewing Sample Task Expression Syntax.....	245
About Condition Expressions.....	246
About Action Expressions.....	246
About Filter Expressions.....	247
About Expression Operators.....	247
About Date/Time Values in Expressions.....	249
About Expression Managed Beans.....	249
28 About Java Providers and Related Classes.....	251
Understanding Task Providers.....	252
29 Processing Task Change Events.....	257
Task Change Event Overview.....	258
Typical Use of Task Change Events.....	258
30 Working with the Task Engine APIs.....	259
About the Task Engine APIs.....	260

About this Guide

- Document Conventions 10
- Online Information and Support 10
- Data Protection 11

This guide contains the Task Development online help in PDF book format. The information in this guide is the same information that you can view via the Software AG Designer online help.

webMethods Task Development enables you to create tasks that address a particular kind of human activity that will be carried out at run time— for example, approving an order, or configuring a new employee's computer. These tasks can be used within a process developed in Designer, and you can also create stand-alone tasks that can be published to My webMethods Server. Within the My webMethods run-time environment, these stand-alone tasks can be started as often as necessary.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies service names and locations in the format <i>folder.subfolder.service</i> , APIs, Java classes, methods, properties.
<i>Italic</i>	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at <https://documentation.softwareag.com>.

In addition, you can also access the cloud product documentation via <https://www.softwareag.cloud>. Navigate to the desired product and then, depending on your solution, go to “Developer Center”, “User Center” or “Documentation”.

Product Training

You can find helpful product training material on our Learning Portal at <https://knowledge.softwareag.com>.

Tech Community

You can collaborate with Software AG experts on our Tech Community website at <https://techcommunity.softwareag.com>. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software AG news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at <https://github.com/softwareag> and <https://hub.docker.com/publishers/softwareag> and discover additional Software AG resources.

Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at <https://empower.softwareag.com>. Many services on this portal require that you have an account. If you do not yet have one, you can request it at <https://empower.softwareag.com/register>. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

1 webMethodsTask Development Introduction

■ About Task Development	14
■ Task Development Prerequisites	14
■ About JavaServer Faces	14
■ About Views	15
■ About CAF Controls	15
■ Other Resources for Task Development	16
■ Setting Task Development Preferences	16

About Task Development

Software AG Designer Task Development enables you to create tasks that address a particular kind of human activity that will be carried out at run time, such as approving an order, or configuring a new employee's computer.

These tasks can be used within a process developed in Designer, and you can also create stand-alone tasks that can be published to My webMethods Server or Integration Server. Within the run-time environment, these stand-alone tasks can be started as often as necessary.

In addition, Task Engine provides a set of APIs that you can use to interact with tasks. For more information, see [“Task Update Overview” on page 154](#).

Task Development Prerequisites

Before you develop task solutions, you should have a fundamental understanding of the following:

- The Eclipse development environment, including the use of perspectives, views, and editors.
- Web development, including development of Web and portlet applications, and Web services.
- The Java Platform, Enterprise Edition, formerly known as Java 2 Platform, Enterprise Edition/J2EE.
- JavaServer Faces (JSF) technology.
- Concepts of service-oriented architecture (SOA).

Additionally, you should be familiar with the webMethods product suite and its integrated approach for developing, deploying, and running Web-based applications.

About JavaServer Faces

Task applications created in the Designer Task Development support the JavaServer Faces (JSF) standard. Task applications created with version 9.6 and earlier were implemented with JSF 1.x support. Beginning with version 9.7, JSF 2.x is supported, enabling you to develop facelet applications.

Support of facelets provides you with the ability to use:

- XHTML to create web pages.
- Facelets tag libraries to augment JavaServer Faces and JSTL tag libraries.
- The Java Unified Expression Language (UEL).
- Templates for components and pages.

Beginning with version 9.7, you can create new task projects in either of two ways:

- As JSF 1.x applications that use View Declaration Language (VDL) files implemented as .view files. These projects will behave exactly like those developed in earlier projects. You can convert them to JSF 2.x facelet applications with VDL files implemented as .xhtml files.
- As JSF 2.x applications that support the use of .xhtml view files. In this case, when you add a new portlet or an individual view file, the new view file is added in JSF 1.x format, and you must migrate it to JSF 2.x format.

Task applications developed with JSF 1.x in earlier versions can be migrated into JSF 2.x facelet applications at the project level. Individual .view files can also be migrated to .xhtml files.

For more information, see “[Migrating Task Projects and .view Files to JSF 2.x Format](#)” on page 24.

For more information about working with facelets, see *webMethods CAF and OpenUI Development Help*.

About Views

Within each task portlet, there are one or more portlet views that define the user interface for the portlet. These portlet views are defined as View Declaration Language (VDL) files. Composite Application Framework (CAF) supports two types of VDL files:

- For JSF 1.x projects, a VDL file is implemented as .view file structured with a proprietary Software AG XML schema.
- Beginning with Task Development version 9.7, CAF also supports VDL files in .xhtml format for JSF 2.x projects. You can convert .view files to .xhtml files as described in “[Migrating Task Projects and .view Files to JSF 2.x Format](#)” on page 24.
- In Task Development version 10.5 and later, the default format for VDL files is .xhtml.

You can open a task view in the task editor, enabling you to modify the view to your custom requirements. You can:

- Drag CAF controls from the Palette view directly on to the design canvas.
- Specify properties for controls, for example, isVisible, isRendered, isDisabled.
- Bind controls to managed beans.
- Import portal resources into your view.

About CAF Controls

To develop a custom user interface for a task application, published to My webMethods Server, you can add CAF controls to View Declaration Language (VDL) files. Composite Application Framework (CAF) provides two sets of its CAF controls:

- Controls for JSF 1.x applications in .view files
- Controls for JSF 2.x facelet applications in .xhtml files

Both sets of CAF controls are available in the Palette view, and are essentially identical, providing the same behavior and control attributes in each case. When you add a control to a VDL file, it is added using the appropriate code for the container file, either .view or .xhtml. In both cases, you can view and modify the control attributes through the Properties view, or directly in the code if you prefer.

For more information about CAF controls, see *webMethods CAF Tag Library Reference*.

Other Resources for Task Development

In addition to the information contained in these online help topics, you can also find information about working with tasks in the following product guides:

- *webMethods Task Engine User's Guide*. This publication describes how to interact with and administer tasks in the My webMethods Server run-time environment.
- The *webMethods Task Engine API and Service Reference*. This publication describes the Task Engine API and services that are available for interacting with tasks from external applications.

Setting Task Development Preferences

You can set a number of Task Development preferences within the Eclipse environment.

> To configure Task Development preferences

1. In the main menu, click **Window > Preferences**.
2. In the preferences list, expand **Software AG** and click **Task Development**.
3. On the Task Development page, select any of the following preferences:

Preference	Description
Automatic task creation using "Implement with New Task" in the process editor	Select this preference to enable simplified creation of a task from within the process editor. When this check box is selected, the Task wizard does not appear, and the task is created with the default settings. This check box is not selected by default.
Default task project name created from the process editor	<p>Use this box to specify the default name for the composite application project containing tasks you create from within the process editor. The default value is {0}Tasks.</p> <p>The variable {0} represents the name of the process you are adding a task to. For example, if your process is named New_Process, with the default value of {0}Tasks, the tasks you create in the process will appear within the Solutions Explorer in a folder named New_ProcessTasks.</p>

Preference	Description
	<p>No other variables are available. Alternatively, you can type in any valid text string as the default name (no spaces are allowed, or special characters other than '\$' and '_').</p>
Always use GUID for new Task Type ID	<p>Select this preference to apply an automatically generated globally unique identifier (GUID) to a new task when the task is created. An example of the type of GUID that is automatically generated is:</p> <p>1436DB52-6E63-311A-5D03-DD3E141EF483.</p> <p>The task type ID cannot be changed after the task is created.</p> <p>When this preference is not selected, you have several choices when you create a new task with the New Task wizard. You can:</p> <ul style="list-style-type: none"> ■ Click the Generate button to have Designer create a GUID for the task type ID. ■ Type your own unique task type identification. ■ Accept a task type ID based on the project name and task type name. <p>In the latter case, a default name of [ProjectName].[TaskTypeName] is entered in the task ID box. In most cases, this will create a unique task type ID. However, if you are creating a task type with the same task type name as another task type, you will not be able to use this naming solution. You can modify this into any unique text string to create the task type ID (no spaces are allowed, or special characters other than '\$' and '_').</p> <p>This preference is ignored if the automatic task creation preference (above) is selected; in this case, a GUID is always automatically generated.</p> <p>This check box is selected by default.</p>
Create Task View Portlet for new Task	<p>Select this preference to always create a task view portlet when a new task is created. This check box is selected by default. This behavior can be overridden when you create a task with the task creation wizard. If a task view portlet is not created, the task will have no user interface in My webMethods. For more information on this option, see “Creating a New Task with the Task Wizard” on page 40.</p>
Create Task Start Portlet for new Task	<p>Select this preference to always create a task start portlet when a new task is created. This check box is not selected by default.</p>

Preference	Description
	This behavior can be overridden when you create a task with the task creation wizard.
Create Task Inbox Portlets for new Task	Select this preference to always create two task inbox portlets when a new task is created (Task Inbox Bar portlet and the Task Inbox Results portlet). This check box is not selected by default. This behavior can be overridden when you create a task with the task creation wizard.
Use task indexed search content provider	Select this preference to always use an indexed search content provider when a new task is created. This check box is not selected by default. This behavior can be overridden when you create a task with the task creation wizard. For more information about indexed search content providers, see “About Indexed Business Data Fields” on page 89 .

4. Click **Apply** to apply your selections and continue working with other preferences, or click **OK** to apply your selections and close the Preferences dialog box.

2 Working with Task Applications

■ About Task Applications	20
■ Task Application Overview	20

About Task Applications

Software AG Designer supports the creation of a proprietary type of application that is specific to tasks. *Task applications* must be created within an existing *task application project*.

Although you can extensively modify and customize the task applications you create, certain aspects of the task application are not available for changes. This is because of the proprietary nature of task applications and the need for them to interact with other webMethods product suite components.

Task applications are deployable only to Software AG runtimes. Supported runtimes are My webMethods Server and Integration Server.

For tasks, running on My webMethods Server, task applications can include a set of customizable task user interface portlets. After successful deployment both the task and its user interface run on the My webMethods Server runtime.

Task Application Overview

A *task application* is a CAF portlet application project (a type of dynamic web project). A task application project can contain one or more tasks and the task supporting components such as assignments, events, control sets, and notifications. Depending on the target runtime, task application projects can contain one or more task user interface components such as portlets or views.

All tasks must be created within a task application project, and all tasks in a single task application project must target the same type of runtime, either My webMethods Server or Integration Server. The number of tasks in a task application project is determined primarily by the complexity of the human activity you are addressing, and by the amount of granularity you want to provide to your project structure. To deploy a task to the target runtime, you must publish the task application project that contains the task; in doing so, all of the tasks in the task application project are deployed to the server.

For example, suppose you are supporting a human activity that comprises six separate task steps. You can create one task application project with all six tasks within it, or you could create six task application projects with one task in each.

In the first case, you can deploy and update all six tasks simultaneously by publishing the single task application project. In the second case, you are able to deploy and update each task separately.

The decision to package multiple tasks into one task application project can also be driven by a requirement to share common Java classes or run-time user session data. The method of implementation depends on how you want to work with the project components.

3 Working with Task Application Projects

■ About Task Application Projects	22
■ Accessing the Portlet Application Project Wizard	22
■ Creating a Task Application Project	22
■ Deleting a Task Application Project in Designer	23
■ Migrating Task Projects and .view Files to JSF 2.x Format	24

About Task Application Projects

Software AG Designer enables you to create tasks that run on the webMethods Task Engine. All tasks must be created within a task application project (a generic portlet application project). For instructions, see [“Creating a Task Application Project” on page 22](#).

Task application projects that you publish to a Task Engine instance, running on My webMethods Server can be created to support either the JSF 1.x standard or the JSF 2.x standard. By default, task application projects are created with JSF 2.x standard support. For more information, see [“About JavaServer Faces” on page 14](#).

For information about creating portlet application projects in general, see *webMethods CAF and OpenUI Development Help*

Accessing the Portlet Application Project Wizard

You can access the portlet application project wizard in the following ways:

- On the UI Development perspective:
 - By clicking **File > New > Portlet Application Project**.
 - By clicking the **New > Portlet Application Project** arrow button in the main toolbar.
 - By clicking the **New** button on the main toolbar and then clicking **Software AG > UI Development > Portlet Application Project**.
- By clicking the **New > Other** command anywhere it is available, then expanding the **Software AG > UI Development** folders and clicking **Portlet Application Project**.
- By clicking the **New** button on the Project Selection page of the Task wizard.
- By right-clicking the **Tasks** entry in the Solutions view and clicking **New Task Portlet Application**.

Creating a Task Application Project

A task must be created in an existing task application project, which is a type of portlet application project. You can create your project with either JSF 1.x or JSF 2.x support. For more information, see [“About JavaServer Faces” on page 14](#) and *webMethods CAF and OpenUI Development Help*.

➤ To create a new task application project

1. Start the new portlet application project wizard as described in [“Accessing the Portlet Application Project Wizard” on page 22](#).
2. In the New Portlet Application Project wizard, in **Project Name**, type a name for the portlet application project.

3. Accept the default location for the project, or specify a custom location.
4. In **Target Runtime** list, select the required runtime.

The options are **My webMethods Server** and **Integration Server**. The default is **My webMethods Server**.

5. In **Dynamic web module version**, accept the default value.
6. In **Configuration**, use the default value for the selected runtime. The default values are **CAF Portlet Application** for **My webMethods Server** and **Default Configuration for Integration Server**.
7. (Optional) In **Working sets**, click **Add Project to working set** to use a subset of resources, and then click **Select** to select the working set resource for the project.
8. In **Initial Project Contents**, select one of the following in the **Project Templates** list:
 - **OpenUI Shell** - creates a task application project with JSF 2.x support. The project is created without any task portlets, but it does contain a page.xhtml page. You can publish task applications, created with the **OpenUI** shell only to **My webMethods Server** runtimes.
 - **Empty Project** (default). The task application project is created with a minimum of folders and resources, with JSF 2.x support. You will create your project structure and resources later by adding tasks with the New Task and New Portlet View wizards.
9. Click **Next** and specify any custom settings you want to apply to the project on the following screens, or click **Finish** to create a project with the default configuration.

Deleting a Task Application Project in Designer

Task application projects can be deleted in Software AG Designer. Deleting a task application project also deletes all tasks or task interfaces within that project. You cannot recover a deleted task application project or its contents.

➤ To delete a task application project

1. Locate the task application project you want to delete in the Solutions view.
2. Right-click the application project and click **Delete**.
3. If you delete a task application project that has already been published, you may also want to delete the task types from the runtime environment. For more information, see [“Deleting Tasks” on page 205](#).

Migrating Task Projects and .view Files to JSF 2.x Format

If you have existing task application projects developed in a version prior to 9.7, or if you have a JSF 1.x project developed in 9.7 or later, you can migrate those projects to JSF 2.x/XHTML support. You cannot change the project back to JSF 1.x format after you migrate it. In addition, you can migrate .view files to .xhtml files.

For more information, see:

- [“Migrating a JSF 1.x Task Application Project to JSF 2.x Format” on page 24](#)
- [“Migrating .view Files to JSF 2.x Format” on page 24](#)

Migrating a JSF 1.x Task Application Project to JSF 2.x Format

You can migrate a JSF 1.x task application project to JSF 2.x format. Migrating the project updates the project facets to ensure they are using the latest versions and converts all .view files in the project to .xhtml files.

➤ To migrate a JSF 1.x projects to JSF 2.x format

Note:

Migration of a JSF 1.x project cannot be undone.

1. Select the project or projects you want to migrate in the Project Explorer view or Navigator view.
2. Right-click the project and click **CAF Tools > Migrate to JSF 2.x**.

If any resources in the selected projects need to be migrated, the Migrate projects panel appears. Otherwise the Migrate .view files panel appears.

3. If the Migrate projects panel appears, set the selections to identify the projects you want to migrate, then click **Next**.
4. On the Migrate .view panel, set the selections to identify the .view files you want to migrate, then click **Finish**.

Migrating .view Files to JSF 2.x Format

You can migrate individual .view files within a task application project into .xhtml files. This is convenient when you have added a JSF 1.x portlet to a JSF 2.x project and you want to migrate the portlet views individually. While it is permissible to mix both .view files and .xhtml files in a JSF 2.x application, you are advised to migrate the added .view files.

Important:

Be sure to migrate the entire task application project *before* you migrate any individual .view files, as described in [“Migrating a JSF 1.x Task Application Project to JSF 2.x Format” on page 24](#). Migrating .view files in a non-migrated project will result in missing libraries and runtime errors.

➤ **To migrate .view files to JSF 2.x format**

Note:

Migration of .view files cannot be undone.

1. Locate the task application project you want to work with in the Project Explorer view or Navigator view.
2. Do one of the following:
 - To migrate an individual .view, right-click the .view file and click **CAF Tools > Migrate to JSF 2.x**.
 - To migrate all .view files in the portlet, right-click the portlet folder and click **CAF Tools > Migrate to JSF 2.x**.
 - To migrate all .view files in the application, right-click the WebContent folder and click **CAF Tools > Migrate to JSF 2.x**.
3. On the Migrate .view panel, set the selections to identify the .view files you want to migrate, then click **Finish**.

4 Task Overview

■ About Tasks	28
■ Task Key Features	28
■ About Task Characteristics	29
■ About Task Status	30
■ About Task Voting	31
■ About Task Names and Task Type IDs	31
■ About webMethods Business Rules Integration	34
■ About Collaboration Tasks	35
■ About Collaboration Processes	36
■ About Task Comments and Attachments Sharing	38

About Tasks

When you create a *task* within Software AG Designer, you are creating what is essentially a template for a particular kind of human activity that will be carried out at run time — for example, approving an order, or configuring a new employee's computer. Tasks are created as part of a *task application project*, and are usually included as a part of a *business process* developed and deployed with Software AG Designer.

When you publish a task application project from Designer to the target runtime, the individual tasks contained by the task application project are created on the server (or updated, if the project has been published previously), and can be used to start a run-time instance of a task. A task can be added to a process model, also created in Designer.

The various component views for tasks, published to My webMethods Server are implemented within the Designer environment as *portlets*, and a task can contain a single portlet, or two or more portlets. Many of the procedures for creating and defining a task portlet are very similar to the procedures for developing any other portlet. For more information about creating and working with portlets in general, see *webMethods CAF and OpenUI Development Help*.

As you develop a task, you can preview the results of your work by publishing and running the task portlet, pages, or views to My webMethods Server from within Designer. For more information, see [“Running Task Portlets and Views in My webMethods Server” on page 75](#).

For more information about what you can do with tasks, see [“Task Key Features” on page 28](#).

Task Key Features

The following key features are available with the tasks you create:

- **webMethods Business Rules Integration.** Software AG Designer enables you to determine task assignments using rule entities (such as decision tables and business rule sets) in webMethods Business Rules, meaning you can move task assignment logic outside of the task type. In addition, you can reference any assignment decision table from multiple task types, thereby centralizing your assignment logic. Task instances can be started by a rule entity by invoking a Manual Decision process action as a rule result. For more information, see [“About webMethods Business Rules Integration” on page 34](#).
- **E-form Integration.** Software AG Designer enables you to integrate electronic forms (*e-forms*) into the task workflow. Supported e-form templates can be imported into Software AG Designer where they are converted to a standard IS document type. You can use the IS document type as the business data for a task, or for any other supported IS document type use. You can implement the ability for users to download and upload an e-form from an e-form enabled task instance interface running in My webMethods.
 - For more information about the use of e-forms in the webMethods product suite by using My webMethods Server as your e-form repository, see the PDF publication *Implementing E-form Support for BPM*.
 - For more information about implementing e-form support in a task, see [“Adding E-form Support to a Task” on page 108](#).

- **Collaborative Workflow.** Both Software AG Designer and My webMethods Server provide a set of design-time and run-time functions that allow users to create, modify, assign, and complete spontaneous collaborative, task-based workflows on an "as-needed" basis.

A key portion of the collaborative workflow feature enables run-time task users to quickly create one or more collaboration tasks that can be assigned to others users, so that those users can collaborate in completing the parent task. These collaboration tasks exist within a unique collaboration process model that is created when the first collaboration task is created. It is also possible to create tasks for business processes that will dynamically generate and assign one or more collaboration tasks automatically, based on task business data. In this case, it is possible to design the specified collaboration task to be assigned not only to one or more users, but to groups and roles as well.

A number of general task capabilities are also available to enhance the collaborative work environment, including:

- **Comments**—By default, tasks provide the ability for users to enter, modify, and view free-form comments within the task they are working on.
- **Attachments**—By default, tasks provide the ability for users to add or delete attachments.
- **Data sharing**—You can define how business data and other task information is shared among collaborative users.

For more information about developing collaboration tasks, see [“Creating a Collaboration Task” on page 44](#).

- **Task E-mail Notifications.** Software AG Designer enables you to create a task e-mail notification that can be sent to any My webMethods Server user who is assigned to the task and subscribed to the notification. You can include a Task Action Link control in the e-mail notification, enabling a My webMethods Server user to interact with the task without requiring a connection to My webMethods Server. For more information about task e-mail notifications, see [“Working with Task Notifications” on page 185](#).
- **External Access Through APIs.** The Task Engine provides a set of APIs that you can use to interact with tasks running on the Task Engine within My webMethods Server. For more information, see [“Task Update Overview” on page 154](#).

About Task Characteristics

When you create a task, you define how the task will appear and behave in the run-time environment by specifying characteristics such as:

- The input and output information for the task.
- Assignment of the task.
- Actions that will take place upon certain task events (for example, when the task is queued, expired, or completed).
- The user interface(s) layout seen by the user at run-time (for tasks published to My webMethods Server).

- Information displayed to the user.
- Controls available to the user (for tasks published to My webMethods Server).
- One or more control sets that define a specific combination of assignments and events.

You can create a task that is used as part of a process, in which case the task is triggered by the existence of the task input conditions or information defined within the process. You can also create *stand-alone tasks* that can be started manually in the run-time environment as often as necessary. Stand-alone tasks can also be started remotely through the Task Engine API, for more information, see [“About the Task Engine APIs” on page 260](#) and *webMethods Task Engine API and Service Reference*.

By defining task assignments at design time, a run-time task can be routed to one or more My webMethods roles, groups, or users based on those assignment definitions.

All task functionality is subject to the inherent role-based access feature of My webMethods. At design time or run time, you can specify which functions included in the task are available to a My webMethods role.

For more information about making task functions available to roles, see [“Working with Security Roles” on page 79](#) and *webMethods Task Engine User’s Guide*.

About Task Status

After a task is started, it can transition through several states. Task status can be set manually by a user with the proper privileges, or automatically by the Task Engine as a result of an event action. The following states are available (user-defined states are not supported):

- **Active**—The task is running normally and is available for user interaction. A task is placed in Active status when it is started, and can be returned to Active status as a result of a manual status change by a user or as a result of a task event evaluation. Only Active tasks appear in the user inboxes.
- **Completed**—The task is complete as a result of manual completion by a user or as a result of a task event evaluation. No further work can be done on a task that is completed, other than deleting it.
- **Error**—The task has transitioned to an error condition as identified operationally by the Task Engine or as a result of a task event evaluation. No further work can be done on a task that is in Error status, other than deleting it.
- **Cancelled**—The task is cancelled as a result of manual cancellation by a user or as a result of a task event evaluation. No further work can be done on a task that is cancelled, other than deleting it.
- **Suspended**—The task is suspended as a result of manual suspension by a user or as a result of a task event evaluation. A suspended task is no longer available in a task type inbox or user’s Task Inbox and can be viewed only on the Task List Management page, and no events or assignments are evaluated for a suspended task. A suspended task can be resumed (placed back into Active status) only by a manual user action.

- **Expired**—The task has expired as a result of a manual action by a user or as a result of a task event evaluation.
- **New**—This status is not displayed in the inbox and Task List Management status display; tasks are in New status only momentarily; immediately after the task starts, it transitions from New to Active status. New status can be used when defining a task event, for example, to detect newly started tasks.

After a task is started, it remains in the system (regardless of its state) until it is deleted. A task can be deleted manually, or global task rules can be created in My webMethods Server to automatically delete tasks.

For example, My webMethods is installed with a global Delete Task schedule rule. This rule deletes any task in Completed, Cancelled, Expired, or Error status after the task has been in that state for a specified period of time (set to seven days by default). This rule is evaluated once per day. For more information on global task rules, see the PDF publication *webMethods Task Engine User's Guide*.

About Task Voting

When creating task types for Task Engine on My webMethods Server, task developers can enable the task types for voting, and configure a specific voting strategy. Task voting provides an additional collaborative task workflow which allows multiple users to work on a task and modify the task status through a collective decision.

When a task type is enabled for voting, task type developers configure the minimum share of assignees that must complete (or cancel) the task in order to allow transition to the required status. Different voting strategies provide the option to set that threshold as an absolute value (i.e. five individual users) or a percentage out of the total number of assignees.

For example, when you enable a task type for voting and configure the voting strategy with the default percentage value, which is 51%, Task Engine will require that more than half of the assignees vote for the same option. At runtime, a task instance of the type is assigned to a group with a total of 20 individual users and the task user interface is configured to display two buttons - one for cancelling, and another one - for completing the task. With the default threshold of 51%, Task Engine requires at least 11 users to vote for the same option before updating the status of the task instance.

Task developers can create a votable task type from scratch, or add voting capabilities to all existing task types, created with the current version, or a later version. Task types, created with earlier versions of Software AG Designer do not provide voting capabilities. Certain considerations apply when adding voting to existing task types. For more information, see [“Runtime Considerations for Votable Tasks” on page 107](#).

About Task Names and Task Type IDs

Tasks are identified by a *task type name* and by a *task type ID* (identification):

- **Task Type Name.** You specify this value when the task is created, and it can be modified later as desired. When you implement a new task from a process model, the step name is used for

the task type name. It is possible to create two or more task types with the same name; for more information, see “About Duplicate Task Names” on page 32 below.

The task type name appears in Designer (for example, in the Solutions view and on the editor tabs) and in all locations of the My webMethods interface where the task type name is identified.

For information about changing the task type name, see “[Changing the Task Type Name](#)” on page 151.

- **Task Type ID.** This name is used by both My webMethods and Software AG Designer to identify a task programmatically. It cannot be modified after it is assigned.

The task type ID is displayed in the General Information area of the task editor **Overview** tab in Designer and in the task information when the task is viewed from the Task Configuration table on the Task Engine Administration page:

Note:

The task type ID is subject to the **Always use GUID for new Task Type ID** preference (for more information on setting this preference, see “[Setting Task Development Preferences](#)” on page 16). When you select this preference, Designer uses an automatically generated globally unique identifier (GUID) as the task type ID when the task is created. An example of the type of GUID that is automatically generated is: 1436DB52-6E63-311A-5D03-DD3E141EF483.

When the **Always use GUID for new Task Type ID** preference is not selected, you have several choices when you are creating a new task. You can:

- Accept a task type ID based on the project name and task name. In this case, a default name of [ProjectName].[TaskName] is entered in the task type ID box. This solution may not be possible if you are creating task types with duplicate task type names.
- Click the **Generate** button on the New Task wizard to have Designer create a GUID for the task type ID.
- Type your own unique task type identification.

When you implement a new task from a process model, a GUID is always automatically generated. Each task type ID must be unique within the Eclipse workspace you are working with. If the task type ID is not unique with the workspace, Designer will not allow you to create a task.

Important:

Designer does not check for uniqueness across multiple workspaces. If you are developing tasks across two or more workspaces, you must ensure that each task you create has a unique task type ID regardless of the workspace in which it is located. If you create tasks that share a task type ID in different workspaces, you will experience problems when the tasks are published to the same My webMethods Server. Specifically, the most recently published task type will overwrite any existing task type with the same task type ID. Use of the generated GUID is recommended to greatly reduce the possibility of duplicate task type IDs.

About Duplicate Task Names

In some instances, it may be necessary to create two or more task types with the same task type name. For example, multiple process models might contain a user task step named “Cancel Order”,

but, to accommodate different conditions in the order processing workflow for different departments, a different task type is implemented and invoked for each process model.

Programmatically, this is not a problem because the Task Engine tracks each task type by its task type ID, not its task name. However, the existence of duplicate task type names in the My webMethods user interface has two major ramifications:

- You will encounter problems when you publish task types with duplicate names unless you take the task editing precautions described below. Specifically, the most recently published task type will overwrite any existing task type with the same task type name.
- Duplicate task type names could be potentially confusing to task users. This would be especially true if multiple custom task inboxes are present, all with identical names.

Duplicate Task Type Name Precautions

Whenever you create a task type with a duplicate name, and that task type contains a task application root page as part of the interface, *before you publish the task type, you must use the task editor to manually rename the task application root page* to make it unique among all tasks deployed to My webMethods Server; otherwise, the most recently deployed duplicate task type will overwrite the existing task application pages. This will also ensure that custom inbox page names will be unique. For information about modifying the task application root page, see [“Renaming a Task Root Page” on page 181](#).

In addition, the **Advanced** tab of the My Inbox and Task List Management pages enable a user to filter the task results by task type name. In the event of task types with duplicate names, the list will contain an entry for each duplicate name, potentially making it difficult for the user to select the desired task. To alleviate this problem, the user can hover the cursor over a task name entry in the list; this displays a tooltip that contains the task type description text entered by the task developer as well as the unique task type ID.

About User Task Step Names

Designer uses the user task step name (label) to create the task type name when you implement a new user task from within the process modeler. When you add a new user task step to a process, the default behavior is to label the step as “Task n ”, where n is a numerical identifier.

Note that it is quite easy to create task types with duplicate names because of this behavior. Any number of processes can have a user task labeled “Task3”, and if you make no changes to the task type name, you will obtain duplicate task names. Although Designer allows this (see “About Duplicate Task Names” on page 32 above), you may be creating confusion for task users by unnecessarily creating duplicate names. In each case, give careful consideration to the resulting task type name.

You can change the display name of the task in both the process editor and in the task editor. These names do not have to match, as the task is programmatically tracked by the task type ID. However, best practices recommend that you keep the display names synchronized.

In the process editor, you can modify the task name of the user task step on the process canvas and on the **General** page of the Properties view. You can also select an existing task type on the step’s **Implementation** tab; for more information, see *webMethods BPM Process Development Help*.

About webMethods Business Rules Integration

Task developers can take advantage of Task Engine integration with webMethods Business Rules to define task assignments by using Task Engine rule entities such as business rule sets and decision tables. Instead of creating task assignments within the task type itself, you can reference a business rule set or decision table within webMethods Business Rules to determine the task assignment based on the values of task business data.

This approach enables you to place the task assignment logic outside of the task, in the decision table, and you can reference the same business rule entity from multiple task types. If you decide to change the assignment conditions, you apply a single change to the rule entity, which is then applied to all task types that reference it. Otherwise, if you were to maintain your assignment logic in the task types themselves, you would have to modify each task type individually when you change the assignment conditions.

- For more information about task assignments with rule entities, see [“Configuring a Task Assignment with a Decision Table” on page 125](#) and [“Configuring a Task Assignment with a Business Rule Set” on page 126](#).
- For more information about working with decision tables and business rules in general, see *webMethods BPM Rules Development Help*.

Business process developers working in Software AG Designer can also use a business rule process action to start a task instance. For example, consider a business process for handling a new wholesale customer account application submitted by a distribution company.

Based on the various pieces of customer data submitted to the process (such as the company’s annual sales, credit rating, and past purchase history), the business process interacts with a decision table that calculates a net discount to be applied to all future purchases, using the business data provided by the process. The decision table determines the discount amount and returns it to the business process.

In some cases, however, the application being handled by the business process may be missing key information that prevents the decision table from determining the discount amount. To address this case, a Manual Decision process action can be included in the decision table.

When the decision table cannot determine a discount amount, the Manual Decision process action starts an instance of a task type that enables the task assignee to review the case and determine a discount amount manually. Upon completion of the task, the discount amount is returned to the process.

The task type instance started by the Manual Decision process action requires no special configuration, although it must be implemented to accept and present the business data provided by the business process and decision table, and provide the ability for the assignee to specify a discount amount. Assignment of the task can be based on business data provided by the process, by standard task assignment mechanisms, or through a decision table.

- For more information about invoking business rules from business processes, see *webMethods BPM Process Development Help*.
- For more information about working with process actions and business rules in general, see *webMethods BPM Rules Development Help*.

About Collaboration Tasks

You can implement a collaborative workflow environment for run-time users by defining and publishing collaboration tasks—that is, tasks that have been specifically configured to operate in a shared work environment.

Enabling a collaborative workflow environment requires you to define a parent/child relationship between two or more task types. When a run-time process (or user) is working with a task that is designated as a parent task, new child collaboration tasks can be created dynamically and assigned to other users.

This can be done automatically (within a process) or manually, by a user who is working with the task. For more information, see:

- [“Using Collaboration Tasks in a Process” on page 35](#)
- [“Using Collaboration Tasks Manually” on page 36](#)

Using Collaboration Tasks in a Process

When you include a task representing human activity in a business process, the potential type and number of activities required to complete the task can be quite numerous. While it is possible to construct detailed logic within a task to anticipate all of these possible outcomes, that approach can be very labor intensive and error prone. In addition, if the business use case around the task changes, updating the logic can be difficult.

You can create and configure a parent task type in Designer, and associate it with one or more collaboration tasks. You can then configure task actions in the parent task so that when specific business data is received by the parent task, one or more collaboration tasks are created by the actions and assigned to specific My webMethods Server groups, roles, or users.

Then, when you include the parent task in a business process, the required collaboration tasks will be queued depending on the business data passed to the task in the process pipeline.

For example:

Suppose a satellite television company runs a business process for new orders, containing a task with the nominal activity of installing a dish and receiver. However, different equipment and installation teams are required depending on the service ordered by the customer. In addition, different parts of the country require different kinds of equipment for each type of service. The business data for the process can be configured to contain a field that defines the type of service and a postal code or other designator describing the service area.

Depending on the specified service type and location data contained in the pipeline for each individual order, the parent task in the process can queue an associated child task that is configured to determine its assignment value from a webMethods Business Rules decision table or business rule set defined in the child task.

The child task is delivered to a role or group representing the required installation team, and another (similarly configured) child task is queued and routed to a role or group representing the

inventory team to prepare the correct type of equipment for the installers. The parent task can be configured to complete automatically when all of the collaboration tasks are completed.

For more information about configuring collaboration tasks, see [“Creating a Collaboration Task” on page 44](#). For more information about working with webMethods Business Rules to create task assignments, see [“Configuring Assignees for a Task Assignment” on page 123](#).

Using Collaboration Tasks Manually

You can configure a task so that an individual user (with the proper permissions) can create collaboration tasks from the task after opening it in an inbox. The user can assign these collaboration tasks to other users to help complete the parent task.

Note:

In this case, the user can assign the collaboration task only to a My webMethods Server user; assignment to groups or roles is not possible.

For example:

Suppose a customer service representative (CSR) receives a task to investigate and resolve a customer's complaint that issues of a magazine subscription are no longer arriving.

To determine the cause, the CSR needs information from a database administrator and the circulation manager. The CSR can create and assign collaboration tasks to both these individuals requesting the information needed to evaluate the problem. In this case, the circulation manager returns confirmation that the subscription is active and issues are being mailed as scheduled. When returning this information, the circulation manager marks the collaboration task assigned to him as complete.

The database administrator completes his collaboration task by providing the customer's account information—where the CSR finds that the address is incorrect. The CSR then creates a new collaboration task for the database administrator, asking for the customer's record to be updated with the correct information. The database administrator in turn delegates this collaboration task to one of his team members, who enters the correct information and marks the collaboration task as completed. The CSR now marks the parent task as completed.

About Collaboration Processes

Although collaboration processes are created and run only in the My webMethods Server environment, there are some aspects of the collaboration process that the Software AG Designer user needs to be aware of. A collaboration process is a unique type of process in the My webMethods Server environment and must not be confused with a standard webMethods business process (which can be created in Designer using the Process Development perspective).

A unique collaboration process is created any time a task user creates a collaboration task on the Collaboration tab of a parent task (the task must be enabled for collaboration). The collaboration process remains running until the parent task is completed, and any additional collaboration tasks that are created are added to the collaboration process. The collaboration process is owned by the user who created the first collaboration task, and the collaboration process owner has full access and privileges to the collaboration process and the collaboration tasks in it.

The key consideration concerning the collaboration process within Designer is the ability to specify sharing of comments and attachments among all child tasks in a collaboration process by setting the comments and attachments scope.

For more information, see:

- [“About Task Comments and Attachments Sharing” on page 38](#)
- [“About the Relationship Between Parent Tasks and Collaboration Tasks” on page 37](#)

About the Relationship Between Parent Tasks and Collaboration Tasks

When you create a new task type in Designer, it is not configured as either a parent or collaboration task. You have two opportunities to define this task parent-child relationship between two or more task types:

- **In Designer.** You can specify a parent task on the Overview tab of the task editor; this automatically designates the task you are working with as a collaboration task and makes collaboration-specific event actions available within both the collaboration and the parent tasks. A parent task can have one or more child collaboration tasks, but a collaboration task can have only one parent task. If you do not want to designate another task as the parent, you can specify the name of the task you are working with as the parent; this keeps the event actions limited to the current task.

Typically, both the parent and child task are contained in the same task application project, for ease of maintenance and so they can be published to the runtime at the same time. This type of parent task is best suited for use in an automated business process, where it can queue its child task(s) as described in [“Using Collaboration Tasks in a Process” on page 35](#).

- **In My webMethods.** On the Task Engine Administration page of My webMethods, you can specify one or more of the task types available in the runtime environment as a child collaboration task for a task type. This enables the user to be able to select from the specified collaboration task types when creating a new collaboration task on the Collaboration tab of the task they are working with (with proper permissions, and assuming the task is enabled for collaboration).

This type of parent-child relationship is best suited for use in a manual situation, where the user can queue collaboration task(s) as described in [“Using Collaboration Tasks Manually” on page 36](#).

When you define a parent-child relationship in Designer, you are able to configure both the parent and child tasks in whatever custom manner you need to address a complex human activity. In this case, you are creating and configuring the parent and child tasks with a specific relationship and specialized parent-child behavior.

The parent-child mechanism in My webMethods is more informal and enables you to define a parent-child relationship between any two task types available in the runtime environment. As these tasks have already been configured in Designer, the amount of functionality shared by the two tasks will be limited.

In both cases, the parent and child tasks run within a single collaboration process and task comments and attachments can be shared, as described in [“About Task Comments and Attachments Sharing” on page 38](#).

Note that it is possible for a single task type to be both a parent task and a child collaboration task at the same time. For example, suppose you have three task types, Task1, Task2, and Task3. Task2 can be a collaboration task for parent task Task1, and Task2 can also be a parent task for collaboration task Task3.

About Task Comments and Attachments Sharing

As part of a collaborative workflow environment, you can specify a scope that defines how comments and attachments are shared among other users and other tasks.

Note:

The sharing of attachments applies only to attachments on the **Comments** tab or to attachments added from an attachments list panel that has been added to Task Details portlet default view (see [“Adding Attachment Capability to a Task” on page 210](#)). Attachments on the **Content** tab are not subject to sharing.

The following scope selections are available for comments and attachments in all tasks:

- **Task Instance.** This is the default scope. Each individual task instance has its own unique attachments and comments (that is, they are not shared).
- **Process Instance.** All tasks within the same process model instance share their comments or attachments. This applies even when different task types are in use within the process model. When this scope is specified, you will see comments and attachments for all tasks started within that process model when you view the task comments or attachments of any task instance started from within the process model.
- **Process Instance AND Control Set.** All tasks within the same process model instance that use the same control set share their comments or attachments. This applies even when different task types are in use within the process model. When this scope is specified, tasks started within that process model must share a task control set to enable sharing of task comments or attachments. For more information about control sets, see [“Working with Task Control Sets” on page 148](#).
- **Collaboration Process.** All tasks within a single collaboration process instance share comments or attachments. This applies even when different task types are in use within the collaboration process. For more information about collaboration processes, see [“About Collaboration Processes” on page 36](#).

For information about setting task scope, see [“Sharing Attachments by Setting the Attachments Scope” on page 211](#) and [“Setting a Scope for Task Comments” on page 214](#).

5 Creating Tasks

■ About Creating Tasks	40
■ Accessing the Task Wizard	40
■ Creating a New Task with the Task Wizard	40
■ Creating a Collaboration Task	44
■ Creating a New Task from an Integration Server Document Type	45
■ Adding a Task to a Process	45

About Creating Tasks

A task can exist only as part of a task application project; for information on creating a task application project, see [“Creating a Task Application Project” on page 22](#).

To create a task, see [“Creating a New Task with the Task Wizard” on page 40](#).

Accessing the Task Wizard

Tasks can be created by opening the Task wizard in the following ways:

- On the UI Development perspective:
 - By starting the Task wizard with the **File > New > Task** menu command
 - By clicking the **New > Task** arrow button in the main toolbar.
 - By clicking the New button on the main toolbar, expanding **Software AG > Composite Applications**, and clicking **Task**.
- By clicking the **New > Other** menu command anywhere it is available, expanding **Software AG > Composite Applications**, and then clicking **Task**.
- By right-clicking an existing project in the Tasks folder of the Solutions view and clicking **New Task**.
- By right-clicking an IS document in the Package Navigator view and clicking **CAF > Generate New Task**.
- By right-clicking a task step within a process model and clicking **Create Implementation > New Task**.

Note:

If you expect the task to use IS documents to define input or output data, ensure that the documents have been created and are available.

Creating a New Task with the Task Wizard

Use this process to create a new task on the UI Development perspective. If you want to create a task directly from a task step in a process, see [“Adding a Task to a Process” on page 45](#).

➤ To create a new task with the Task wizard

1. Start the Task wizard as described in [“Accessing the Task Wizard” on page 40](#).

Note:

If you start the Task wizard by right-clicking an existing project in the Tasks folder of the Solutions view and clicking **New Task**, the Task wizard opens with the Task Information

page; in this case, skip to Step 4. Otherwise, the Task wizard opens with the Project Selection page as described in Step 2.

2. On the Project Selection page, accept the default project, select an existing project name from the drop-down list, or click the **New** button to create a new task application project. You have two options:
 - The **Enable advanced options in this wizard** check box is selected by default. Clear this check box if you want to click the **Finish** button and create the task with the default settings.
 - If you clear the **Enable advanced options in this wizard** check box, you have the option of selecting a check box to skip the Task wizard entirely for future task creation. In this case tasks are always created with the default settings.
3. If you have selected the **Enable advanced options in this wizard** option, click **Next**.
4. On the Task Information page, do the following:
 - a. Type a task type name for the task. The task type name cannot contain the following characters: < > ' ' & | \ / ;

Important:

If you create a task type that duplicates the name of an existing task type, *before publishing the task type, you must use the task editor to manually rename the task application root page* to make it unique among all tasks deployed to My webMethods Server; otherwise, the most recently deployed duplicate task type will overwrite the existing task application pages. This will also ensure that custom inbox page names will be unique. For information about modifying the task application root page, see [“Renaming a Task Root Page” on page 181](#).

- b. If you have not selected the task development preference, "Always use GUID for new Task ID," you must do one of the following:
 - Accept the default [ProjectName].[TaskTypeName] value. If you have created a duplicate task type name, this solution cannot be used; however, you can edit the text to create a unique value.
 - Type a unique task type identifier.
 - Click **Generate** to have Designer create a GUID for you.

For more information about the GUID preference, see [“Setting Task Development Preferences” on page 16](#); for more information about task type names, see [“About Task Names and Task Type IDs” on page 31](#).

- Select **HPSTRA enabled** to enable the indexing of the task data in an Elasticsearch store for high-performance searching. You can specify which business data fields to index in Elasticsearch on the **Business Data** tab. For more information about business data in HPSTRA-enabled tasks, see [“About HPSTRA-enabled Tasks and Business Data Fields” on page 89](#).

- Select **Voting enabled** to enable multiple runtime users to modify the task status through a collective decision. You can specify the voting strategy for the type on the **Overview** tab. For more information about task voting, see [“About Task Voting” on page 31](#) and [“Configuring Votable Tasks” on page 106](#).
- c. Select the check boxes to indicate the default portlets you want to create with the task and to specify your settings for those portlets:

Option	Description
Generate default Task View portlet	<p>Select this option to include a Task View portlet in your task. A task with no Task View portlet will display no user interface in My webMethods. The default user interface is described in “About the Task View (Task Details) Portlet” on page 52.</p> <p>You can clear this option if:</p> <ul style="list-style-type: none"> ■ You want to build your own custom interface without using any of the default functionality. ■ You are creating a task that will be accessed only through the Task Engine APIs and will not be accessed through a user interface. <p>This option is enabled by default and is controlled by the Task Development preference settings (see “Setting Task Development Preferences” on page 16).</p> <p>For information about creating portlets, see “Accessing the New Portlet Wizard” on page 66.</p>
Enable user routing	<p>Select this check box to enable a user to assign a task in the user's inbox to another user, group, or role on the task Data View tab. If you do not select this check box now, you can change this setting later, as described in “Modifying Client Options for a Task” on page 171.</p>
Auto-accept task upon modification	<p>Select this check box to cause a task to be automatically accepted by a user when the user modifies the task, and released immediately after the modification. This option serves as a temporary lock, allowing the user to make a modification to the task without explicitly accepting it. When this check box is not selected, the user must manually accept the task before updating it. You can change this setting later, as described in “Modifying Client Options for a Task” on page 171.</p>
These options enable you to specify how (or if) the task's portlets are generated for the task:	<p>Default. The default "out of the box" portlet is used, and the portlet is added to the Overview portlet with a "portlet include" statement. This the simplest approach, providing a "ready to go" user interface and a minimum number of portlets in the task. However, you are not able to customize the included portlets.</p>

Option	Description
	<p>Customizable. A separate portlet is generated within the task, using the same user interface as the default portlet, but you now have full access to the portlet and can modify it in the task editor to meet your specific needs.</p> <p>None. No portlet is created and the portlet contents are omitted from the task. The corresponding tab is hidden from the task page.</p>
These are the available portlets:	<p>Task Details Portlet. Creates the Details View tab in the task interface. The Details View provides operational information about the task, such as name, priority, status, assignment, creation date, and expiration date.</p> <p>Task Audit Portlet. Creates the Audit View tab in the task interface. The Audit View displays a table of all operations applied to the task, with details about each action, and the ability to roll back an action.</p> <p>Task Comments Portlet. Creates the Comment tab in the task interface. The Comments tab enables a user to enter, read, and modify comments, and add or delete attachments. For more information about comments and attachments, see “About Task Comments and Attachments Sharing” on page 38.</p> <p>Task Collaboration Portlet. Creates the Collaboration tab in the task interface. When a task is enabled for collaboration, the Collaboration tab enables a user to create and monitor a collaboration task. For more information about collaboration tasks, see “About Collaboration Tasks” on page 35.</p>
Generate default Task Start portlet	Select this check box to include a default Start portlet in the task. For more information about the Task Start portlet, see “About the Task Start Portlet” on page 56 . This option can be enabled by default depending on the Task Development preference settings (see “Setting Task Development Preferences” on page 16).
Generate default Task Inbox portlets	Select this check box to include the default Task Inbox Bar and Task Inbox Results portlets in the task. For more information about these portlets, see “About the Task Inbox Bar Portlet” on page 53 and “About the Task Inbox Results Portlet” on page 54 . This option can be enabled by default depending on the Task Development preference settings (see “Setting Task Development Preferences” on page 16).
Auto-accept task upon opening in task type inbox results	Select this check box to cause a task to be automatically accepted by the first user to open the task from the task type inbox. If a user opens the task from My Inbox or Task List Management, the user still must click Accept to accept the task. You can change this setting later, as described in “Modifying Client Options for a Task” on page 171 .

Option	Description
Select task search content provider	<p>Select the search content provider instead to use with the task from the following options:</p> <ul style="list-style-type: none"> ■ Default search provider ■ Task indexed search provider ■ HPSTRA search provider <p>For more information about task search providers, see “Task Search Overview” on page 158.</p>

5. Click **Finish** to create the task.

You can now configure the task as required. For example, you can add task business data, define task assignments and event actions, and create task control sets. For more information on configuring a task, see [“Configuring Tasks” on page 85](#).

Creating a Collaboration Task

Within Designer, you can define any available task type as a collaboration task. For more information about collaboration tasks, see [“About Collaboration Tasks” on page 35](#) and [“About the Relationship Between Parent Tasks and Collaboration Tasks” on page 37](#).

➤ To create a collaboration task

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. Click the **Overview** tab.
3. In the **Parent Task Type** list, select the task you want to use as the parent task.

Note:

The Parent Task Type list is populated with the names of all the task types available in the workspace. If no other tasks exist, the list is empty. In this case you must create another task to serve as the parent task. Only one task can be specified as the parent task. If you want to develop a "stand-alone" collaboration task (that is, a generic collaboration task that can be used as a child collaboration task with any other task), you can specify the current task name as the parent task. For example, if you are working with Task1, you can select Task1 from the list as the parent task.

4. Save your changes.

You can continue to configure the task as you would with any other task, as described in [“Configuring Tasks” on page 85](#). In particular, you may want to configure the following:

- **Configuring task event actions**—for more information, see [“Configuring Tasks” on page 85](#).

- **Task Scope**—for more information, see [“Working with Task Attachments” on page 209](#) and [“Task Comments Overview” on page 214](#).

Creating a New Task from an Integration Server Document Type

You can create a new task directly from an Integration Server document type; in this case, the contents of the document type are automatically added to the task as business data.

➤ To create a new task from an IS document type

1. In the Package Navigator view, locate the IS document type you want to work with.
2. Right-click the IS document type and click **CAF > Generate New Task**.
3. In the New Task wizard, define the task as described in [“Creating a New Task with the Task Wizard” on page 40](#).
4. When the new task opens in the task editor, you can view the business data as described in [“Viewing Business Data” on page 87](#).

For more information about working with business data, see [“Configuring Business Data \(Task Inputs and Outputs\)” on page 86](#).

Adding a Task to a Process

As you develop a process within Software AG Designer, you may want to add one or more tasks to the process to represent human activity. There are two methods for doing so:

- You can create a task step within the process and then implement a new task from the task step as described in [“Creating a New Task from a Task Step in a Process” on page 45](#).
- You can select an existing task and add it to the process, as described in [“Adding an Existing Task to a Process” on page 47](#).

If you want to create a new task on the My webMethods perspective, see [“Creating a New Task with the Task Wizard” on page 40](#).

Creating a New Task from a Task Step in a Process

Use this method when you have created a task step within a process in Designer, but have not yet implemented a task for that task step. If you want to create a task step using an existing task, see [“Adding an Existing Task to a Process” on page 47](#).

Note:

If you plan for the task to use document types to define input or output data, be sure to include those documents in the process model and connect them to the task. When this is done, the resulting task will automatically reference those document types.

➤ **To implement a new task from a task step in a process**

1. Add a new Activity step to the process.
2. Right-click the Activity step, click **Change Step Type**, and click **Task**.

Important:

Be sure the task step has a unique name before completing this procedure. When a new task step is added to a process, the step retains the default name of "Activity". To rename the task step, click it in the process editor design canvas and then modify the **Label** value in the Properties view. This value will be used to create the task type name when you implement the new task.

3. Right-click the task step and click **Create Implementation > New Task**. One of the following will occur:

- If you have specified the task development preference, "Automatic task creation using "Implement with New Task..." in Process Modeler," the task is created automatically using the default settings of the Task wizard; no wizard panels will appear. For more information about this preference, see ["Setting Task Development Preferences" on page 16](#).

In this case, the task is created within a task application project named (by default) [ProcessName]Tasks. For example, if the process is named "PO_Approval," the task application project is named "PO_ApprovalTasks." You can modify the default name; for information about doing so, see ["Setting Task Development Preferences" on page 16](#). The task type name of the new task implementation is set to the task step label displayed in the process.

Note:

The task type name must be unique. If Designer determines that the name of the task you are attempting to create is not unique, the Task wizard will open to enable you to type a unique task type name. Type a unique name in the **Task Type Name** box and click **Finish**. *Be sure to modify the task step label value as described in step 2 above to use the same name.*

- If you have not selected the automatic task creation preference, the Task wizard appears. For more information about the settings in the Task wizard, see ["Creating a New Task with the Task Wizard" on page 40](#).
4. After the new task is created, the task appears under the task application project in the Solutions view, as well as the Navigator view, and all other locations where tasks are displayed. You can continue to work with the task from those locations as described in ["Working with Task Portlets" on page 49](#).

You can now configure the task as required. For example, you can add task business data, define task assignments and event actions, and create task control sets. For more information on configuring a task, see ["Configuring Tasks" on page 85](#).

Adding an Existing Task to a Process

You can add an existing task to a process in one of two ways:

- By adding a task in the Solutions view directly to the process as a new step. For more information, see [“Adding a Task by Drag and Drop” on page 47](#).
- By creating a new task step and specifying an existing task. For more information, see [“Adding a Task by Selecting from a Task List” on page 47](#).

Adding a Task by Drag and Drop

Use this procedure to drag and drop an existing task into a process to create a task step.

➤ To add an existing task to a process by dragging and dropping

1. In the Solutions view, open the process you want to work with by double clicking it.
2. In the Solutions view, click the task you want to add and drag it to the process design canvas. After the task is added to the design canvas as a task step, note that you can open the task view in the task editor by double clicking the task step or by right-clicking the task step and then clicking **Open in Task Editor**.
3. Add any required documents or Java types to the process and connect them to the newly added task step.
4. Save the changes.

Adding a Task by Selecting from a Task List

Use this procedure to select an existing task for use with a task step in a process.

➤ To add an existing task to a process by selecting from a list of tasks

1. Add a new Activity step to the process.
2. Right-click the Activity step, click **Change Step Type**, and click **Task**.
3. Click anywhere on the process canvas, then click the new step.
4. In the Properties view, click the Implementation tab.
5. Click the browse button next to the Task Name box to view a list of all available tasks.
6. Select the task you want to use and click **OK**.

7. If control sets are available for the task, click the one you want to use in the **Task Control Set** box.
8. Save the process.

6 Working with Task Portlets

■ Task Portlet Overview	50
■ Viewing and Opening Task Portlets	63
■ Adding a New Portlet to a Task	64
■ Adding a New View to an Existing Task Portlet	71
■ Previewing a Task Portlet View in the Task Editor	73
■ Deleting Task Portlets and Portlet Views	74
■ Running Task Portlets and Views in My webMethods Server	75

Task Portlet Overview

Because the contents of a task are essentially specialized portlets, you can modify many of those portlet characteristics just as you would any other portlet. For example, you can:

- Add a task view to the portlet, as described in [“Accessing the New Portlet View Wizard” on page 72](#).
- Modify the task views presented to the user by adding, moving, or rearranging controls and other components on the view's design canvas, as described in [Example Modifying the Default Search Fields and Results](#).
- Link a functional privilege from the task application to a security (user) role for personalization, enabling/disabling a privilege, or for security purposes, as described in [“Binding a Task Functional Privilege to a User Interface Control” on page 121](#).
- Bind a security (user) role to a control, as described in [“Binding a Security Role to a User Interface Control” on page 83](#).
- Add your own Java methods.

For more information about general procedures and concepts for working with portlets, see *webMethods CAF and OpenUI Development Help*.

There are a number of portlet characteristics that are unique to tasks; you can modify these as described in the following topics:

- [“About Default Task Portlets” on page 50](#)
- [“About Customizable Task Portlets” on page 57](#)

About Default Task Portlets

When you create a task, you implement much of its functionality by defining portlets within the task. When you use the task wizard to create the task, you have the option of:

- Adding default portlets to the task.
- Adding customizable versions of the default portlets to the task.
- Omitting any of the selectable default portlets.

You can also create a task containing no portlets, if you want to build the task entirely on your own. When you create a task with the task wizard, a task view (Task Details) portlet is selected for creation by default (this behavior can be disabled in the Designer preferences). You can specify preferences for creating other portlets by default as well; for more information, see [“Setting Task Development Preferences” on page 16](#).

Note:

Default task portlets are created with JSF 1.x-based .view files, even if you create the project as a JSF 2.x task application. You can migrate these .view files to .xhtml files with JSF 2.x support.

For more information, see [“Migrating Task Projects and .view Files to JSF 2.x Format” on page 24.](#)

When you select the default portlets at creation time, the resulting task has many of the standard features that most developers would want to include in a task, and it can be immediately published to the run-time environment for viewing. You can then modify the portlets to your custom needs, instead of having to start with an empty portlet.

However, much of the portlet content is fixed as part of the default portlet configuration and cannot be modified. For example, if you want to modify the Task Details View portlet that creates the **Details View** tab in the task interface, that page is not available to you in the task editor. When you create a task with all of the available portlets selected in default mode, your task is populated with the following portlets:

- [taskName]InboxBar (see [“About the Task Inbox Bar Portlet” on page 53](#))
- [taskName]InboxResults (see [“About the Task Inbox Results Portlet” on page 54](#))
- [taskName]Overview (see [“About the Task Overview Portlet” on page 51](#))
- [taskName]Start (see [“About the Task Start Portlet” on page 56](#))
- [taskName]View (see [“About the Task View \(Task Details\) Portlet” on page 52](#))

By selecting the **Customizable** option for a portlet when creating a task, the internal pages are broken out and provided as separate portlets. In addition to the default portlets listed above, the following portlets are made available:

- [taskName]Audit (see [“About the Task Audit Portlet” on page 58](#))
- [taskName]Collab (see [“About the Task Collaboration Portlet” on page 60](#))
- [taskName]Comments (see [“About the Task Comments Portlet” on page 59](#))
- [taskName]Details (see [“About the Task Details View Portlet” on page 57](#))

You can also choose the **None** option for each available portlet, which excludes it from the task. You can also create an empty task containing no portlets by clearing all of the **Generate default...** options in the New Task wizard.

If you do choose not to add a portlet at creation time, you can add it later as described in [“Adding a New Portlet to a Task” on page 64.](#)

For more information about selecting portlet options, see [“Creating a New Task with the Task Wizard” on page 40.](#)

About the Task Overview Portlet

A Task Overview portlet is created for every new task by default when the **Generate default Task View** portlet option is selected in the New Task wizard. The Task Overview portlet serves as a container for providing a formatted message panel for displaying any JSF context messages, as well as the following portlets (through the use of portlet include controls):

- Task View portlet (creates the task **Data View** tab in the task interface)
- Task Details View portlet (creates the task **Details View** tab in the task interface; optional and may not be present)
- Task Audit portlet (creates the task **Audit View** tab in the task interface; optional and may not be present)
- Task Comments portlet (creates the task **Comments** tab in the task interface; optional and may not be present)
- Task Collaboration View portlet (creates the task **Collaboration** tab in the task interface; optional and may not be present)

You can add or remove components or further customize the default view to address your specific business needs.

About the Task View (Task Details) Portlet

The default task view portlet is identified as the Task Details portlet in the task editor, and contains the following components:

- A formatted message panel for displaying any JSF context messages.
- A **Task Data** panel that contains an empty property subgroup for you to add controls to.
- A **Task Info** panel that contains:
 - A **Toggle Optional Task Info** button that enables the user to view or hide the following task information:
 - **Name**—The name you want to assign to the task.
 - **Description**—The description you want to assign to the task.
 - **Priority**—The priority you want to assign to the task.
 - **Custom Task ID**—The custom task identification you want to assign to the task.
- A modal **Select Principals** dialog box (People Picker Dialog control) for selecting assignees.
- A **Task Voting Statistics** table - a table with a list of assignees and task voting details. At runtime, Task Engine displays the table only if the task is enabled for voting and the user has appropriate permissions to control the visibility of the table with the **Voting Details** button.

The following buttons are also included:

- **Accept**—Accepts the task for the current user.
- **Submit**—For My webMethods users, this updates all task information from the data fields on the page but does not complete the task (that is, the status is not changed).
- **Complete**—Updates all task information from the data fields on the page and applies a status of Completed to the task.

- **Return**—Discards any user changes and returns the user to the inbox.
- **Release**—Releases an accepted task, indicating it is no longer accepted by the current user. Visible only if the task is not marked for auto-acceptance and is accepted.
- **Assign to Users**—Enables the user to assign the task to one or more selected users, groups, or roles.
- **Voting Details**—Toggles the visibility of the task voting statistic table for votable task types. At runtime, Task Engine displays the button only if the task is enabled for voting.

You can add or remove components or further customize the default view to address your specific business needs.

About the Task Inbox Bar Portlet

The Task Inbox Bar portlet defines the user interface for the **Basic** tab portion of the Search For window in the task type inbox page in My webMethods. The controls in this portlet enable the user to specify the parameters for a task search. The default task inbox bar portlet contains the following components:

- A formatted message panel for displaying any JSF context messages. This panel provides a **Show Details** button as well as a **Close** button for reading and hiding message details.
- A **Task Data** panel that contains an empty property subgroup for you to add controls to.
- A **Task Info** panel that contains controls to enable specification of the following search parameters:
 - **Task ID**—A text entry field that enables you to search by task type identification number.
 - **Priority**—A drop-down list that enables you to search by task priority.
 - **Accepted Only**—A check box that limits the search to tasks accepted by the user.
- A **Date Ranges** panel that contains the following controls:
 - **Created**—A drop-down list that provides a list of time intervals (last 24 hours, previous week, etc.).
 - **Expiration**—A drop-down list that provides a list of time intervals (last 24 hours, previous week, etc.).
 - **Last Updated**—A drop-down list that provides a list of time intervals (last 24 hours, previous week, etc.).

The following buttons are also included

- **Go**—Starts the search with the specified parameters.
- **Save**—Saves the search parameters for reuse.
- **Clear**—Clears the displayed search parameters.

You can add or remove components or further customize the default view to address your specific business needs. For more information about creating a custom search bar portlet, see [“Working with Custom Task Inboxes” on page 215](#).

About the Task Inbox Results Portlet

The Task Inbox Results portlet contains two portlet views that define the elements of the user interface for the Inbox Results portion of the task type inbox page in My webMethods:

- The *default view* defines the overall presentation of the task search results (see [“About the Inbox Results Default View” on page 54](#)).
- The *edit view* defines the formatting controls available to the user for the inbox results. See [“About the Inbox Results Edit View” on page 55](#). The end user accesses the edit view in My webMethods by clicking the menu icon in the Inbox window, clicking **Properties**, and then clicking the **Preferences** tab.

You can add or remove components or further customize the default view to address your specific business needs. For more information about creating a custom results portlet, see [“Working with Custom Task Inboxes” on page 215](#).

About the Inbox Results Default View

The inbox results are displayed in My webMethods in tabular format, with sortable column heading elements. The Inbox Results provides a formatted message panel for displaying any JSF context messages and a custom script control.

The following column headings are available (the column headings are enabled for sorting):

- A **Select All** check box.
- An Accepted icon that indicates the tasks accepted by the user.
- A Delegated To icon that indicates the tasks that have been delegated to the user.
- A Delegated From icon that indicates the tasks delegated to others by the user.
- **Task ID**—The task identification number assigned to the task by the Task Engine.
- **Priority**—The priority of the task (None, Low, Medium, High, Critical).
- **Created By**—The user that created the task.
- **Modified By**—The user that last modified the task.
- **Created Date**—The date the task was created.
- **Expiration Date**—The expiration date of the task (if specified).
- **Last Updated Date**—The date the task was last modified.
- **Custom ID**—The custom ID of the task as typed by the user or defined by a process.
- **Task Type**—The type of task (that is, the task type name).

- **Assigned to**—The name of the users, groups, or roles that the task is assigned to.
- **Accepted By**—Name of user who last accepted the task.
- The following links are included

- **Subscriptions**—takes the user to the Task Subscriptions page.

Note:

Limitations apply to the use of this link in certain situations. For more information, see [“Limitations to Subscription Access” on page 188](#).

- **Scheduled Delegations**—Takes the user to the Scheduled Delegations page.

Note:

Limitations apply to the use of this link in certain situations. For more information, see [“Limitations to Scheduled Delegation Access” on page 198](#).

- A modal Select Users dialog box for delegating tasks.
- A modal dialog box for confirming the removal of task delegations.

The following buttons are included:

- **Refresh**—enables the user to refresh the Inbox Results table. This button is hidden by default when a Task Inbox Bar portlet is present, as running a search refreshes the Inbox Results table. When no Task Inbox Bar portlet is present, the button is displayed automatically.
- **Delegate**—enables the user to specify a user to whom the selected tasks will be delegated.
- **Remove Delegation**—enables the user remove the delegation from a selected task.
- **Export Table**—enables the user to export the Inbox Results table.

In addition, display fields are included to:

- Show the number of selected tasks, the number of displayed tasks, and the total number of tasks in the inbox.
- Show the current page number for task results, and enable the user to move to the Next or Previous page, or a specific page number.

You can add or remove components or further customize the default view to address your specific business needs. With the proper functional privileges, the user can customize the results display in the run-time environment. For more information, see the PDF publication *webMethods Task Engine User’s Guide*.

About the Inbox Results Edit View

You access the inbox results edit view in My webMethods by clicking Properties in the Task Results window toolbar menu. The **Preferences** tab in the resulting page is defined by the edit view in the default InboxResults portlet. The default edit view contains the following components:

- A formatted message panel for displaying any JSF context messages.

- A **Display Defaults** panel that contains the following controls:
 - **Number of Rows to Display**—Provides a list of display values, or Show All. The default value is 20.
 - **Sort By**—Provides a list of column headings. The default value is Task ID.
 - **Sort Order**—Specifies that the default sort order will be ascending or descending. The default value is ascending.
- A **Column Display** panel that enables the user to specify the columns that are displayed by moving column names into and out of the Selected list.

The following buttons are included:

- **Save**—applies the users specified settings and saves them for use until they are changed again.
- **Cancel**—discards any user changes and returns the user to the task inbox page.

You can add or remove components or further customize the default view to address your specific business needs.

About the Task Start Portlet

The Task Start portlet enables a user to start a new task on the Task Engine Administration page of My webMethods. Include a task start portlet in your task when you want the user to be able to start a task manually in the run-time environment, or remotely with the Task Engine API. For more information, see *webMethods Task Engine API and Service Reference*. If a task is used exclusively within a process (where it will be triggered only by the process), you do not need to include a task start portlet.

The user interface components of the task start portlet are displayed after the user starts the new task on the Task Engine Administration page of My webMethods. In addition, the task start portlet can be placed on any custom page to enable a non-administrator user to start tasks (you must also grant the user access to this page). The default task start portlet contains the following components:

- A data flow that calls a modal dialog box that appears when the task is started, containing:
 - A formatted message panel for displaying any JSF context messages; for example, "The new task has been successfully started."
 - A show/hide details button to display the task ID number.
 - A link to open the new task.
 - An **OK** button to close the dialog box.
- A **New Task** panel that contains:
 - An empty property subgroup where you can add controls.
 - A **Toggle Optional Task Info** button that enables the user to view or hide the following task information:

- **Name**—The name you want to assign to the task.
- **Description**—The description you want to assign to the task.
- **Priority**—The priority you want to assign to the task.
- **Custom Task ID**—The custom task identification you want to assign to the task.

The following buttons are also included:

- **Start**—Starts a new instance of selected task type.
- **Return**—Closes the Start page and returns the user to the Task Engine Administration page.

You can add or remove components or further customize the default view to address your specific business needs.

Note:

With the Task Update wizard, you can add standard attachment support to the Start portlet. Using attachment support in the Start portlet, you can queue a standalone task in the runtime and specify an attachment at start time to test attachment behavior and network connections. Unless you want to use this test capability, you would normally not add attachment support to a Start portlet.

About Customizable Task Portlets

If you choose to add portlets to your task by selecting the default mode when you create the task, some page views are not available to you for customization. These views are made available by selecting the **Customizable** option for each portlet when you create the task.

Note:

Customizable task portlets are created with JSF 1.x-based .view files, even if you have created or migrated the project as a JSF 2.x task application. You can migrate these .view files to .xhtml files with JSF 2.x support. For more information, see “[Migrating Task Projects and .view Files to JSF 2.x Format](#)” on page 24.

When the **Customizable** option is selected, the following portlets are available in the task:

- “[About the Task Details View Portlet](#)” on page 57
- “[About the Task Audit Portlet](#)” on page 58
- “[About the Task Comments Portlet](#)” on page 59
- “[About the Task Collaboration Portlet](#)” on page 60

About the Task Details View Portlet

The Task Details View portlet provides the **Details View** tab in the task user interface, which offers a view of all the key information for a task. When added to a task in default mode, this portlet is included in the Task Overview portlet and is not available for customization. When the

Customizable option is selected for this portlet during task creation, the portlet is exposed as [taskName]Details, with these components (all fields are read-only except as noted):

- **Task Type**—The type of task (that is, the task type name).
- **Task ID**—The task identification number assigned to the task by the Task Engine.
- **Name**—The name of the task as typed by the task creator (editable).
- **Description**—The description of the task as typed by the task creator (editable).
- **Priority**—The priority of the task (None, Low, Medium, High, Critical) (editable).
- **Status**—The current status of the task.
- **Accepted By**—The names of users who accepted the task.
- **Accepted Date**—The date when the task was last accepted.
- **Assigned to**—The names of the users, groups, or roles that the task is assigned to.
- **Delegation**—The names of users to whom the task has been delegated.
- **Created Date**—The date and time the task was created.
- **Created By**—The user that created the task.
- **Last Updated Date**—The date and time the task was last modified.
- **Last Updated By**—The user that modified the task.
- **Expiration Date**—The expiration date of the task as typed by the task creator (editable).
- **Custom ID**—The custom ID of the task as typed by the user or defined by a process (editable).

The following buttons are also included:

- **Apply**—Updates all task information from the data fields on the page but does not complete the task (that is, the status is not changed and additional work can be performed on the task).
- **Return**—Returns the user to the inbox.

About the Task Audit Portlet

The task audit portlet provides the **Audit View** tab in the task user interface, which presents a tabular audit log of all of the operations performed on a task. When added to a task in default mode, this portlet is included in the Task Overview portlet and is not available for customization. When the **Customizable** option is selected for this portlet during task creation, the portlet is exposed as Task_customAudit, and includes the components, listed in the following table:

Column	Description
ID	A unique identifier number for each audited operation.

Column	Description
Operation	The type of operation performed on the task, such as "Task Updated" or "Routing Rule Fired." Some operations appear as a link that enable you to view additional information about the operation.
Created Date	The date and time the operation was applied to the task.
Source	The actor that applied the operation to the task; this can be a user, or a mechanism, such as a routing rule. Some entries appear as a link that enable you to view additional information about the source.
Delegated By	Displays the name of any user that delegates a task. Applies only to delegation operations.
Roll Back	When enabled, the user can click this icon to roll back, or "undo" the associated operation. If roll back is not possible, the icon is disabled.

The tab also provides a **Return** button to take the user back to the inbox page (this button is hidden in the interface by default).

About the Task Comments Portlet

The task comments portlet provides the **Comments** tab in the task user interface, which enables users to add, modify, and delete comments and attachments in a task. When added to a task in default mode, this portlet is included in the Task Overview portlet and is not available for customization. When the **Customizable** option is selected for this portlet during task creation, the portlet is exposed as [taskName]Comments. This portlet contains two views, the Default view and the Add-edit view. For more information, see:

- [“About the Task Comments Default View” on page 59](#)
- [“About the Task Comments Add-edit View” on page 60](#)

About the Task Comments Default View

The task comments default view is the initial view seen by the user when the **Comments** tab is opened. The following buttons are included:

- **Add Comment**—Opens the Add-edit view as the New Comment panel (see [“About the Task Comments Add-edit View” on page 60](#)).
- **Refresh**—Updates the contents of the **Comments** tab.
- **Return**—Returns the user to the inbox.

In addition, the **Comments** tab includes a table that contains the following:

- Comments entered by task users, including the user name, comment text, a time and date stamp.
- Attachments added by task users. Each attachment is represented by a link to the file.

- An Update link opens the Add-edit view as the Edit Comment panel (see [“About the Task Comments Add-edit View” on page 60](#)).
- A Delete link removes the associated comment and attachment (if present). No confirmation prompt is provided.

About the Task Comments Add-edit View

The Add-edit view is presented to the user as either the New Comment panel or the Edit Comment panel, depending on where the view is called from.

- The New Comment panel opens from the **Add Comment** button on the default view.
- The Edit Comment panel opens from the Update link on the default view.

The Add-edit view contains the following boxes:

- **Comment Text**—Free form text area for entering comments.
- **Attachments**—Display box showing the name of the file selected for attachment by using the associated **Browse** button.
- A Delete link removes the associated comment and attachment (if present). No confirmation prompt is provided.

As well as the following buttons:

- **Browse** button—Opens a standard file system browsing dialog box, enabling the user to select any file available through the local operating system.
- **Attach**—Marks the selected file for attachment to the task.
- **Create**—Available on the New Comment panel only. Adds any comment or attachment to the task.
- **Update**—Available on the Edit Comment panel only (replaces the **Create** button after a comment is added). Applies any changes to the comment or attachment.
- **Cancel**—Closes the New Comments panel without saving any changes.

About the Task Collaboration Portlet

The task collaboration portlet provides the **Collaboration** tab in the task user interface, which enables users to create and manage collaboration tasks. When added to a task in default mode, this portlet is included in the Task Overview portlet and is not available for customization. When the **Customizable** option is selected for this portlet during task creation, the portlet is exposed as [taskName]Collab. This portlet contains two views, the Default view and the Createnew view. For more information, see:

- [“About the Task Collaboration Default View” on page 61](#)
- [“About the Task Collaboration Createnew View” on page 62](#)

About the Task Collaboration Default View

The task collaboration default view is presented to the user when the user opens the **Collaboration** tab. If no collaboration tasks exist, the only elements displayed are a message to that effect, and the **Create Child Task** button. The view consists of:

- A formatted message panel for displaying any JSF context messages.
- A Process Information panel that displays the following information about the collaboration process associated with the task (the collaboration process is not created until the first collaboration task is completed):
 - **Name**—name of the collaboration process (editable).
 - **Created By**—name of the user that created the first collaboration task, thereby creating the collaboration process.
 - **Last Modified By**—name of the user who last modified the process or any of its contents.
 - **Created Date**—date and time the collaboration process was created.
 - **Last Modified Date**—date and time the collaboration process was last modified. This field also displays a duration value showing the total time the collaboration process has been running.
 - **Status**—returns the user to the inbox.

The following buttons are provided on the Process Information panel:

- **Open Details**—opens the Process Detail page which displays a graphic representation of the collaboration process, as well as tabbed information displays of Process Info, Step Info, Task Info, Task Audit, and Task Comments.
- **Delete**—enables the user to delete the collaboration process and all of its collaboration tasks.
- **Update**—refreshes the process information display.
- **Return**—returns the user to the inbox.
- A modal Select Principals dialog box (People Picker Dialog control).
- A modal process delete confirmation dialog box.
- A modal delete collaboration task confirmation modal dialog box.
- A modal status change dialog box that enables the user to set the status of a collaboration task.
- A search results tree that displays information about the parent task and all active collaboration tasks, arranged in row and column format.

The following buttons are provided:

- **Create Child Task**—Opens the New Collaboration Task dialog box.

- **Assign To**—enables the collaboration process owner to assign a selected collaboration task to one or more users, groups, or roles.
- **Set Status**—enables the collaboration process owner to set the status of a selected collaboration task.
- **Delete**—enables the collaboration process owner to delete a selected collaboration task.
- **Return**—returns the user to the inbox.

About the Task Collaboration Createnew View

The task collaboration createnew view is presented to the user as the New Collaboration Task panel when the user clicks the **New Collaboration Task** button on the **Collaboration** tab. The view consists of the following:

- A formatted message panel for displaying any JSF context messages.
- **Collaboration Task Type**—a list of all the available task types that can be used to create a collaboration task, as assigned on the Task Engine Administration page.
- **Name**—The name you want to assign to the collaboration task.
- **Description**—The description you want to assign to the task.
- **Priority**—The priority you want to assign to the task.
- **Expiration Date**—The date on which the task will be placed in expired status, as selected by an included calendar control.
- **Time**—A specific time of day on which the expiration will take place.
- **Assignees**—A list of the users, groups, and roles to which the collaboration task has been assigned.
- **Queue Task Immediately**—A check box option that controls whether the task is started immediately upon creation or is deferred for later starting. If this check box is cleared, the collaboration task is created with a status of New, and it appears in the collaboration task list and in the Task List Management results, but will not appear in the assigned user's inbox or the task type inbox. To queue the collaboration task (causing it to appear in the assignee's inbox), set the status for the collaboration task to Active from the collaboration task list or from the Task List Management results.

The following buttons are also included

- **Pick**—Opens the included modal Select Users dialog box for assigning the task to one or more users.
- **Create New**—Creates the new collaboration task with the specified values.
- **Cancel**—Discards all information and returns the user to the **Collaboration** tab.

About Modifying the Default Task Portlets

You can modify the default task portlets just as you would modify any other portlet created in Software AG Designer. You can add or remove controls, change the layout, add functionality, and so on. For information about modifying custom task inboxes, see [“Working with Custom Task Inboxes” on page 215](#)

Note that if you choose to add portlets to your task by selecting the default mode when you create the task, some page views are not available to you for customization. These views are made available by selecting the **Customizable** option for each portlet when you create the task. For more information, see [“Creating a New Task with the Task Wizard” on page 40](#) and [“About Customizable Task Portlets” on page 57](#).

Note:

Task portlets are created with JSF 1.x-based .view files, even if you create the project as a JSF 2.x task application. You can migrate these .view files to .html files with JSF 2.x support. For more information, see [“Migrating Task Projects and .view Files to JSF 2.x Format” on page 24](#).

For more information about working with portlets in general, see *webMethods CAF and OpenUI Development Help*.

Viewing and Opening Task Portlets

You can view and open any of the portlets available in a task.

➤ To view and open task portlets

1. In the Solutions view, expand either **Tasks** or **User Interfaces**
 - In the **Tasks** list, expand the task project that contains the task you want to work with. Then expand the task to view the portlets within it.
 - In the **User Interfaces** list, expand the task project to view the portlets within it.
2. Double-click any task portlet (for example, *TaskStart or *TaskView). The portlet editor opens.
3. Click **Configuration** to view all of the task portlets contained in the selected task application project. From this location you can specify the following portlet attributes:
 - Initialization parameters
 - Supports
 - Preferences
 - Security roles

In addition, you can define custom portlet modes, custom window states, and user attributes to be associated with the project's portlets.

Adding a New Portlet to a Task

There are two typical scenarios for adding a portlet to a task:

- When you create a task in the Task wizard, you specify the creation of several default portlets within the task. If you choose not add a particular portlet at creation time but want to add it later, you can add from a list of default portlets after you create the task.
- After you create the task, you may want to replace the default portlet with a customized portlet.

Tip:

You can customize the existing default portlet in the task editor, or create an entirely new portlet to replace the portlet. For example, if you created the task with the default settings in the task wizard, the audit, comment, and collaboration portlets are not available in the Solutions view and cannot be customized. When you create a new portlet with the procedure below, the existing portlet remains in the task, but it is replaced by the new portlet in the task definition, making it immediately available for publication.

Many of the procedures for creating and defining a task portlet are very similar to the procedures for developing any other CAF portlet. For more information about creating and working with portlets in general, see *webMethods CAF and OpenUI Development Help*.

Note:

When you add a task portlet to a task, the task portlet is created with JSF 1.x-based .view files, even if you create the project as a JSF 2.x task application. You can migrate these .view files to .xhtml files with JSF 2.x support. For more information, see [“Migrating Task Projects and .view Files to JSF 2.x Format” on page 24](#).

There are certain specific procedures that pertain to portlets within a task. For a description of the available task portlet type templates, see [“About the Task Portlet Type Templates” on page 67](#).

➤ To add a new portlet to a task

1. Open the portlet wizard as described in [“Accessing the New Portlet Wizard” on page 66](#).
2. On the Project Selection page, specify a project by doing one of the following:
 - Accept the predefined contents of the **Project** box.
 - Click a task application project in the **Project** box.
 - Click **New** and create a new task application project.
3. Click a task portlet type template in the **Portlet Type** box.

Important:

If you create a new portlet in a task that contains an existing portlet of the same name, the new portlet will automatically replace the existing one when the task application is published. The original portlet remains in the task but is not used.

4. When a task portlet type template is selected, additional controls appear on the page.
 - For a list of available portlets, see [“About the Task Portlet Type Templates” on page 67](#).
 - For information about the individual portlets that are available, see [“About Default Task Portlets” on page 50](#) and [“About Customizable Task Portlets” on page 57](#).
5. Specify the task that will contain the portlet by doing one of the following:
 - Accept the predefined contents of the **Task** box.
 - Click a task in the **Task** box.
 - Click **New** and create a new task.
6. Select your settings for the following check boxes (some options do not apply to all type templates):

Value	Description
Auto-accept task upon modification	Applies to task data view portlet only. Select this check box to cause a task to be automatically accepted by a user when the user modifies the task, and released immediately after the modification. This option serves as a temporary lock, allowing the user to make a modification to the task without explicitly accepting it. When this check box is not selected, the user must manually accept the task before updating it. You can modify this setting later, as described in “Working with Task Client Options” on page 169 .
Display Style	Applies to search results portlet only. Enables you to define the search results display as table view, tree view, or toggle between table and tree view.
Enable user routing	Applies to task data view portlet only. Select this check box to provide the ability for a user to assign a task in the user's inbox to another user, group, or role. If you do not select this check box now, you can still add this functionality later by adding the appropriate controls and functionality to the task portlet.
Include Task Data View	Applies to task overview portlet only. Select this check box to add a task data view PORTLET INCLUDE statement to the Overview portlet. The task data view portlet appears when the user clicks the Data tab in the task interface.
Note: The generated Overview portlet will still contain a Data tab even if you clear this check box. To remove the Data tab, open the view and select the Toggle Tabs control. In the Properties view, click the Choices tab and delete the TaskDataOption entry.	

Value	Description
Use task indexed search content provider	Select this preference to create the portlet with an indexed search content provider instead of a standard (full) search content provider; for more information, see “About Indexed Business Data Fields” on page 89 . This option can be enabled by default depending on the Task Development preference settings (see “Setting Task Development Preferences” on page 16 For more information about indexed search content providers, see “About Indexed Business Data Fields” on page 89 .
Use AJAX controls	Select this check box to enable the use of AJAX (Asynchronous JavaScript And XML) controls in the portlet.

7. From this point on, you can click **Next** to specify custom settings for the portlet, or click **Finish** to create the portlet with remaining default settings. The remaining pages in the wizard provide standard portlet configuration options.
8. Click **Finish** or click **Next** to continue configuration (see the **Important** note below if you click **Finish**).
9. On the Create Portlet page, specify a unique portlet name, and a portlet title. Accept the default values for the remaining values on the page, or specify new values. Click **Next**.
10. On the Create Portlet page, specify any text you want to appear at deployment time. Click **Next**.
11. On the Create Portlet page, specify the portlet class values you want to apply, or accept the default values. Click **Next**.
12. On the Create Portlet page, specify any resource bundle values you want to apply, or accept the default values.
13. When you click **Finish**, the new portlet is created within the specified task, and the default view for the portlet opens in the portlet view editor.

Important:

If you are replacing or adding to any of the portlets that are referenced in the Portlet Include controls in the overview portlet, be sure to open the overview portlet in the task editor and modify the appropriate Portlet Include control to reference the newly added portlet. If the new portlet is not already included in the overview portlet references, add a new Portlet Include control with a reference to the new portlet.

Accessing the New Portlet Wizard

You can create a new portlet in a task in the following ways:

- On the UI Development perspective:
 - By clicking **File > New > Portlet**.
 - By clicking the **New > Portlet** arrow button in the main toolbar.
 - By clicking the New button on the main toolbar and clicking **Software AG > Composite Applications > Portlet**.
- By clicking the **New > Other** command anywhere it is available and clicking **Software AG > Composite Applications > Portlet**.
- By right-clicking an existing project or task in the Tasks folder of the Solutions view and clicking **New Portlet**.

All of these methods start the portlet wizard and enable you to create a new portlet. In those cases where the project is known the task application project and task are preselected. In other instances, you must specify these values.

For specific procedures, see [“Adding a New Portlet to a Task” on page 64](#).

About the Task Portlet Type Templates

When you add a new portlet to a task, you have the ability to select from the following portlet type templates included in Software AG Designer:

Portlet Type	Description
Generic	As the name indicates, use this template to create a generic portlet that you can customize into any function. Typically, you would select the generic template when none of the other templates address your need and you intend to design your own custom portlet.
Search Bar	Use this template to add a basic search parameters portlet that conforms to the common search framework of My webMethods.
Search Results	Use this template to add a basic search results portlet that conforms to the common search framework of My webMethods. You can specify if you want a table view of the results, a tree view, or the ability to toggle between table and tree view.
Search Bar + Results	Use this template to add a portlet that provides both the search bar and the search results, requiring no portlet-to-portlet wiring. The portlet conforms to the common search framework of My webMethods. You can specify if you want a table view of the results, a tree view, or the ability to toggle between table and tree view.
Task Overview	Use this template to add a new task overview portlet (see “About the Task Overview Portlet” on page 51). This portlet implements the basic tab structure and embeds the other portlets used in the task.

Portlet Type	Description
Task Data View	Use this template to add a new task data view portlet (see “About the Task View (Task Details) Portlet” on page 52).
Task Details View	Use this template to add a new task details view portlet. For more information, see “About the Task Details View Portlet” on page 57.
Task Audit View	Use this template to add a new task audit view portlet. or more information, see “About the Task Audit Portlet” on page 58.
Task Start	Use this template to add a default start portlet (the start portlet enables the task to be started manually in the My webMethods run-time environment). or more information, see “About the Task Start Portlet” on page 56.
Task Inbox Bar	Use this template to add the default task inbox search parameters portlet. This template is equivalent to the default TaskInboxBar portlet that is added to the task when the task is created (if you select the Generate Default Task Inbox Portlets option at creation time). See “About the Task Inbox Bar Portlet” on page 53.
Task Inbox Results	Use this template to add the default task inbox results portlet. This template is equivalent to the default TaskInboxResults portlet that is added to the task when the task is created (if you select the Generate Default Task Inbox Portlets option at creation time). See “About the Task Inbox Results Portlet” on page 54.
Task Comment View	Use this template to add a new task comments view portlet. For more information, see “About the Task Comments Portlet” on page 59.
Task Collaboration View	Use this template to add a new task collaboration view portlet. For more information, see “About the Task Collaboration Portlet” on page 60.

Important:

In some situations, if the task you are working with contains more than one Task View or Task Start portlet, you may need to modify the view definition for the task.

Modifying the Task View Definition

You modify the task view definition only when you want to revert to an existing task overview portlet from the overview portlet currently in use. Otherwise, there is no need to modify the task view definition manually. The overview portlet is a "top-level" portlet and contains Portlet Include controls that refer to the other portlets used in the task.

➤ To modify the task view definition

1. In the Solutions view, open the task in the task editor by double-clicking the task name.

2. Click the **taskDefinition.xml** tab.
3. Locate the `wm_xt_task_definition` element. You can edit the text directly in the editor pane, or you can click the `wm_xt_task_definition` element and do the editing in the Properties view.
4. To modify the setting for the task view portlet, locate the attribute `view_task_portlet_uri`. The definition for this element contains a comma separated list of identifiers for portlets that are used to view task details. For example:

```
/portlet/project_1___newtask1view,/portlet/project_1___newtask1taskviewapproval
```

Modify the definition so that the task view portlet you want to use is the first entry in the definition. For example, to change the above definition to use the approval view portlet:

```
/portlet/project_1___newtask1taskviewapproval,/portlet/project_1___newtask1view
```

Software AG Designer will always use the first view portlet in the definition.

5. Save your modifications.

You can view the results on the My webMethods Server associated with the task application project. For more information, see [“Running a Task Portlet” on page 76](#). Otherwise, to view the results in the My webMethods Server run time, you must re-publish the task application. For more information, see [“Publishing a Task” on page 202](#).

About Modifying the Task Details Page Contents

By default, each new task is created with a task overview portlet that implements the task Details page. This page has the ability to provide the following tabs, depending on your selections when you create the task with the task wizard:

- **Data**
- **Details**
- **Audit**
- **Comments**
- **Collaboration**

Each link on the Toggle Tabs control in this portlet is related to a Portlet Include control that references the related portlet. If you want to create and specify a custom view for any of these tabs, you must either modify the existing portlet that contains that view, or, add a new portlet, customize it to meet your needs, and then modify the appropriate Portlet Include control to reference the new portlet.

Some important points to consider:

- If you created the task with the default settings in the task wizard, the audit, comments, and collaboration portlets are not available to you in the Solutions view, and so cannot be

customized. However, you can add a new portlet for any of these portlet types as described in [“Adding a New Portlet to a Task” on page 64](#) and customize that portlet.

- If any of the default tabs are not wanted, you can delete the link and the associated Portlet Include control from the overview portlet.
- After you are satisfied with your task interface, be sure to delete any unused portlets.

Modifying the Task Start Definition

If you add a new task start portlet in a task that contains an existing task start portlet, the new portlet will automatically replace the existing one. However, you must modify the task start definition if you want to revert to a previous task start portlet contained in the task.

➤ To modify the task start definition

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. On the Overview page, make a note of the task type ID.
3. In the Navigator view, navigate to the taskStart.xml file as follows:

[taskApplication] > WebContent > WEB-INF > tasks > [taskTypeID] > taskStart.xml

4. Double-click the taskStart.xml file to open it in the editor and click the **Source** tab.
5. This XML file defines the task start portlet usage in the following elements:

```
<projectName__portletName name="Display Name">
```

...

```
</projectName__portletName>
```

where:

projectName = the name of the task application project (an alias may be included)

portletName = the name of the portlet used as the task start portlet

Display Name = the name that will be displayed in the design-time and run-time interfaces.

For example:

```
<myportletapplication__mynewtaskstart name="Task Start"
alias="taskTypeID.task.app.start.portlet"> <layout name="titlebar"
value="false"/> <layout name="border" value="false"/>
</myportletapplication__mynewtaskstart>
```

6. To redefine the task start portlet, do the following:
 - Substitute the new portlet name for the existing name.

- Add or modify the portlet alias if desired.
- Modify the display name for the portlet if desired.

7. Save your modifications.

You can view the results on the My webMethods Server associated with the task application project. For more information, see [“Running a Task Portlet” on page 76](#). Otherwise, to view the results in the My webMethods Server run time, you must re-publish the task application. For more information, see [“Publishing a Task” on page 202](#).

Adding a New View to an Existing Task Portlet

After you create a task, you may determine that one or more additional views are needed in a portlet. For example, you may want to add a help view containing instructions for the task user.

Note:

When you add a new view, it is created as a JSF 1.x-based .view file, even if you have created or migrated the project as a JSF 2.x task application. You can migrate these .view files to .xhtml files with JSF 2.x support. For more information, see [“Migrating Task Projects and .view Files to JSF 2.x Format” on page 24](#).

➤ To add a new view to a task portlet

1. Open the Portlet View wizard as described in [“Accessing the New Portlet View Wizard” on page 72](#).
2. If you started the Portlet View wizard by right-clicking an existing portlet in the Tasks or User Interfaces folder of the Solutions view and clicking **New Portlet View**, the parent folder will be pre-selected and you can proceed to Step 3. Otherwise, do the following:
 - a. On the JSF Portlet View File page, expand the project that contains the portlet you want to work with.
 - b. Expand the **WebContent** folder and click the folder for the portlet you want to add a view to. For example, for a portlet named "MyTaskView," the folder name will also be named MyTaskView
3. Type a unique file name for the new view in the File name box. The file name must end with the extension '.view'. For example, `view_name.view`.
4. Click **Next**.
5. Specify the following:

Value	Description
For Portlet	Select a task portlet from the drop-down list.
For Portlet Mode	Click the portlet mode you want to apply. The value specified here defines the basic functionality of the portlet. Click None if this view is not a starting view for a corresponding portlet mode, or click View , Edit , or Help .
MIME Type	Enables you to specify the MIME type for View, Edit, and Help views.
Template	<p>Enables you to select from a list of view templates:</p> <ul style="list-style-type: none">■ Empty■ Form■ Search Bar■ Task Notification <p>By selecting one of these, you populate the view with a predefined layout and set of controls based on the template selected. If you want to create your own layout and controls, select Empty. For more information about the available templates, see “About the Portlet View Templates” on page 73.</p>

6. Click **Next**.
7. On the Managed Bean page, accept the default values or type a Managed Bean Name and click a different Managed Bean Scope. Click **Next**.
8. On the Java Type page, accept the default values or specify a different package and name. If you want to add interfaces to the view, click **Add**.
9. Click **Finish**.

The view is added to the specified task portlet and is opened in the portlet view editor. You can preview the new portlet view as you make changes to it; for more information, see:

- [“Previewing a Task Portlet View in the Task Editor” on page 73](#)
- [“Running a Task Portlet View” on page 77](#)

Accessing the New Portlet View Wizard

You can create a new view in a portlet by opening the New Portlet View wizard in the following ways:

- In the UI Development perspective by clicking the New button on the main toolbar and clicking **Software AG > Composite Applications > Portlet View**.

- By clicking the **New > Other** command anywhere it is available and clicking **Software AG > Composite Applications > Portlet View**.
- By right-clicking an existing portlet in the Tasks or User Interfaces folder of the Solutions view and clicking **New Portlet View**.

All of these methods start the portlet view wizard and enable you to create a new portlet view. In those cases where the project is known the task application project and task are preselected. In other instances, you must specify these values.

For specific procedures, see [“Adding a New View to an Existing Task Portlet” on page 71](#).

After you create a new view, you can use the JSF Faces Configuration editor to define how the run-time user can navigate to the view. For more information, see [“Working with Tasks in the Faces Configuration Editor” on page 231](#).

About the Portlet View Templates

When you add a new view to a portlet, you have the ability to select from the following templates included in Software AG Designer:

Template Name	Description
Empty	As the name indicates, use this template to create an empty view that you can customize into any function. Typically, you would select the empty template when none of the other templates address your need and you intend to design your own custom view.
Form	Use this template to add a basic form view that you can customize further. The view is created with a JSF command form and a formatted JSF context message display.
Search Bar	Use this template to add a basic search parameters view. The view is created with a formatted JSF context message display and a search bar containing a refine panel.
Task Notification	This template is used internally when a new task notification is created and should not be used directly by the user when creating a new portlet view.

Important:

If you are adding a view to a task view or start portlet that is not a default portlet, you may need to make additional modifications to the task to display the new view.

Previewing a Task Portlet View in the Task Editor

You can preview a portlet view in the Software AG Designer task editor at any time. A preview of a portlet view displays all of the subgroups, fields, controls, and other components of the view as they will appear in the run-time environment. This is non-editable, visual representation of the interface; it is not programmatically functional. For example:

- All included components are displayed, regardless of their hierarchy in the run-time environment. For example, you may include a secondary dialog box that would normally appear only when a button is clicked by the user. In the preview, the secondary dialog box is displayed with all of the other interface components.
- Buttons may be clicked but no actions will take place.
- Text may be typed in editable fields but will not be recognized

Note:

You must have the My webMethods Server Support option installed in Designer to use the preview tab. If you have not already done so, shut down all webMethods applications, run the Software AG Installer, and select **Designer > Composite Applications > My webMethods Server Support** to install the preview functionality.

> To preview a task portlet view in the task editor

1. In the Solutions view, expand either **Tasks** or **User Interfaces**.
 - In the **Tasks** list, expand the task project that contains the task you want to work with. Then expand the task to view the portlets within it.
 - In the **User Interfaces** list, expand the task project to view the portlets within it.
2. Locate and double-click any task portlet view (for example, the default view of the TaskStart or TaskView portlet). The portlet view editor opens. The **Design** tab displays the design canvas of the view editor, where you can add, remove, and modify interface components.
3. Click the **Preview** tab to view a non-editable display of the view as it will appear in the run-time environment (subject to the conditions outlined above).
4. To modify the interface, return to the **Design** tab. After modifying the interface, click the **Preview** tab to see the results.

Deleting Task Portlets and Portlet Views

Task portlets and portlet views can be deleted in Software AG Designer; deleted items cannot be recovered.

> To delete a task portlet or portlet view

1. Locate the task you want to work with in the Solutions view.
2. Expand the task to locate the portlet or portlet view you want to delete.
3. Right-click the item for deletion and click **Delete**.

You may also want to republish the task to My webMethods if the task has been previously published to My webMethods Server.

Running Task Portlets and Views in My webMethods Server

With the proper configuration, you can publish and run task portlets, task views, and task pages on My webMethods Server, accessed from within Designer. This enables you to see and interact with all of the subgroups, fields, controls, and other components of the portlet or view as they appear in the run-time environment.

For more information, see the following topics:

- [“Defining a Server for a Task Application Project” on page 75](#)
- [“Running a Task Portlet” on page 76](#)
- [“Running a Task Portlet View” on page 77](#)
- [“Running a Task Page” on page 78](#)

Defining a Server for a Task Application Project

Before you can run task portlets and views in My webMethods Server, your environment must be configured to support connectivity between My webMethods Server and Designer. In many development environments, it is common for My webMethods Server and Designer to be installed on the same system.

For this reason, a default installation of Designer and My webMethods Server results in this configuration:

- My webMethods Server is configured to run on port 8585, resulting in a URL of `http://localhost:8585`.
- Designer is configured to use this URL as the default My webMethods Server instance.

In this configuration, when you start My webMethods Server and Designer, the first time you access My webMethods Server, you will be asked to enter your credentials for logging in. The default user account is `sysadmin`.

Designer uses the `localhost:8585` URL as the default server instance for the task application projects you create, meaning that it is the target server when you:

- Publish a task application project.
- Execute the **Run on Server** command.

Note:

If a task application project does not have a default server, the task portlets, views, and pages within it cannot be run.

➤ **To define a My webMethods Server instance at a location that is different than `localhost:8585`**

1. Define the server location in the Servers view, as described in the *Eclipse Web Tools Platform User Guide*.
2. Specify the server as the server for the project:
 - a. Right-click the project in the Project Explorer view and click **Properties**.
 - b. In the list of properties, click **Server**.
 - c. Select the server and click **Apply** or **OK**.

Running a Task Portlet

You can run a task portlet in the server associated with the task application project at any time. For more information, see [“Defining a Server for a Task Application Project” on page 75](#).

➤ To run a task portlet in My webMethods Server

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. Click the **Overview** tab.
3. On the **Overview** tab, expand the User Interfaces area if it is not already visible.
4. Click the task portlet you want to run.
5. Click **Run** in the User Interfaces area. Designer automatically publishes the task application project to the default server for the task project.
6. A My webMethods Server session opens in the task editor.
7. Log in to My webMethods Server to view the page.

Note:

You can also run a task portlet from the Solutions view. To do so, select the task portlet and:

- Right-click the portlet and click **Run on Server**, or
- Click the **Run** menu or  on the main toolbar, and select **Run on Server** from the options.

You can open the portlet in the editor, make and save your changes, and then run the portlet again to view the changes. Designer automatically republishes the modified task application project to the default server for the task project.

Running a Task Portlet View


You can run a portlet view in the server associated with the task application project at any time. For more information, see [“Defining a Server for a Task Application Project” on page 75](#).

➤ To run a task portlet view in My webMethods Server

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. Click the **Overview** tab.
3. On the **Overview** tab, expand the User Interfaces area if it is not already visible.
4. Click the task portlet view you want to run.
5. Click **Run** in the User Interfaces area. Designer automatically publishes the task application project to the default server for the task project.
6. A My webMethods Server session opens in the task editor.
7. Log in to My webMethods Server to view the page.
8. To modify the interface, open the appropriate portlet view in the view editor, make and save your changes, and then run the portlet again to view the changes.

Note:

You can also run a portlet view from the Solutions view or the Project Explorer view. To do so, select the portlet view and:

- Right-click the portlet view and click **Run on Server**, or
- From the **Run** menu or the Run button  on the main toolbar, click **Run on Server**.

You can open the portlet view in the editor, make and save your changes, and then run the portlet view again to view the changes. Designer automatically republishes the modified task application project to the default server for the task project.

Note:

You can also select and run a page in the Pages folder within the User Interfaces area. The Details page and the Start page (where present) are equivalent to the default task view from the Task View portlet and the default start view from the Task Start portlet, respectively. The Inbox page presents the default view of the Inbox Bar portlet as well as the default and edit views of the Inbox Results portlet.

When a task view portlet or task details page is run, Designer connects to the Task Engine and queries for any existing task instances for this task type. If any task instances are found, the first one is used for displaying the task instance details. If no instances are found, then a new instance with no business data is automatically created and used.

Running a Task Page

You can run a task page in the server associated with the task application project at any time. For more information, see [“Defining a Server for a Task Application Project” on page 75](#).

➤ To run a page in a task

1. In the Solutions view, open the task in the editor by double-clicking the task name.
2. Click the **Overview** tab.
3. On the **Overview** tab, expand the User Interfaces area if it is not already visible.
4. Click the page you want to run.
5. Click **Run** in the User Interfaces area. Designer automatically publishes the task application project to the default server for the task project
6. A My webMethods Server session opens in the task editor.
7. Log in to My webMethods Server to view the page.

You can open the page in the editor, make and save your changes, and then run the page again to view the changes. Designer automatically republishes the modified task application project to the default server for the task project.

When a task view portlet or task details page is run, Designer connects to the Task Engine and queries for any existing task instances for this task type. If any task instances are found, the first one is used for displaying the task instance details. If no instances are found, then a new instance with no business data is automatically created and used.

7 Working with Security Roles

■ Security Roles Overview	80
■ Viewing Task Security Roles	80
■ Adding a Security Role to a Task	80
■ Binding a Security Role to a User Interface Control	83
■ Modifying and Deleting a Task Security Role	83

Security Roles Overview

Security roles have the following functions in Software AG Designer:

- To provide a means of binding a functional privilege contained in a task to a control in a portlet view. In this case, the security role has no direct relationship to any My webMethods user role. For example, you can add a security role to a task portlet such as `*TaskView` or `*TaskStart`; this enables you to link a specific privilege within a task to the security role. In this way, you can isolate various task privileges to different security roles.
- To provide a means of assigning a task to a My webMethods user role. In this case, the security role must have the exact same name as a My webMethods user role. When a task application containing a security role is published to My webMethods Server, My webMethods Server will create a corresponding user role if one does not already exist. The role can be further modified in My webMethods after it is created.

Viewing Task Security Roles

➤ To view task security roles

1. Open the portlet editor as described in [“Viewing and Opening Task Portlets” on page 63](#).
2. In the portlet list, expand the portlet you want to work with.
3. Expand the Security Roles listing for that portlet to view any existing security roles. If the portlet contains no security roles, you can add one or more as described in [“Adding a Security Role to a Task” on page 80](#).
4. Make note of the following characteristics and behavior of security roles:
 - Security roles with no linked privileges are used to assign tasks to a corresponding My webMethods user role during run time; you can view the available user roles in My webMethods to ensure that there is an equivalent user role for this type of security role. For more information about viewing user roles in My webMethods, see the PDF publication *webMethods Task Engine User's Guide*.
 - Security roles that contain a linked privilege are used to bind that privilege to a specific control on the task's user interface. For more information, see [“Binding a Task Functional Privilege to a User Interface Control” on page 121](#).

Adding a Security Role to a Task

The specific procedures for adding a security role depend on how you plan to use the security role:

- To provide a means of binding a functional privilege contained in a task to a control in a portlet view, see [“Adding a Security Role for Binding a Task Privilege to a User Interface Control” on page 82](#).

- To provide a means of assigning a task to a My webMethods user role, see [“Adding a Task Assignment Security Role” on page 81](#).

Adding a Task Assignment Security Role

➤ To add a task assignment security role to a task

1. Navigate to the portlet editor and click the Security Roles listing for the task portlet you want to work with (for more information, see [“Viewing Task Security Roles” on page 80](#)).
2. Click **Add**.
3. In the Create Security Role Reference dialog box, specify the following:

Value	Description
Portlet	This value is provided automatically as the name of the selected task portlet.
Role Name	Your public name for the security role. This name must exactly match a corresponding user role in My webMethods, if the user role already exists. If you have not yet created this user role in My webMethods, be sure to use this exact name if you create it in My webMethods manually. When a task application containing a security role is published to My webMethods Server, My webMethods Server will create a corresponding user role if one does not already exist.
Role Link	Do not add any functional privilege links to a task assignment role; doing so will prevent the role from being selected when you create the action expression for the associated assignment rule.
Description	Your description of the security role. These value are used during run-time; you can specify different locales, and a text description for each locale. Click Add to add a new entry to the table.

4. Click **Finish** to add the security role to the portlet.

Although the procedure above creates a new security role within the task portlet, it has no functionality associated with it. You must include the security role in an action rule expression for an assignment rule.

Tip:

You can also add a security role from the Bindings view. First, open any portlet view in the view editor. In the Bindings view, right click the **Managed Beans > [TaskName]View** entry and click **Add > Security Role**. The security role you create is added to the portlet that contains the view in the view editor.

Adding a Security Role for Binding a Task Privilege to a User Interface Control

➤ To add a security role for binding a task privilege to a user interface control

1. Navigate to the portlet editor and click the Security Roles listing for the task portlet you want to work with, as described in [“Viewing Task Security Roles” on page 80](#).
2. Click **Add**.
3. In the Create Security Role Reference dialog box, specify the following:

Value	Description
Portlet	This value is provided automatically as the name of the selected task portlet.
Role Name	Your public name for the security role. This name must exactly match a corresponding user role in My webMethods, if the user role already exists. If you have not yet created this user role in My webMethods, be sure to use this exact name when you create it.
Role Link	Click the Privileges button to view a list of all of the functional privileges contained in the task. Click a privilege and click OK to link it to the security role.
Description	Your description of the security role. These value are used during run-time; you can specify different locales, and a text description for each locale. Click Add to add a new entry to the table.

4. Click **Finish** to add the security role to the portlet.

Although the procedure above creates a new security role within the task portlet, it has no functionality associated with it. You must modify the task view to provide a binding between the role, its linked privilege, and a user interface control on the portlet view design canvas. For more information, see [“Binding a Security Role to a User Interface Control” on page 83](#).



Tip:

You can also add a security role from the Bindings view. First, open any portlet view in the view editor. In the Bindings view, right click the **Managed Beans > [TaskName]View** entry and click **Add > Security Role**. The security role you create is added to the portlet that contains the view in the view editor.

Binding a Security Role to a User Interface Control

Before you carry out this procedure, you must first create a security role and link it with a functional privilege. For more information, see [“Adding a Security Role for Binding a Task Privilege to a User Interface Control” on page 82](#).

➤ To bind a security roles role to a user interface control

1. In the Solutions view, expand the task you want to work with.
2. Double-click the task view portlet that contains the user interface control you want to work with (for example, `[TaskName]TaskView`).
3. In the portlet view editor, click the user interface control you want to work with.
4. In the Properties view, locate the appropriate property value that you want the binding to apply to. For example, if you want to make information available for modification to some users, and read-only to other users, you would work with the **Read Only** property (assuming it exists in the control). Click  in the Properties view toolbar to show or hide all available properties.
5. Click the  associated with the property definition field. The Expression Binding dialog box appears.
6. Expand the entry for the `[TaskName]TaskView`. This listing displays the available security roles in the format User in Role (`[RoleName]`).
7. Click the security role you want to bind to the control and click **OK**.

Note:

The **OK** button will not be enabled if the property field does not support the use of a security role.

8. Save your changes.

You now have the capability of assigning the bound functional privilege to a My webMethodsuser role during run time. For more information about working with user roles in My webMethods, see the PDF publication *webMethods Task Engine User's Guide*.

Modifying and Deleting a Task Security Role

Note:

You cannot delete any of the linked privileges that exist in a security role. You must delete and recreate the security role to remove existing linked privileges.

➤ **To modify or delete a task security role**

1. Navigate to the portlet editor and click the Security Roles listing for the task portlet you want to work with, as described in [“Viewing Task Security Roles” on page 80](#).
2. Click the security role you want to work with and do one of the following:
 - To modify the security role, make your changes in the **Role Name**, **Role Link**, and **Description** boxes and save the changes.

Note:

Do not add any functional privilege links to a security role that is used in a task assignment rule; doing so will prevent the role from being selected when you create the action expression for the associated assignment rule.

- To delete the security role, click **Delete**.
3. Publish the modified task application project to the run-time environment.

8 Configuring Tasks

■ Task Configuration Overview	86
■ Configuring Business Data (Task Inputs and Outputs)	86
■ Configuring Votable Tasks	106
■ Adding E-form Support to a Task	108
■ Working with Services in Tasks	113
■ Working with Task Privileges	117
■ Working with Task Assignments	121
■ Working with Task Events	136
■ Configuring Simple Task Conditions	145
■ Working with Task Control Sets	148
■ Changing the Task Type Name	151
■ Using Gadgets to Render the Task Detail Page in webMethods Business Console	151

Task Configuration Overview

After you create a task, you can define how the task will behave in the run-time environment. For example, you can:

- Add or define business data to be associated with the task, such as a document type that defines task inputs and outputs, one or more data entry fields, or e-form data.
- Create task assignments that will assign a task to a specified user, group, or role depending on the occurrence of a defined condition. For example, when a task status is marked for escalation, assign the task to a manager. You can also configure task assignments to be managed externally, through integration with webMethods Business Rules.
- Create task events that apply an action when a specified change is applied to the task; for example a change in the status, priority, business data, or assignment.
- Create one or more task control sets, each one consisting of a specific collection of task assignments and task events. This enables you to define a specific behavior profile for each control set; when you add the task to a business process in the Process Development perspective, you can specify which control set you want to apply to that particular instance of the task.

Configuring Business Data (Task Inputs and Outputs)

Task business data typically defines the inputs and outputs of a task, and can be added to a task as an Integration Server document type or as a custom business data field.

Note:

Business data field names cannot contain the following characters: < > ' ' & | \ / ;

You have the opportunity to specify the document types and fields you want to use at any time after you create a task. You can also modify the data passed into and out of the task. For more information, see [“About the Task Data Envelope” on page 87](#) and [“Modifying Task Information in the Task Envelope Data Flow” on page 88](#).

You can add business data to a task in the following ways:

- By adding document types and custom fields with the available editors. For more information, see [“Adding an Integration Server Document Type as Business Data” on page 95](#) and [“Adding A Business Data Field with the Editor” on page 94](#).
- By creating a new task from an Integration Server document type located in the Package Navigator view. For more information, see [“Creating a New Task from an Integration Server Document Type” on page 45](#).
- By creating step inputs and outputs for a task activity step using the process editor. For more information, see *webMethods BPM Process Development Help*.

Note:

When you add inputs or outputs to a task step in a process, only those business data fields that are marked as In, Out, or InOut will appear in the task business data display on the Overview page of the task editor.

The input fields of a document type (also known as the input signature) define the information that the task expects for input; the output fields (known as the output signature) define the information that is passed on by the task to the next step in the process. A stand-alone task may not require any output elements.

For example, your document type may contain task input elements for "name," "price," and "quantity." Similarly, a document type may define output elements such as "accept" and "escalate." Or, you might include both input and output elements in a single document type.

You can create Integration Server document types in Software AG Designer using the Service Development perspective. For more information about creating a document type, see *webMethods Service Development Help*.

Note:

You must be connected to your designated Integration Server to create a document type.

Viewing Business Data

You can view the business data associated with a task in the following ways:

- In the Business Data summary area of the **Overview** tab in the editor. The information presented here is read-only and cannot be edited; double-clicking any of the business data fields in the summary will open the **Business Data** tab where editing is possible. You can add business data to a task by dragging it to and dropping it into the Business Data summary area; in this case, the **Business Data** tab opens automatically after you drop the data.
- On the **Business Data** tab, where you can add, delete, modify, and refresh task business data. You can add business data to a task by dragging it to and dropping it into the business data display on the **Business Data** tab.
- In the Business Data entry of the Bindings view.

The Bindings view shows the contents of the Business Data in tree format. In the Bindings view, expand the **Business Data** entry to view the business data (if you make any changes to business data, you must save the task to view the changes in the Bindings view).

You can right click any business data element in the Bindings view to click commands on the pop-up menu, enabling you to delete an element, as well as add data, or add an action, managed bean, or content provider, as appropriate.

About the Task Data Envelope

Tasks in a business process model typically receive data from the process pipeline when the task is started (queued). When the task completes, the task passes the resulting data back to the pipeline. This data includes both business data processed by the task, as well as task information such as task ID, task name, status, and so on.

Task business data flows from the process instance pipeline running in webMethods Process Engine to the task instance running in webMethods Task Engine, and then the task output flows

back to the Process Engine when the task completes. From a task development point of view, this data is referred to as follows:

- The task info passed from the process pipeline to the task being queued is the *task input envelope*.
- The task info passed back to the pipeline from the task when it completes is the *task output envelope*.

Portions of the task envelope are available to you to enable you modify the data being passed to and from the Task Engine. For more information, see [“Modifying Task Information in the Task Envelope Data Flow” on page 88](#).

Modifying Task Information in the Task Envelope Data Flow

You can add a functioning task activity to your process model by either adding an existing task from the Solutions view, or by implementing a new task directly from the process canvas. For more information, see *webMethods BPM Process Development Help*.

Either way, when the task is added, Designer creates two wrapper services to handle the task envelope data:

- `[taskName]InputMapService`. This is the input mapping service that is responsible for mapping business data and task info from the pipeline to queue a task. Task info is passed to this service in the document `pub.task:TaskQueueInfo`.

You can use this document to modify the data being passed to the task from the pipeline. For example, you can use the *assignedToList* string element to specify one or more task assignees.

- `[taskName]OutputMapService`. This is the output mapping service that is called by Process Engine when a task is completed (that is, ended with "Completed", "Error", "Cancelled", or "Expired" status). It contains the document `pub.task:TaskCompletionInfo` which is populated with the data from the completed task.

You can use this document to modify the data being passed to the pipeline from the task. For example, you can create error logic that determines a particular error code and add it to the document.

The use of these documents is not required, but they are always available should you want to use them.

For more information about the documents `pub.task:TaskQueueInfo` and `pub.task:TaskCompletionInfo` and their contents, see the built-in services chapter of the PDF publication *webMethods Task Engine API and Service Reference*. For general information about Task Engine services, see [“Working with the Task Engine APIs” on page 259](#).

For more information about working with input and output mapping and document handling, see *webMethods Service Development Help*.

About Indexed Business Data Fields

Indexed business data fields are available as a standard feature with Task Development Version 8.2 and later. Task applications created with earlier versions can be modified to support indexed data field searches. Any task application, regardless of version, can be configured to continue using the standard (full) search provider. For general information about indexed searches, see [“About Indexed Searches” on page 159](#).

Indexed data fields display the text **Yes** in the Indexed column of the business data display. For more information about working with indexed business data fields, see the following:

- To mark business data fields as indexed fields, see [“Adding A Business Data Field with the Editor” on page 94](#) and [“Modifying Task Business Data Fields” on page 96](#).
- To add indexed fields to a task search user interface, see [“Creating Custom Search Fields from Task Business Data” on page 104](#).
- To add indexed fields to a task view or page interface, see [“Adding Task Business Data Fields to a Page or View” on page 102](#).
- For information about how indexed fields appear in the bindings view, see [“About Task Indexed Search Content Provider Binding Properties” on page 175](#).

To configure an existing task type to support indexed fields, see [“Implementing an Indexed Search Content Provider for Existing Task Types” on page 161](#).

To configure a task type to use the full (standard) search mechanism instead of indexed fields, see [“Implementing a Standard Search Provider for Existing Task Types” on page 167](#).

For more information about the behavior of indexed fields in the run-time environment, see [“About Business Data Fields in the Run-time Environment” on page 90](#).

About HPSTRA-enabled Tasks and Business Data Fields

You can enable High Performance Task Search Reference Architecture (HPSTRA) support for task types and individual business data fields in a task type. By enabling HPSTRA support, you can index the task information fields, and task business data fields to an Elasticsearch store for high-performance search in the runtime. You can enable/disable tasks types and business data fields for indexing with HPSTRA when designing a task, or at any time after you create the task type.

By default, the complete task data of a HPSTRA-enabled task type is stored in Elasticsearch. This includes all task information and task business data fields. To limit the number of indexed business data fields, you must disable the HPSTRA-indexing for particular fields, and redeploy the task type. Each time you make changes to the HPSTRA settings of a task type, or modify the HPSTRA-enabled business data fields, you must redeploy the task type. When you redeploy a task type, Task Engine removes the existing Elasticsearch index for the task type and all its contents, and creates a new, empty index. You must manually run a reindexing operation to populate the newly created Elasticsearch index with data from existing task instances in your system.

For more information about modifying the HPSTRA settings of a business data field, see [“Modifying Task Business Data Fields” on page 96](#). For more information about HPSTRA reindexing in the runtime, see *webMethods Task Engine User’s Guide*.

About Business Data Fields in the Run-time Environment

When a task application is published to the My webMethods Server runtime, individual task instances can be started either automatically from a process or manually by a user. The specified business data for a task instance is populated from various sources as the task moves through its life cycle.

These business data fields and their values are stored in one of two database tables created and maintained by the Task Engine:

- The Task Engine stores all task business data for all tasks as a binary large object (BLOB) within a full-search database table. While this makes all business data available for all tasks, the BLOB must be de-serialized to make the information available, and it can be quite large when many tasks are present.
- When a task contains indexed business data fields, those fields are stored in a separate indexed fields database table as well as the BLOB; the indexed fields table contains only indexed business data fields and their values. The indexed field table is created dynamically when a task application containing indexed business data fields is published to My webMethods Server. The indexed field table provides a limited but much faster search mechanism that is especially useful in situations when a very large number of tasks are present in a task inbox.

The Task Engine maintains both database tables and updates them dynamically as new tasks are queued in the system or as existing tasks are deleted from the system.

As noted above, the indexed field table is populated any time a published task contains indexed business data fields. Special considerations apply when you publish a task type that contains indexed business fields to the run-time environment; for more information, see [“Considerations When Publishing Tasks with Indexed Data” on page 168](#).

To enable the run-time user to access this data, you can implement either or both of the following methods:

- Configure the task to access the data with an indexed search provider.
- Access the indexed business data table with a built-in search service in the WmTaskClient package on Integration Server. For more information, see the indexed search services described in the PDF publication *webMethods Task Engine API and Service Reference*.

Note that if you configure a task with only an indexed search provider and you do not include any indexed data fields, run-time business data searches executed by the user will return no results even though the business data exists in the task.

About Business Data and Collaboration Tasks

Task business data can be shared between tasks queued from different task types, *as long as they have identical business data* (for example, the task data in each task has been created from the same IS document type).

On the Task Engine Administration tab of My webMethods, you can enable or disable data sharing for each task type. When the user queues a new collaboration task from a task type where data sharing is enabled, the new collaboration task will contain a copy of the business data from the parent task.

You can configure your parent and child tasks to keep business data synchronized between them. You can do this by creating a task event that incorporates one of the available "Set Business Data" actions. For more information, see [“Synchronizing Parent-Child Task Business Data” on page 91](#).

Synchronizing Parent-Child Task Business Data

If you choose to share business data between parent and child collaboration tasks, you can configure these tasks to keep business data synchronized between them. For example, you can update the child task data to reflect a change in parent task data, and the reverse is also true.

➤ To synchronize parent-child task business data

1. Ensure that both parent and child tasks have identical business data.
2. Open the task you want to work with in the task editor.
3. Create a task event as described in [“Creating a Task Event” on page 137](#). Choose an event type that will trigger your business data update, such as Business Data Updated.
4. Specify an action for the event:
 - If you are updating child task business data from a parent task, specify one of the following actions:
 - **Set Business Data Date field**
 - **Set Business Data field**
 - If you are updating parent task business data from a child task, specify one of the following actions:
 - **Set Parent Business Data Date field**
 - **Set Parent Business Data field**

For more information about working with these fields, see [“Configuring Simple Task Actions” on page 140](#).
5. Save your changes.

About Business Data and E-forms

You can use electronic forms (e-forms) as task business data. Basic e-form support can be implemented by storing e-form templates and instances in My webMethods Server or in a file system or web server. For more information and a list of the supported e-form types, see the PDF publication *Implementing E-form Support for BPM*.

Before you can work with e-forms, you must:

- Create a repository for your e-form instances and templates.
- Configure a webMethods e-form environment in My webMethods and deploy that environment to the Integration Server host(s) where you will be creating your IS document types.
- Ensure that you have network connectivity between all of the host servers.
- Review various implementation issues for the supported e-form types.

For additional information and instructions about the use of e-forms in the webMethods product suite, see *Implementing E-form Support for BPM*.

The first step in using an e-form as business data is to import it into Software AG Designer as an IS document type, making the e-form data available in that format. For information about importing an e-form template as an IS document type, see *webMethods Service Development Help*.

After the IS document type is created, it can be used just like any other IS document type. For example:

- You can add the e-form data contained in the IS document type to a task as described in [“Adding an Integration Server Document Type as Business Data” on page 95](#).
- After adding the IS document type to a task, you can add business data fields from the IS document type to a page or view, as described in [“Adding A Business Data Field with the Editor” on page 94](#) and [“Adding Task Business Data Fields to a Page or View” on page 102](#).
- You can identify any of the individual business data fields in the IS document type as indexed fields, as described in [“Modifying Task Business Data Fields” on page 96](#).
- You can create a new task from the IS document type, as described in [“Creating a New Task from an Integration Server Document Type” on page 45](#).

When working with e-form sourced IS document types, keep these points in mind:

- After you add an e-form sourced IS document type to a task, you must configure the task for e-form support. For more information about implementing e-form support in a task, see [“Adding E-form Support to a Task” on page 108](#).
- Special considerations apply to keeping an e-form template in sync with its resulting IS document type; for more information, see [“Synchronizing E-form Templates and IS Document Types” on page 92](#).

Synchronizing E-form Templates and IS Document Types

Because e-form templates are converted into IS document types, you must be aware that any modification to either the e-form template or to the IS document type will result in mis-matched (or out of sync) documents.

- After an e-form is imported and an IS document type is created from it, any modifications to the content or structure of the IS document type will cause it to be out of sync with the e-form template from which it was created. When an instance of the e-form template is received, it will no longer match the IS document type, and therefore will be unusable by the Process Engine.
- The same situation will occur if you change the e-form template, and instances arrive in the repository from the modified e-form. The instances will no longer match the IS document type created from the initial template.

In both cases, you must:

1. Delete the existing IS document type.

Note:

The renaming action required in the following step is necessary only if the IS document type and its associated schema are used in more than one place. The file name of the e-form template is used to set the schema domain for the associated schema, and a unique file name will yield a unique schema domain. Integration Server maintains a central schema registry, so if a schema for a particular document type is used anywhere in the Integration Server, that definition will take precedence in all locations, regardless of subsequent imports. The only way to get the new definition to take precedence is to rename the imported e-form template, or to ensure that the old one is completely deleted from the Integration Server. Finding the old schema definition may not be easy because the original schema may be used in another package or folder.

2. Change the name of the e-form template you have modified; for example, if you have modified an e-form template named `my_template.xdp`, add a revision number, such as `my_template_r1.xdp`.
3. Use the New Document Type wizard to recreate the IS document type from the latest version of the e-form template.

If there are changes to the IS document type that will affect your process model (such as data structure changes), additional procedures are required, as follows:

Note:

Changes to field attributes in the e-form template do not require the following actions.

You must:

- Delete the existing process model receive step that uses the IS document type.
- Drag the recreated IS document type to the process canvas to create a new receive step in process model and configure the step accordingly.
- Map fields from the receive step to the pipeline as needed.
- Build and upload the process model.

For more information about working with IS document types, see *webMethods Service Development Help*.

For more information about webMethods e-forms solutions, see *Implementing E-form Support for BPM*.

Adding A Business Data Field with the Editor

After you create a task, you can add a custom business data field with the appropriate editor. This field is added to the task business data as a separate data element; you cannot add a data field to an IS document type with this method—to do so, you must edit the IS document type, then refresh the IS document type in the task editor.

➤ To add a business data field in the editor

1. In the Solutions view, double-click the task to open the selected item in the editor.
2. Click the **Business Data** tab. Task business data is displayed in the Business Data area in the left area of the tab.
3. Click **Add**.
4. In the Field Type Selection dialog box, select the type of business data you want to add in the display list on the left.
5. On the right side of the editor, define the properties of the new field:
 - **Field Name.** Type the name you want to apply to the field.
 - **Input.** Select this check box if the field is to be used as an input.
 - **Output.** Select this check box if the field is to be used as an output.
 - **List.** Select this check box if the field is to be used as a list.
 - **IS document.** (Available for document reference fields only) Select this check box if the field is to be used as an IS document reference. Click **Browse** to locate and specify an IS document type.
 - **Index field in database.** Select this check box to include this field in an indexed field database table in the run-time environment.

Note:

This check box is not available for IS document references, but can be applied to fields within an IS document reference. Also, only non-list primitive types can be marked as indexed fields (that is, the **List** check box must be cleared; otherwise, the field is an array and cannot be indexed).

Specify the following database field settings:

- **Name.** The name of the field representing the business data in the indexed search provider business data in the Bindings view. The business data field name is displayed by default; however, the name must be unique; modify this name here as required.

For example, suppose you want to index two separate but similar fields: *Shipping Address > City* and *Billing Address > City*. By default, both fields display the name “City”; if you simply accept the default value, both will appear as “City” in the Bindings view and will not be distinguishable. The solution is to apply two different names such as “Shipping City” and “Billing City” so they are easily identified in the Bindings view.

- **Type.** The field type of the business data as it will be stored in the database table. The business data field type is displayed by default; however, if you want to store the data in the indexed field database table in a different type format, specify that format here.
- **Size.** Enabled only for field types that require a size definition (for example, var char). Type the field size definition here.
- **Binding.** Available only for fields that are children of a list. At run time, the total number of objects in a list is not known; therefore, you must specify which object from the list you want to index by providing an index value in the binding expression displayed in the Binding field:

```
#[currentTask.taskData.ISDocumentName.ListName[0]]
```

where “0” (zero) represents the index value. The value “0” (the first item in the list) is specified by default, but you can replace it with any index value. You can index only one object from a list.

6. Click **OK**.

The business data field is added to the task. If you want to expose the business data field in the user interface, see [“Adding Task Business Data Fields to a Page or View” on page 102](#).

Adding an Integration Server Document Type as Business Data

➤ To add an Integration Server document type as business data

1. In the Solutions view, double-click the task to open it in the editor.
2. Click the **Business Data** tab. Task business data is displayed in the Business Data area in the left area of the tab.
3. Locate the Integration Server document type you want to work with in the appropriate data browser (for example, the Package Navigator view).
4. Drag the document type and drop it onto the Business Data area.

Tip:

You can also drop the IS document type onto the Business Data summary area on the **Overview** tab; this automatically opens the **Business Data** tab.

5. Click **Edit** if you want to modify the document type properties:
 - If the document type represents an input of the task, select **Input**.
 - If the document type represents an output of the task, select **Output**.
 - If the document type is a list, select **List**. You cannot select a document list as input or output; the **Input** and **Output** check boxes are automatically cleared when you select **List**. To use a list of document types as input and output, wrap the list as a child of another document type and then select the parent document type.
6. Click **OK**.
7. Save the task.

After you add the IS document, you can:

- Add the business data to a view or page as described in [“Adding Task Business Data Fields to a Page or View” on page 102](#) and [“Dragging and Dropping Task Business Data Fields to a Page or View” on page 103](#).
- Mark any of the data fields in the document as indexed fields, as described in [“Modifying Task Business Data Fields” on page 96](#).
- View the name and expression values for the document type by clicking it in the Bindings view and then opening the Properties view.

Modifying Task Business Data Fields

You can modify top-level business data fields in a task at any time.

➤ To modify business data fields

1. In the Solutions view, double-click the task to open it in the editor.
2. Click the **Business Data** tab. Task business data is displayed in the Business Data area in the left area of the tab.
3. Click the business data field you want to modify.

Note:

In the task editor, you can modify field properties for top-level business data fields only. If you select a child field (such as a field contained in an IS document type), the field properties will be disabled for editing (you can still mark a child field for indexing, however). To modify the field properties for a child field, you must modify the IS document type that contains the field. To do so, select the IS document in the business data display and click **Open in**

Editor. After you modify and save the IS document, you must click **Refresh** to update the IS document type in the task editor.

4. Click **Edit** to modify any of the following business data field properties:

- **Field Name.** Type the name you want to apply to the field.
- **Input.** Select this check box if the field is to be used as an input.
- **Output.** Select this check box if the field is to be used as an output.
- **List.** Select this check box if the field is to be used as a list.
- **IS document.** (Available for document reference fields only) Click **Browse** to locate and specify an IS document type.
- **Index field in database.** Select this check box to include this field in an indexed field database table in the run-time environment. This check box is not available for IS document references, but can be applied to fields within an IS document reference. Also, only non-list primitive types can be marked as indexed fields (that is, the **List** check box must be cleared; otherwise, the field is an array and cannot be indexed).

Specify the following database field settings:

Field	Description
Name	The name of the field representing the business data in the indexed field database table. The business data field name is displayed by default; however, the database name must be unique; modify this name here as required to create a unique database field name.
Type	The field type of the business data as it will be stored in the database table. The business data field type is displayed by default; however, if you want to store the data in the indexed field database table in a different type format, specify that format here.
Size	Enabled only for field types that require a size definition (for example, var char). Type the field size definition here.
Binding	<p>Available only for fields that are children of a list. At run time, the total number of objects in a list is not known; therefore, you must specify which object from the list you want to index by providing an index value in the binding expression displayed in the Binding field:</p> <pre><code>#{currentTask.taskData.ISDocumentName.ListName[0]}</code></pre> <p>where “0” (zero) represents the index value. The value “0” (the first item in the list) is specified by default, but you can replace</p>

Field	Description
	it with any index value. You can index only one object from a list.

- **HPSTRA enabled.** Available only for fields in HPSTRA-enabled task types. Select this check box to include the business data field in an Elasticsearch index.

For **Type**, select the type of Elasticsearch index to use for the field. Available options are:

- **analyzed** - the value of the field is indexed for full-text search. This is the default value.
- **not_analyzed** - the value of the field is indexed for searching, but only exact matches will be returned, including case and special characters.
- **stored** - the value of the field is stored in Elasticsearch, but is not indexed. Stored fields are not available for filtering and can only be retrieved as part of a result set.

For more information about Elasticsearch index settings, see [“About HPSTRA Searches” on page 160](#) and the Elasticsearch documentation.

The **Binding** field displays the index value of the current field in the binding expression. You cannot modify this value for HPSTRA fields.

5. Click **OK**.
6. Save your changes.

Considerations When Modifying Task Business Data

During the lifecycle of a task instance, the task business data is obtained from various sources, serialized, and stored in the My webMethods Server database. For more information about task data serialization, see [“About Business Data Fields in the Run-time Environment” on page 90](#).

When you modify and republish a task type to the runtime, Task Engine overwrites the existing task type. In the same way, the business data definition, associated with a task type is overwritten each time you change and republish the definition.

To ensure that task business data is available at runtime, the changes you make to the task data definition must conform to the structure of the task data, already serialized in the database. For example, when you use an IS document type for task business data, you can modify that document by adding or removing business data fields. After republishing the IS document type, the business data, created before republishing the IS document type will continue to exist, and will be retrievable by Task Engine.

However, certain changes to the data structure will affect how the data is serialized in the My webMethods Server database, and the business data for task instances, queued before the modification of the data definition will not be available at runtime. For example, when you change the type of an existing business data field in an IS document type, the business data for tasks, queued before the change will become unavailable.

My webMethods Server uses Java object serialization when storing task business data to the database. For more information about the limitations of Java object serialization, see the *Java Object Serialization Specification*.

Deleting Task Business Data

You can delete business data in a task at any time.

> To delete business data

1. In the Solutions view, double-click the task to open it in the editor.
2. Click the **Business Data** tab. Task business data is displayed in the Business Data area in the left area of the tab.
3. Click the business data element you want to delete.

Note:

Only top-level business data fields can be deleted in the task editor. If you want to delete a child field such as a field contained in an IS document type, you must delete it in the IS document type. To do so, select the IS document in the business data display and click **Open in Editor**. After you modify and save the IS document, you must click **Refresh** to update the IS document type in the task editor.

4. Click **Delete**.
5. Click **Yes** in the Delete confirmation dialog box.
6. Save your changes.

Note:

You can also delete business data elements from the Bindings view. In the Bindings view, expand the Business Data entry and right-click the business data element you want to delete, then click **Delete**. Be sure to modify all existing references to the deleted data, including interface controls, data bindings, and process model steps.

Updating or Refreshing Task Business Data

As you work with a task, the associated business data can be modified from within the editor or from the process editor (for more information, see [“Configuring Business Data \(Task Inputs and Outputs\)” on page 86](#)). Similarly, an Integration Server document may have been updated outside of the task development process.

When task business data has changed, use one of the following methods to update or refresh the data:

- To refresh the task business data from the task editor, see [“Refreshing Document Business Data in the Editor” on page 100](#).
- To refresh the task business data from an Integration Server document, see [“Refreshing Task Business Data from an Integration Server Document” on page 100](#)
- To update the task business data with changes you have entered to task step in a process, see [“Updating Business Data from a Task Step in a Process” on page 101](#).

Refreshing Document Business Data in the Editor

You can refresh the document business data from the task editor at any time. This procedure will sync the business data to the latest version of an IS document.

Note:

The **Refresh** button in the editor is enabled only for Integration Server documents.

➤ To refresh document business data

1. In the Solutions view, double-click the task to open it in the editor.
2. Click the **Business Data** tab. Task business data is displayed in the Business Data area in the left area of the tab.
3. Click the IS document you want to refresh (you can select two or more by holding down the CTRL key while clicking).
4. Click **Refresh** in the Business Data area. The selected IS documents are refreshed to match the latest version of the document in the source location.
5. Save your changes.

Refreshing Task Business Data from an Integration Server Document

From the Package Navigator view, you can refresh the task business data for all of the tasks in a task application that consume an individual Integration Server document. This procedure will sync the business data used by all tasks in that project to the latest version of an IS document.

This procedure is an efficient way to apply changes made to an IS document to all of the tasks that use the document; otherwise, you would have to open each task individually and manually update the business data.

➤ To refresh task business data for an IS document

1. In the Package Navigator view, locate the Integration Server document you want to work with.
2. Right-click the document and click **CAF > Update Documents**.

3. In the Generate IS Doc Wrapper Java Types wizard, select the task application project the contains the tasks you want to refresh in the **Project** box.
4. Do one of the following:
 - If you want to completely overwrite (delete and recreate) the underlying Java bean that contains the IS document information for each task, click **Finish**. This completes the procedure.
 - If you want to retain the existing Java bean for each task and only add, remove, or update changed fields, click **Next**.
5. Clear the **Completely overwrite generated Java classes** check box.
6. Click **Finish**.

The business data for all tasks in the selected task application project is refreshed to match the contents of the selected IS document.

Updating Business Data from a Task Step in a Process

Use this procedure to update task business data when you have modified the task business data in a task step in a process. To refresh business data in the task editor, see [“Refreshing Document Business Data in the Editor” on page 100](#).

Note:

When you add inputs or outputs to a task step in a process, only those business data fields that are marked as In, Out, or InOut will appear in the task business data display on the Overview page of the task editor.

➤ To update task business data from a task step in a process

1. With the process open in the process editor, locate the task step you want to work with.
2. Right-click the task step you have modified and click **Update Task Business Data**.
3. The task business data is updated as follows:
 - Inputs and outputs are added or removed as necessary.
 - Any changed input or output types are removed and re-added.
 - Private task data fields (those that are not marked as input or output) are left unmodified.

You can view the updated task business data on the **Overview** and **Business Data** tabs of the task editor. Double-click the task step to open the task in the editor, or right-click the task step and click **Open in Task Editor**.

Adding Task Business Data Fields to a Page or View

You can add existing task business data fields to a task portlet view, or a notification view in the editor, as described below. You can also add business data fields by dragging and dropping, as described in [“Dragging and Dropping Task Business Data Fields to a Page or View” on page 103](#).

Note:

You cannot add business data to the Task Inbox Search Bar view or the Task Inbox Search Results edit view using this procedure. To add business data to the Task Inbox Search Bar view, see [“Creating Custom Search Fields from Task Business Data” on page 104](#)

➤ **To add business data fields to a page or view**

1. In the Solutions view, double-click the task to open it in the editor.
2. Click the **Overview** tab. Task business data is displayed in the Business Data area in the upper-right area of the editor. If the business data you want to work with is not present, add it as described in [“Adding A Business Data Field with the Editor” on page 94](#) and [“Adding an Integration Server Document Type as Business Data” on page 95](#).
3. On the **Overview** tab, expand the User Interfaces area if it is not already visible.
4. Click the page or view you want to add business data to; for example, the default view of the Task Details portlet.
5. Click **Update** in the User Interfaces area. The Task UI Update wizard appears with the current properties for the selected element displayed.
6. Click the **Add task business data field(s) to selected task UI page** option.
7. Click **Next**. The Add Task Business Data page appears.

Note:

This display is empty if:

- There is no business data associated with the task.
- Business data has been added but the task has not been saved.
- The task uses an indexed search content provider and no business data has been marked as an indexed field.

8. Expand the table entries to locate and click the business data fields you want add.
9. If you want to open the page or view in the view editor after applying the business data changes, select the **Open updated page in Editor** check box.
10. Click **Finish**. The business data fields are added to the view as follows:

- For the Task View, Task Start, and Task Details views, the business data is added to the Task Data subgroup in the user interface.
- For the Task Inbox Results default view and Task Inbox view, the business data is added as a column in the results table.

11. The changes to the task are saved automatically.

Dragging and Dropping Task Business Data Fields to a Page or View

You can add existing task business data fields to a page or view by dragging and dropping, as described below. You can also add business data fields using the editor as described in [“Adding Task Business Data Fields to a Page or View” on page 102](#).

➤ To drag and drop business data fields

1. In the Solutions view, double-click the task to open it in the editor.
2. Click the **Overview** tab. Task business data is displayed in the Business Data area in the upper-right area of the editor. If the business data you want to work with is not present, add it as described in [“Adding A Business Data Field with the Editor” on page 94](#) and [“Adding an Integration Server Document Type as Business Data” on page 95](#).
3. On the **Overview** tab, expand the User Interfaces area if it is not already visible.
4. Double-click the page or view you want to add business data to (for example, the default view of the Task Details Portlet) to open it in the task view editor.

Tip:

You can also open the view editor from the Solutions view by expanding the task in the Solutions view and then double clicking the view you want to work with.

5. In the Bindings view, view the available business data fields. For example, for the default view of the Task Details portlet, expand **Managed Beans > [TaskName]View/default > [TaskName] > Business Data**.

Note:

This display is empty if:

- There is no business data associated with the task.
- Business data has been added but the task has not been saved.
- The task uses an indexed search content provider and no business data has been marked as an indexed field.

6. Click the business data field you want to work with and drag it to the desired location in the editor canvas.

7. Save your changes.

Creating Custom Search Fields from Task Business Data

You can create custom search fields in a Task Inbox Bar portlet view by using business data fields, as described below.

Note:

By default, custom field searches are always case-sensitive. If you want to make case-insensitive searches available, see [“Implementing Case-Insensitive Searching” on page 158](#).

➤ **To create a custom search field from a business data field**

1. In the Solutions view, double-click the task to open it in the editor.
2. Click the **Overview** tab. Task business data appears in the Business Data area in the upper-right area of the editor. If the business data you want to work with is not present, add it as described in [“Adding A Business Data Field with the Editor” on page 94](#) and [“Adding an Integration Server Document Type as Business Data” on page 95](#).
3. On the **Overview** tab, expand the User Interfaces area if it is not already visible.
4. Click the default view of the Task Inbox Search Bar portlet.
5. Click the **Update** button in the User Interfaces area.
6. In the Task UI Update wizard, make sure the **Implement task custom inbox bar search for task business data field(s)** option is selected, then click **Next**.
7. On the Add Searchable Field page, the **Data Fields** table displays the available business data.

Note:

This table is empty if:

- There is no business data associated with the task.
- Business data has been added, but the task has not been saved.
- The task uses an indexed search content provider and no business data has been marked as an indexed field.

8. Click the business data field you want to work with to place it in the **Search term name** box. Only one field is accepted each time; you must repeat steps 4 - 7 to add more fields.
9. Select the operator you want to apply to the field in the **Search operator** box.

Note:

To implement a wildcard search, the search operator for the selected field *must* use the `like` operator. For more information about the available operators, see [“About Task Search Operators” on page 105](#).

10. If you want a column to be added to the search results table, select the **Add column displaying this field to Inbox Search Results table** check box.
11. If you want to open the page or view in the view editor after applying the business data changes, select the **Open updated page in editor when finished** check box.
12. Click **Finish**.
13. The changes to the task are saved automatically.

About Task Search Operators

When you create custom search fields as described in [“Creating Custom Search Fields from Task Business Data” on page 104](#), you specify an operator to be used by the Task Engine to find matching field values.

Note:

By default, custom field searches are always case-sensitive. If you want to make case-insensitive searches available, see [“Implementing Case-Insensitive Searching” on page 158](#).

All operators do a compare on the object type, except for the `like` operator, which implements a wildcard search (see below). For example, for a data type of Integer, all data comparisons are done with the integer object. Your search input is converted to an Integer and compared to the business data field's Integer value.

The following table lists the available operators:

Operator	Definition
=	Equals.
>	Greater than.
<	Less than.
<>	Does not equal.
>=	Greater than or equal.
<=	Less than or equal.
like	The <code>like</code> operator enables you to find task business data for a field by specifying partial search characters and an asterisk (*) character (a wildcard search). For example, to do a wildcard search for a zip code, you could enter "980*", which returns results for all zip code data from 98000 to 98099.

Operator	Definition
not like	This operator provides the same wildcard searching as the <code>like</code> operator, but with the opposite results.
between	The data returned by the search falls between two dates.
in	The data returned by the search contains a specified value in an array. For example, if an array <code>[1,2,3]</code> exists, searching for a value of 2 will obtain a match. This operator also works for a single string, but acts like the <code>=</code> operator.
not in	This operator provides the same array searching as the <code>in</code> operator, but with the opposite results.

Configuring Votable Tasks

You can enable voting by selecting the **Voting enabled** option when creating a new task type with the Create New Task Wizard, or add voting capabilities to existing task types as described in [“Configuring Votable Tasks” on page 106](#).

Configuring Task Voting

➤ To configure voting for a task type

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. In the **General Information** section of the **Overview** tab, select the **Voting enabled** checkbox, if not already selected.
3. Under **Select voting strategy**, select the type of vote threshold. The options are: **Percentage** and **Fixed value**.

For more information about strategy calculations, see [“About Task Voting Strategies” on page 106](#).

4. Optionally, modify the default threshold in the **Value** field.

The default value is 51 for both voting strategies. Depending on the type of strategy, the value is calculated as either percentage or number.

About Task Voting Strategies

Task types can be configured with two voting strategies for setting the vote threshold beyond which Task Engine updates the status of a votable task instance:

- **Percentage** - the minimum number of users to select the same status, calculated as a percentage from the total number of users. The default value is 51%. Task Engine calculates the percentage and rounds the result to ceiling.
- **Fixed value** - a minimum number of users, as an integer value. The default is 51.

Task assignees can be individual users, groups, or roles. Task Engine considers the number of individual users in the groups or roles when performing task vote calculations.

For example, if you apply the default configuration of the percentage strategy, which is 51%, to a list of assignees to the total of 30 users, Task Engine rounds the calculation to a minimum of 16 user votes required to update the task status to a particular option. For a list of 300 users, the threshold is 153.

You can modify the type of strategy or the threshold value only at design time.

Runtime Considerations for Votable Tasks

Regardless of the user interfaces that a task developer creates for a votable task type, task assignees are presented with certain voting options, each corresponding to one of the following task statuses:

- For tasks in status Active, possible transitions are to statuses Suspended, Cancelled, Completed, and Error
- For tasks in statuses Expired and Suspended, possible transitions are to statuses Active and Error.

Depending on the voting strategy that a task developer configures and the number of runtime assignees for a task, some task instances might never meet the requirements for transitioning to one of the available statuses. In such situations, task type developers can modify the strategy and redeploy the task type, or Task Engine administrators can manually change the status of the task at runtime.

For example, a task instance from a type which is configured with a voting threshold of 60% and provides two voting options, gets assigned to a group of ten individual users. If five users select one of the voting options, and the other five select the other, the voting threshold is never reached and the task instance cannot transition to any of the configured statuses. Such tasks require manual intervention by an administrator, or a modification in the voting strategy. Runtime users with the appropriate permissions can monitor task voting statistics, and take actions when a task instance cannot reach a final status through voting.

The following considerations apply when modifying the voting strategy for a task type:

- The modified voting strategy for the task type applies to instances, queued after redeployment.
- Existing task instances can be updated to the modified strategy by manually triggering vote recalculation.
- When recalculating task vote according to the new voting strategy, Task Engine also recalculates the number of users in groups and roles.
- Some task instances might transition to a new status as a result from the recalculation.

- When updating the status of a task as a result from vote recalculation, Task Engine places the task in the first status that reaches the percentage threshold by timestamp.

Task type developers can enable voting in existing task types, created with the current (or later) version of Designer. By default, all task instances queued after the redeployment of the modified task type will be enabled for voting with the configured strategy. Instances of the type that were available in the runtime before redeployment cannot be enabled for voting. The same consideration applies to task types, that were initially enabled for voting. If you modify the task type, voting will remain enabled for each instance, queued before you make the change and redeploy the task type, until the instance gets completed or reaches another terminal state. All new instances, queued after you redeploy the task type will be enabled for voting.

Software AG recommends that redeployment of task types with voting modifications and vote recalculation are executed during scheduled maintenance windows.

For more information about recalculating task votes and handling voting issues at runtime, see *webMethods Task Engine User's Guide*.

For more information about task statuses, see [“About Task Status” on page 30](#).

About the Task Voting Chart and Statistics Table

The default task user interfaces of all tasks, enabled for voting include a voting distribution chart. All task assignees that have view permissions for the task user interfaces can also view the voting chart. Before the voting starts and after the voting completes the chart displays no values.

You can modify the properties of the CAF Include HTML control that contains the diagram canvass on the **Design** tab when you open the Task View portlet in the editor. No design time preview is available.

For runtime users with the appropriate permissions, the **Voting Details** button on the default Task View portlet toggles the visibility of the **Task Voting Statistics** table. For more information about the user permissions, required to view voting details, see *webMethods Task Engine User's Guide*.

Adding E-form Support to a Task

The webMethods product suite enables you to integrate e-forms with your tasks by using IS document types created from an e-form template.

Adding e-form support to a task requires the following high-level steps:

- Creating and configuring the content provider for the e-form instances, and the template provider for the specified e-form template.
- Adding a basic interface with download and upload controls to the default view of the Task Details portlet.
- Optionally, embedding a copy of the e-form template into the task application.
- Optionally, adding download capability to the task inbox results list.

You must follow the high-level steps for each e-form you want to add to the task.

For more information about adding e-form support for tasks, see [“Adding E-forms Stored in a My webMethods Server Folder Location” on page 109](#) and [“About Business Data and E-forms” on page 92](#). For complete information about implementing e-form integration with the webMethods product suite, see *Implementing E-form Support for BPM*.

Adding E-forms Stored in a My webMethods Server Folder Location

You integrate e-forms with your tasks using e-forms stored in a My webMethods Server folder location. Before adding e-form support to a task, you must create an IS document type by importing an e-form template. For more information, see [“About Business Data and E-forms” on page 92](#).

➤ To add e-form support to a task

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. Add an e-form-sourced IS document type to the task's business data, as described in [“Adding an Integration Server Document Type as Business Data” on page 95](#), or create a new task from an e-form-sourced IS document type, as described in [“Creating a New Task from an Integration Server Document Type” on page 45](#).
3. On the **Overview** tab, expand the User Interfaces area if it is not already visible.
4. Click the Default view for the Task Details portlet.
5. Click the **Update** button in the User Interfaces area.
6. In the Task UI Update wizard, do one of the following:
 - Click the **Add E-form support to the selected task UI page** option. Use this option if you want to add e-form support to the task, but you do not want to add any of the business data in the e-form-based document to the task interface. Click **Next** and proceed to step 7.
 - Click the **Add task business data field(s) and E-form support to the selected task UI page** option. Use this option if you want to add e-form support to the task, and you want to add some or all of the business data in the e-form-based document to the task interface. If you choose this option, you must then:
 1. Click **Next**.
 2. On the Add Task Business Data page, expand the table entries to locate and click the business data fields you want to add.
 3. If you want to open the page or view in the view editor after applying the business data changes, select the **Open updated page in Editor** check box.
 4. Click **Finish**. The business data fields are added to the view.

7. On the Add E-form Support dialog box, select the e-form-sourced IS document type in the **Document Name** list.

Note:

All IS document types added to the task appear in the Document Name list; be sure that the IS document type you select is one that was created from an e-form template.

8. In the **Template Type** list, select the e-form application that was used to create the e-form template that the IS document type was created from (for example, InfoPath or LiveCycle).
9. Click **Browse** to specify a value for the **Template Location** box. In this procedure, you are specifying the location of the e-form template that you imported to create the IS document type added to the task in step 2.
10. On the Select Source dialog box, do one the following:
 - Click **From File System** and click **Browse** to select a template file located in a file system folder.

Click **Next** if you want to specify an optional name for the e-form template; otherwise, click **Finish** to return to the Add E-form Support dialog box.
 - Click **From Repository** to select from a list of available repository definitions.

Note:

This list will be empty if no webMethods e-form environments have been deployed to the connected Integration Server. For information about creating and deploying e-form environments, see *Implementing E-form Support for BPM*.

Click **Next**.

On the resulting page, locate and click the appropriate template.

Click **Next** if you want to specify an optional name for the e-form template; otherwise, click **Finish** to return to the Add E-form Support dialog box.

11. Click **Next**.
12. On the Task Client Options panel, select or clear the check boxes for the following options as required.

Option	Description
Add a copy of the e-form template to the task	Select this check box to keep a copy of the e-form template in the task application. When a task user downloads the e-form information from the task, the embedded template is used to format the business data in the IS document type to its e-form format. If this option is not selected, the task application connects

Option	Description
	<p>to the specified template location and downloads the template each time the e-form data is downloaded from the task.</p> <p>Consider the following reasons to decide if you want to store a copy of the template in the task application:</p> <ul style="list-style-type: none"> ■ The original template may not be available to the task application at run time. This is particularly true for a template file stored in a local file system, but can be true for any location (repository, URL, or file system). ■ Performance or security reasons that may prevent the task application from accessing the template at run time. ■ You do not want the task application to automatically pick up any changes made to the template in the public repository. <p>The decision should be based on the above considerations, but generally, if the repository is available at run time and any expected template changes will be fully backward compatible, you are advised to avoid using a local copy of the template.</p>
Add download capability to the task Inbox	Select this check box to add a download link to the task inbox results list. This enables the user to download the e-form directly from the inbox, without having to open the task.
Open updated page in Editor when finished	Select this check box if you want to open the page in the task editor after enabling e-form support.

13. Click **Finish**. The changes to the task are saved automatically.

14. If you want to customize the e-form interface, you can edit the page view in the task editor.

Adding an E-form User Interface to the Task Details View with Snippets

You can add a basic default e-form user interface to your task as described in [“Adding E-forms Stored in a My webMethods Server Folder Location” on page 109](#). You can also create a user interface with components available from the Snippets view. This procedure adds the ability to upload and submit an e-form by including Form Submission controls.

The procedure requires adding an e-form content provider and an e-form template provider to the task, as described in [“Adding E-forms Stored in a My webMethods Server Folder Location” on page 109](#).

➤ To add interface controls from the Snippets view to the task details view

1. Open the Task Details default view in the task editor.

2. Open the Snippets view.
3. Expand E-forms.
4. Drag the Download and upload UI snippet onto the view canvas and drop it inside a Form control.
5. In the Variables area of the Insert Template dialog box, specify a value for providerBeanName by typing the Java property name of your e-form content provider.

Note:

To see Java property names in the Bindings View, click the Show Java Type Names toolbar button.

6. Click **Insert**.
7. Make any additional changes to the user interface, as desired.
8. Save your changes.

Adding an E-form Download Link to the Inbox Results View with Snippets

You can add a download link column to your task inbox results list as described in [“Adding E-form Support to a Task” on page 108](#). You can also add a column with a download link to the task inbox results table with a component available from the Snippets view.

The procedure requires adding an e-form content provider and an e-form template provider to the task, as described in [“Adding E-forms Stored in a My webMethods Server Folder Location” on page 109](#).

➤ To add a download link column to the task inbox results view

1. Open the Task Inbox Results default view in the task editor.
2. Open the Snippets view.
3. Expand E-forms.
4. Drag the Download Column for Task Inbox snippet onto the view canvas and drop it into the Search Result Table control.
5. In the Variables area of the Insert Template dialog box, specify a value for providerBeanName by typing the Java property name of your e-form content provider.

Note:

To see Java property names in the Bindings View, click the Show Java Type Names toolbar button.

6. Click **Insert**.
7. Make any additional changes to the user interface, as desired.
8. Save your changes.

Customizing Task E-form Behavior

Like any other task behavior, you can customize how e-form functionality is made available to the task user. For example, you can configure the task so that:

- The task is automatically accepted upon downloading an e-form.
- The **Download** button is disabled if the task is not accepted.
- Downloading an e-form is permitted even if the task is not accepted.

Similar functionality can be applied to the **Upload** button. You can apply these conditions by applying the appropriate binding expressions to the control in question, binding a control property to a task attribute, such as the `AcceptedByCurrentUser` attribute under `TaskInfo`.

Also note that you can use similar functionality to define the file name of the e-form being downloaded. By default, the e-form template file name is applied to the downloaded file, but you can include the task ID or any business data into the downloaded file name.

Extending this ability, you can also implement some logic to check that the uploaded file name matches the task ID or the business data in the file name to ensure that the correct e-form is being processed.

Working with Services in Tasks

Software AG Designer enables you to include services in a task; these services can be invoked implicitly or as a simple task event action. Implicit service invocation is the recommended way to use services.

The Invoke Service action should be used only if:

- The service does not have any output results, or,
- The output results are not important or are not used.

An added service can be executed and used implicitly in any Assignment or Event condition or result configuration panel. When configuring a condition or result, you can select any service output result property just as you would select a Task Info or Task Data field. In this case, the service is executed automatically whenever it is referenced by the Task Engine. For more information, see [“Configuring Simple Task Conditions” on page 145](#).

When executed, the service resolves all bound input parameters (if any). If these parameters changed since the last service invocation, the service updates the output results before they are used.

Note:

This automatic execution behavior is applicable *only* if the service is not used in any Invoke Service action. When a service is called by the Invoke Service event action, its automatic execution behavior is disabled.

For more information about working with services in tasks, see:

- [“Adding a Service to a Task” on page 114](#)
- [“Modifying a Service in a Task” on page 115](#)
- [“Extending a Task with Custom Java Code” on page 116](#)

Adding a Service to a Task


You can add a service to a task by using either of the following methods:

- By dragging an existing Integration Serverservice from the Package Navigator view and dropping it into the Bindings view. For more information, see [“Adding a Service by Dragging and Dropping” on page 114](#).
- By using the web service connector wizard to locate and specify a WSDL file or Integration Server service and add it to a service. For more information, see [“Adding a Service with the Web Service Connector Wizard” on page 115](#).

For more information about invoking services, see [“Working with Services in Tasks” on page 113](#).


Adding a Service by Dragging and Dropping

➤ To add a service to a task by dragging and dropping

1. In the Solutions view, open the task in the task editor by double-clicking the task name. Note that the **Services** entry is displayed in the Bindings view.
2. In the Package Navigator view, locate the service you want to add to the task.
3. Drag the service from the Package Navigator view and drop it on the **Services** entry in the Bindings view.
4. The service is added to the task and is now available for selection with the Invoke Service action.
5. In the Bindings view, expand the newly added service and then expand the Input Parameters entry (if the service has input parameters).
6. Configure any required binding of input parameters to Task Info, Task Data or other service outputs by clicking the input parameter in the Bindings view and clicking the Data Binding tab in the Properties view. Click  to select a binding attribute.

Adding a Service with the Web Service Connector Wizard


➤ To add a service to a task with the web service connector wizard


1. On the UI Development perspective, start the web service connector wizard from any of the following locations:
 - By clicking **File > New > Web Service Connector**.
 - By clicking the **New > Web Service Connector** arrow button in the main toolbar.
 - By clicking the New button on the main toolbar and clicking **Software AG > Composite Applications > Web Service Connector**.
2. On the Select WSDL Location page, click the arrow at the right of the WSDL field to specify if the WSDL source is a local file or an existing IS service:
 - Click **Browse for a Local File** and locate the WSDL file in the resulting file browser dialog box.
 - Click **Choose a Web Service from a Data Provider** to locate an existing IS service.
 - You can also paste or type in the URL of the WSDL.
3. Click **Next** and select an operation within the WSDL file.
4. Continue to click **Next** through the remaining pages, making changes as required.
5. On the Add Managed Bean Reference Property to Other Managed Beans page, click the [TaskName] Rule Context entry for the task you want to add the service to.
6. Click **Finish**.
7. The service is added to the task and is now available for selection with the Invoke Service action.
8. In the Bindings view, expand the newly added service and then expand the Input Parameters entry (if the service has input parameters).
9. Configure any required binding of input parameters to Task Info, Task Data or other service outputs by clicking the input parameter in the Bindings view and clicking the Data Binding tab in the Properties view. Click  button to select a binding attribute.

Modifying a Service in a Task

You can modify a service in a task by editing the various properties of the service using the Bindings view.

> To modify service properties

1. In the Solutions view, double-click the task you want to work with to open it in the editor.
2. In the Bindings view, expand Services and then expand the service you want to work with to view all of the default properties of the service; to see all available properties, click  in the Bindings view toolbar.
3. Click or expand the property element you want to work with; for example:
 - Click **Endpoint Address** and modify the address values as required in the Properties view.
 - Expand **Authentication Info** and modify the following values as required in the Properties view:
 - **User Name**
 - **Password**

When a binding expression is needed, type it into the appropriate box in the Properties view or click  to select a binding expression.
4. Save your modifications.

Extending a Task with Custom Java Code

You can extend the functionality of a task in the following ways:

- By implementing and calling custom Java code added through customization of the task's Services bean. For more information see [“Adding Your Own Java Code” on page 116](#).
- By adding actions and data to the Services bean in the Bindings view. For more information, see [“Adding Actions and Data to a Task in the Bindings View” on page 117](#).

Adding Your Own Java Code

You can extend the functionality of a task by implementing and calling custom Java code through customization of the task's "Services" bean. This Java bean is available for each task, and you can add Java methods as actions and Java getters as properties of this bean.

Java actions can be called from task events. Java getters (for example, functions such as `get<Name>()`, which returns a value) can be used in both assignments and events to deliver data. As an example, a getter may return a list of user IDs for assignment or it may return a value to be used for comparison in event condition or for update in an event action.

> To add custom Java code to a task

1. In the Solutions view, double-click the task you want to work with to open it in the editor.
2. In the Bindings view, double click **Services** or right-click **Services** and click **Open** to open `[TaskName]RuleContext.java` in an editor.

Note:

If you have already added any Java services to the task, or specified actions or data as described in [“Adding Actions and Data to a Task in the Bindings View” on page 117](#), the Java code for those additions is visible here.

3. Add your own custom Java code and save the file.

Adding Actions and Data to a Task in the Bindings View

You can extend the functionality of a task by adding actions (methods) and data (properties) to the Services bean in the Bindings view.

➤ To add actions and data to a task

1. In the Solutions view, double-click the task you want to work with to open it in the editor.
2. In the Bindings view, right-click **Services** and:
 - Click **Add > Action** to open the Add New Action wizard. For more information about working with this wizard, see *webMethods CAF and OpenUI Development Help*.
 - Click **Add > Data** to open the Add New Property wizard. For more information about working with this wizard, see *webMethods CAF and OpenUI Development Help*.
3. To view the results of step 2, double click **Services** in the Bindings view to open `[TaskName]RuleContext.java` in an editor. The new method or value has been added to the file.

Note:

If you have added any Java services to the task, the Java code for those services is visible here also.

Working with Task Privileges

When you create a task, Software AG Designer automatically creates privileges for all of the functionality included in the task; for example, accept a task, assign a task, or modify task information. You can assign these functional privileges to one or more user roles in My webMethods to control the run-time functions available to a user, according to the roles that user is a member of.

Refer to the following topics for information on how to view the available functional privileges for a task, or to add, delete, modify, or bind a privilege:

- [“About Privilege Containers” on page 118](#)

- [“Viewing Task Functional Privileges” on page 118](#)
- [“Adding a Task Functional Privilege” on page 119](#)
- [“Adding a Task Functional Privilege Container” on page 119](#)
- [“Modifying and Deleting a Task Functional Privilege Component” on page 120](#)
- [“Binding a Task Functional Privilege to a User Interface Control” on page 121](#)

About Privilege Containers

By default, all tasks are created with three functional privilege containers, named Task Administration, Task Management, and Rules Management. All of the default privileges are located within these containers.

When you add a functional privilege to a task, you can do it in one of three ways:

- By adding it directly under the task name (that is, without a container).
- By adding it to an existing privilege container in the task.
- By creating a new container within the task and adding the privilege there.

Privilege containers enable you to organize privileges for efficient use in the My webMethods environment, especially when assigning functional privileges to user roles. For example, in My webMethods, you can select a container and thereby add all of the privileges in the container to the user role. Similarly, you can remove all of the functional privileges in a container from a user role by clearing the container selection.

In the functional privileges editor, you can create a container within another container. With foresight and planning, you can structure your functional privilege display for maximum ease of use in My webMethods. For more information about assigning functional privileges in My webMethods, see the PDF publication *webMethods Task Engine User's Guide*.

Viewing Task Functional Privileges

➤ To view task functional privileges

1. In the Solutions view, expand **User Interfaces** and double-click **Functional Privileges**.
2. The functional privileges editor displays all of the tasks contained in the specified project, and the functional privileges associated with each task.
3. You can also view task privileges in My webMethods and assign them to or remove them from user roles. For more information, see the PDF publication *webMethods Task Engine User's Guide*.

Adding a Task Functional Privilege Container

A task is created with three default privilege containers. You can add additional privilege containers to further organize your privilege structure. For more information about containers, see [“About Privilege Containers” on page 118](#)

➤ To add a task functional privilege container

1. Navigate to the functional privileges editor as described in [“Viewing Task Functional Privileges” on page 118](#).
2. In the **Functional Privileges** list, click the task container folder where you want to create a new container.
3. Click **Add Container**.
4. In the New Functional Privilege Container dialog box, type a name and description for the container.
5. Click **OK**. The new container is added to the selected task folder as a folder.

Adding a Task Functional Privilege

➤ To add a task functional privilege

1. Navigate to the functional privileges editor as described in [“Viewing Task Functional Privileges” on page 118](#).

You can add a privilege in one of three ways:

- By adding it directly under the task name (that is, without a container).
- By adding it to an existing privilege container in the task.
- By creating a new container within the task and adding the privilege there.

For more information about containers, see [“About Privilege Containers” on page 118](#).

2. In the **Functional Privileges** list, click the task name, or the name of a container within the task. Click **Add Privilege**.
3. In the New Functional Privilege dialog box, specify the following:

Value	Description
Identifier	A globally unique identification string for the privilege. As this value is used to programmatically identify the privilege, consider using a format that can be easily used within the programming environment. For example, if the name of the privilege is "Modify Purchase Order," create an ID such as "modify.purchase.order".
Name	Your public name for the privilege. This name is displayed in the functional privileges editor and identifies the privilege in My webMethods.
Description	Your description of the privilege. This value appears only in the functional privileges editor; it does not appear in My webMethods.

- Click **Finish** to add the functional privilege to the list.
- Although the procedure above creates a new privilege within the task, it has no functionality associated with it. You must modify a task view to link the privilege with a security role and then provide a binding between the role and one or more controls on the portlet view design canvas. For more information, see [“Binding a Task Functional Privilege to a User Interface Control” on page 121](#).

Modifying and Deleting a Task Functional Privilege Component

Important:

Do not delete any of the default task privileges! You cannot individually delete any of the default privileges. However, it is possible to delete the default privilege containers, which deletes all of the privileges within the default container. Containers and privileges you create can be individually deleted.

➤ To modify or delete a task functional privilege component

- Navigate to the functional privileges editor as described in [“Viewing Task Functional Privileges” on page 118](#).
- In the **Functional Privileges** list, click the privilege or container you want to work with and do one of the following:
 - To modify the privilege or container, type your changes in the **Name** and **Description** boxes and save the changes. The identification information cannot be changed.
 - To delete the privilege or container, click **Delete**.
- Publish the modified task application project to the run-time environment.

Binding a Task Functional Privilege to a User Interface Control

When you create a task, you may want to provide a task privilege to some run-time users and withhold it from others. For example, you may want to allow some users to modify a purchase order by adding attachments, and other users to view those attachments but not be able to add to them. The basic procedure is as follows:

1. Create a new privilege within the task (see [“Adding a Task Functional Privilege” on page 119](#)).
2. Create a specific security role for this privilege (see [“Adding a Security Role for Binding a Task Privilege to a User Interface Control” on page 82](#)). When you create the security role, you link the privilege to the security role.
3. Bind the security role to a user interface control (see [“Binding a Security Role to a User Interface Control” on page 83](#)).
4. You now have the capability of assigning the bound functional privilege to a My webMethods user role during run time. For more information about working with user roles in My webMethods, see the PDF publication *webMethods Task Engine User’s Guide*.

Working with Task Assignments

You can define the assignment behavior of a task by configuring one or more task assignments. With each task assignment you configure, you specify the users, groups, or roles you want the task assigned to, and any conditions that you want to apply to the assignment.

Note:

A user must be explicitly assigned to a task (either directly or through a group or a role) to see the task in any of his inboxes. If no assignments are added to a task, the task will appear only in Task List Management (where it can be assigned to a user, group, or role).

You can create two or more task assignments, and place them in different task control sets; this enables you to select different assignee lists and conditional behavior for different task implementations in the run-time environment. For more information about task control sets, see [“Working with Task Control Sets” on page 148](#).

You have the following choices for defining task assignments:

- You can type in one or more user, group, or role IDs.
- You can select from existing role on a connected My webMethods Server.
- You can select a task information or task business data field that will define the assignee(s).
- You can select from configured security roles (for more information, see [“Working with Security Roles” on page 79](#)).
- You can specify a service that will return task assignment value(s).

- You can specify a webMethods Rules Development decision entity that will return task assignment values.

Task users working in the My webMethods user interface can, with the proper privileges, assign tasks to other users, groups, or role with the Select Principals dialog box. This dialog box is contained in several of the default task portlets as the People Picker Dialog control, which is available in the **MWS Html > Control > MWSX > People** folder of the Palette view. For more information about this control, see *webMethods CAF and OpenUI Development Help*

Creating a Task Assignment

After you create a task you can create one or more task assignments. If you create two or more task assignments, you can place them in different task control sets; this enables you to select different assignee lists and conditional behavior for different task implementations in the run-time environment. For more information about task control sets, see [“Working with Task Control Sets” on page 148](#).

If you are working with a collaboration task, you can preconfigure a task assignment—this is generally done when the collaboration task is to be used inside a business process and you want the task assignment completed automatically.

If the collaboration task is to be used manually (that is, queued by a user from a parent task in the user's inbox) you can opt to omit the preconfigured assignment and let the user assign the collaboration task at creation time.

➤ To create a task assignment

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. Click the **Assignments** tab. Task assignments are displayed in the list in the Assignments area.
3. Click the **Add** button in the Assignments area. A New Assignment entry appears in the assignment list.

Note:

You can add My webMethods Server roles to the Assignments area by dragging them from the MWS Admin view and dropping them into the Assignee list.

You can now configure the following assignment attributes:

- General information about the assignment (see [“Configuring Assignment General Information” on page 123](#)).
- Assignees (see [“Configuring Assignees for a Task Assignment” on page 123](#)).
- Option and condition settings [“Configuring Task Assignment Options and Conditions” on page 132](#)).

- Distribution management settings (see [“Configuring Distribution Management for a Task Assignment” on page 133](#)).

Configuring Assignment General Information

➤ To configure assignment general information

1. Open the task in the task editor by double-clicking the task in the Solutions view.
2. Click the **Assignments** tab of the task editor and click the assignment you want to work with in the Assignments list, or double-click the assignment in the Assignments area on the **Overview** tab.
3. Expand the General Information area if it is not already visible.
4. Specify the following values:
 - **Assignment Name** — Type the display name for the assignment. You must save the task to see this name applied in the Assignment list. This name is also displayed in My webMethods. Assignment names cannot contain the following characters: < > ' & | \ / ;
 - **Assignment Description** — Type a description of the assignment. This text is also displayed in My webMethods.
 - **Task Control Set** — Type a new task control set name, or select an existing task control set name from the drop-down list. For more information on task control sets, see [“Working with Task Control Sets” on page 148](#).

Configuring Assignees for a Task Assignment

You can specify one or more assignees for a task; when task assignment is triggered in the run-time environment, the task will be assigned to all assignees listed here. You can create two or more task assignments, and place them in different task control sets; this enables you to select different assignee lists for different task implementations in the run-time environment. For more information about task control sets, see [“Working with Task Control Sets” on page 148](#).

Note:

You can also create and modify task assignments by defining a task event and configuring it to include the **Assign to Users** task action. When the event is triggered, the task assignments are modified according to the task action definition. For more information, see [“Creating a Task Event” on page 137](#) and [“Configuring Simple Task Actions” on page 140](#).

➤ To configure assignees for a task assignment

1. Open the task in the task editor by double-clicking the task in the Solutions view.

2. Click the **Assignments** tab of the task editor and click the assignment you want to work with in the Assignments list, or double-click the assignment in the Assignments area on the **Overview** tab.
3. Expand the Assignee area if it is not already visible.
4. Click the **Add** button in the Assignee area. The Assignment Target Selection dialog box appears.
5. In the list on the left, click the type of assignee you want to define:

- **Role** — Use this entry if you want to choose from a list of existing roles on a connected My webMethods server. Click the browse button to select a role.

Note:

You can add My webMethods Server roles by dragging them from the MWS Admin view and dropping them into the Assignee list.

- **Role UID** — User this entry to type the name of the My webMethods Server role or click the browse button to select a role definition from the task business data. Use this method if the role does not yet exist on My webMethods Server.
- **User UID** — Use this entry to type the name of a My webMethods Server user, or click the browse button to select a user definition from the task business data or a task info field.
- **Security Role** — Click or type the name of an available security role (for more information, see [“Working with Security Roles” on page 79](#)).

Note:

Do not specify a security role that has been configured for binding a functional privilege to a control. Use only security roles that are linked to My webMethods Server roles.

- **Group UID** — Use this entry to type the name of a My webMethods Server user, or click the browse button to select a group definition from the task business data or a task info field.
 - **Service** — Use this entry if you want to specify a service that produces a list of roles, user names, user IDs, or URI's as the output result. Click the browse button to select the "Output Results" item of a service (these services must be previously defined in (added to) the Bindings view).
 - **Business Rule** — Use this entry if you want to use a webMethods Rules Development decision table or business rule set to define the assignment values. For more information, see [“Configuring a Task Assignment with a Decision Table” on page 125](#) and [“Configuring a Task Assignment with a Business Rule Set” on page 126](#).
6. Click **OK**.
 7. Repeat steps 4-6 to add additional assignees.
 8. Save the task.

Configuring a Task Assignment with a Decision Table

When you determine the assignments for a task with a webMethods Rules Development decision table, you place the logic for determining the assignment outside of the task and the Task Engine. Any changes to the rule logic in the decision table will automatically be propagated to all of the tasks that use the decision table. Otherwise, if you implement assignments within the task, you must update each task assignment individually.


For an example of using a decision table to provide a task assignment, see [“Example Task Assignment with a Decision Table” on page 128](#). For more information about decision tables and their operation, see the *webMethods BPM Rules Development Help*.


Note:

The Rules Explorer displays the rule projects and decision tables you have defined in your local Designer workspace. If you have no rule projects or decision tables defined, the lists will be empty.

To work with decision tables, you must have workspace indexing enabled (**Window > Preferences > Software AG > Workspace Index**). You must re-start Designer to apply a change to this setting.

➤ To configure a task assignment with a decision table

1. From the Solutions view, open the task with which you want to work.
2. Click the **Assignments** tab.
3. Next to the list of assignments, click **Add** to add a new assignment and then select the new assignment.
4. In the Assignee area, next to the Assignee List, click **Add** and select **Business Rule**.
5. In the **Business Rule** field, click .
6. In the **Project** field, select the rules project with which you want to work.
7. Select **Use Decision Table** and select a decision table in the drop-down list.
8. Click **Next**. Designer populates the list of action input parameters with the data model of the decision table.
9. Specify values for the action input parameters with which you want to work by selecting an input field and doing one of the following:
 - Type a value to assign to a data model element.

- Click  in the Bindings column for a data model element. In the Expression Binding dialog box, select a value from the task business data or task info to assign to the data model element.

Note:

Not all fields in the task business data and task info represent conditions and results in the rules. You must be familiar with the conditions and results requirements of the decision table to select the correct fields. For an example, see [“Example Task Assignment with a Decision Table” on page 128](#).

10. Click **Next**. Designer populates the list of rule output fields with the data model of the decision table.
11. Specify the rule output fields that you want to use as the rule results.
 - a. In the Selection column, click in the table cell for the element with which you want to work.
 - b. Click **Yes** in the drop-down list to specify that the data model element will contribute to the rule results.

When the rule executes, the selected elements provide the assignment value back to the task.

12. Click **Finish**.

Note:

Before you can run the task in your run-time environment, you must export the rule project that contains the selected decision table to the target run-time environment to ensure that the task assignment operates as expected. For more information about exporting rule projects, see the *webMethods BPM Rules Development Help*.

Configuring a Task Assignment with a Business Rule Set

When you determine a task’s assignments with a webMethods Rules Development business rule set, you place the logic for determining the assignment outside of the task and the Task Engine. Any changes to the rule logic in the business rule set will automatically be propagated to all of the tasks that use the business rule set; otherwise, if you implement assignments within the task, you must update each task assignment individually.

A business rule set typically contains two or more decision tables or other rule metaphors; the presence of multiple rule metaphors expands the potential use of the business rule set. For example, this is useful in cases where you want two or more decision tables to interact with each other, with the results from one decision table used as input for another decision table (both within the same business rule set). Or, in a case where you want to create logically-related decision tables that apply only to order processing, you can create an order processing business rule set and use it as a container for all decision tables that apply to order processing.

The selection of input parameters and results parameters is similar to the method used for a single decision table (see [“Example Task Assignment with a Decision Table” on page 128](#)). However,



with a business rule set, the parameters contained in all decision tables are available for selection. For more information about business rule sets, decision tables, and their operation, see the *webMethods BPM Rules Development Help*.

Note:

The Rules Explorer displays the rule projects and business rule sets you have defined in your local Designer workspace. If you have no rule projects or rule sets defined, the lists will be empty.

To work with business rule sets, you must have workspace indexing enabled (**Window > Preferences > Software AG > Workspace Index**). You must re-start Designer to apply a change to this setting.

➤ **To configure a task assignment with a business rule set**

1. From the Solutions view, open the task with which you want to work.
2. Click the **Assignments** tab.
3. Next to the list of assignments, click **Add** to add a new assignment and then select the new assignment.
4. In the Assignee area, next to the Assignee List, click **Add** and select **Business Rule**.
5. In the **Business Rule** field, click the browse button .
6. In the **Project** field, select the rules project with which you want to work.
7. Select **Use Rule Set** and select a rule set in the drop-down list.
8. Click **Next**. Designer populates the list of action input parameters with the data models of all rule metaphors in the rule set.
9. Specify values for the data model input parameters with which you want to work. When these data model elements are matched, the rules will execute. Select an input field and do one of the following:
 - Type a value to assign to a data model element.
 - Click the browse button  to open the Expression Binding dialog box and select a value from the task business data or task info to assign to a data model element.

Note:

Not all fields in the task business data and task info represent conditions and results in the rules. You must be familiar with the conditions and results requirements of the business

rule set to select the correct fields. For an example, see [“Example Task Assignment with a Decision Table” on page 128](#).

10. Click **Next**. Designer populates the list of rule output fields with the data models of all rule metaphors.
11. Specify the rule output fields that you want to use as the rule results.
 - a. In the Selection column, click in the table cell for the element with which you want to work.
 - b. Click **Yes** in the drop-down list to specify that the data model element will contribute to the rule results.

When the rule executes, the selected elements provide the assignment value back to the task.

12. Click **Finish**.

Note:

Before you can run the task in your run-time environment, you must export the rule project that contains the selected business rule set to the target run-time environment to ensure that the task assignment operates as expected. For more information about exporting rule projects, see the *webMethods BPM Rules Development Help*.

Example Task Assignment with a Decision Table

The following scenario describes an example of configuring a task to obtain its assignment values from a webMethods Rules Development decision table. For implementation information, see [“Configuring a Task Assignment with a Decision Table” on page 125](#). For more information about working with rule projects, see the *webMethods BPM Rules Development Help*.

Suppose you have a task that processes incoming orders. You want to use a decision table to return assignment values that will:

- Assign the task to the correct North American, Central American, or South American order desk when the order is less than \$10,000.
- Assign the task to the corresponding Regional Sales Manager if the order is \$10,000 or more.

In this case, the task business data contains these fields:

- TaskOrderRegion— identifies the originating region.
- TaskOrderAmount— identifies the order amount.

The decision table data model contains these fields:

- RuleRegion
- RuleAmount
- RegionalOrderDesk

■ RegionalSalesManager

The decision table itself looks like this:

	RuleRegion	RuleAmount	RegionalSalesManager	RegionalOrderDesk
1	= NorthAmerica	< 10000		= NA_OrderDesk
2	= NorthAmerica	>= 10000	= peter_jones	
3	= CentralAmerica	< 10000		= CA_OrderDesk
4	= CentralAmerica	>= 10000	= enrico_salazar	
5	= SouthAmerica	< 10000		= SA_OrderDesk
6	= SouthAmerica	>= 10000	= elena_cortez	

The rules in the decision table look for these region conditions:

- RuleRegion=NorthAmerica
- RuleRegion=CentralAmerica
- RuleRegion=SouthAmerica

and for these order amount conditions:

- RuleAmount<10000
- RuleAmount>=10000

and define these results:

- RegionalSalesManager=peter_jones (Note: This is a *user ID*. You can specify multiple user IDs with a comma-separated list. To assign a task to multiple user IDs, use a String List data type for the assignee field of your IS document type.)
- RegionalOrderDesk=NA_OrderDesk (Note: This is a *role*.)
- RegionalSalesManager=enrico_salazar
- RegionalOrderDesk=CA_OrderDesk
- RegionalSalesManager=elena_cortez
- RegionalOrderDesk=SA_OrderDesk

In your assignment configuration, you map your task info to the decision table data model input elements:

- RuleRegion=TaskOrderRegion
- RuleAmount=TaskOrderAmount

For the assignment results, you would select both of the following fields from the decision table data model:

- RegionalOrderDesk

■ RegionalSalesManager

When the decision table executes, it will examine the region and amount of the order and return an assignment role (for orders under \$10,000) or an assignment user ID (for orders of \$10,000 or more).

Example Task Assignment with a Business Rule Set

You can configure a task assignment to be set with a business rule set defined with the webMethods Rules Development feature. These rule sets typically contain two or more rule metaphors, such as decision tables and event rules. When you invoke the business rule set with the proper data, the decision tables, event rules, and other rule metaphors contained in the business rule set will return assignment results; you can also configure the business rule set to provide other actions as well.

The following scenario describes an example of configuring a task to obtain its assignment value from a webMethods Rules Development business rule set that also calculates a discount percentage and sends a notification message.

- For more information about how decision tables work, see [“Example Task Assignment with a Decision Table” on page 128](#)
- For implementation information, see [“Configuring a Task Assignment with a Business Rule Set” on page 126](#).
- For more information about working with rule projects, see the *webMethods BPM Rules Development Help*.

Suppose you have a task that processes incoming support requests. You want use a business rule set to return a support role assignment value (Expert, Senior, or Junior) based on:

- The number of purchases the customer has made.
- The price of the items purchased by the customer.
- The customer’s enrollment status (Gold, Silver, or Regular)

In this case, the task business data contains these fields:

- prevPurchases— identifies the number of previous customer purchases.
- itemPrice— identifies the dollar amount of the item purchased.
- custStatus—identifies the customer’s enrollment status.

This business data is passed to the business rule set, which consists of two decision tables and an event rule (in this example, the business data fields are mapped to rule fields of the same name):

Decision Table 1

	prevPurchases	itemPrice	discountTier
1	> 10	> 50	= Tier1
2	5 <= ... < 11	> 50	= Tier2
3	1 <= ... < 6	> 50	= Tier3
4		<= 50	= Tier4

Decision Table 2

	custStatus	discountTier	discPerCent	support
1	= Gold	= Tier1	= 20	= Expert
2	= Gold	= Tier2	= 20	= Expert
3	= Gold	= Tier3	= 15	= Senior
4	= Gold	= Tier4	= 10	= Senior
5	= Silver	= Tier1	= 15	= Senior
6	= Silver	= Tier2	= 10	= Junior
7	= Silver	= Tier3	= 5	= Junior
8	= Silver	= Tier4	= 2	= Junior
9	= Regular		= 1	= Junior

Event Rule

Event			
Whenever a	support	Event Type	is changed

Results	
1	supportChange = SendNotification()

Decision Table 1 uses the prevPurchases and itemPrice business data fields to determine the discountTier value (this is the output of Decision Table 1).

Decision Table 2 uses the custStatus business data field plus the discountTier output from Decision Table one to determine two results: a discount percentage (discPerCent) and a support role name (Expert, Senior, or Junior). The business rule set returns the support result value to the task to set the task assignment. The discPerCent field is also available for use in additional task transactions, as needed.

As time passes, the customer may make additional purchases. This increases the value of prevPurchases, which may result in a change in support level for the customer. The event rule in the business rule set is configured so that any time the support value changes, a notification is sent to the customer, stating that the customer's support level has been upgraded to the next level.

Configuring Task Assignment Options and Conditions

You can specify the options and conditions that govern the assignment of a task; when task assignment is triggered in the run-time environment, it will be assigned to all assignees listed based on these options and conditions.

You can create two or more task assignments, and place them in different control sets; this enables you to select different assignee lists and conditional behavior for different task implementations in the run-time environment. For more information about control sets, see [“Working with Task Control Sets” on page 148](#).

Note:

If you do not define any options and conditions, task assignment occurs with the default setting "Evaluate once when task is queued." No conditions or distribution management are applied. For more information about distribution management, see [“Configuring Distribution Management for a Task Assignment” on page 133](#).

➤ To configure options and conditions for a task assignment

1. Open the task in the task editor by double-clicking the task in the Solutions view.
2. Click the **Assignments** tab of the task editor and click the assignment you want to work with in the Assignments list, or double-click the assignment in the Assignments area on the **Overview** tab.
3. Expand the Assignment Options and Conditions area if it is not already visible.
4. Specify the following options and conditions:
 - **Assignment Options** — This option specifies when the task is assigned to the members of the assignee list:
 - **Evaluate once when the task is queued** — In this case, the task assignment is made only when the task is first queued in the run-time environment. No further assignment activity will occur. This is the default setting.
 - **Evaluate for every task change** — In this case, a task assignment occurs each time a change is applied to the task in the run-time environment (for example, task submittals, status changes, and priority changes). This option is useful if assignment depends on the task business data and changes to the data should result in an assignment change.
 - **Execute from Re-assign event only** — In this case, a task assignment occurs only when an associated task event with a re-assign action is triggered. For more information about task events, see [“Working with Task Events” on page 136](#).
 - **Condition Type** — This option specifies whether you want to create conditions with the simple conditions editor (default), or use the advanced condition expression editor. For more information, see:

- [“Configuring Simple Task Conditions” on page 145](#)
- [“Configuring Advanced Task Expressions” on page 243](#)

5. Save the task to apply your changes.

Configuring Distribution Management for a Task Assignment

You can specify how tasks are distributed when task assignment is triggered in the run-time environment.

➤ To configure distribution management for a task assignment

1. Open the task in the task editor by double-clicking the task in the Solutions view.
2. Click the **Assignments** tab of the task editor and click the assignment you want to work with in the Assignments list, or double-click the assignment in the Assignments area on the **Overview** tab.
3. Expand the Distribution Management area if it is not already visible.
4. In the **Distribution Management** box, select the management scheme you want to apply:

Option or Check Box	Available Selections
No distribution management	Use this entry if you do not want to apply any distribution management (default setting). In this case, task assignment is governed only by the assignee list and any defined assignment conditions.
Round robin	<p>With this setting, tasks are assigned sequentially among the available users who are members of the roles or groups in the assignee list.</p> <ul style="list-style-type: none"> ■ Select the Check Availability Against User Calendar check box to assign the task to the next available user when the next user in the rotation is not available. ■ For more information about round-robin distribution, see “About Round-Robin Task Distribution” on page 134. For more information about user availability by calendar, see “Working with User and Business Calendars” on page 237 and “About User Calendars and Task Distribution Management” on page 135.
Queue length	<p>Use this entry to assign the task to the role or group-based assignee with the shortest queue length (that is, with the fewest assigned tasks).</p> <ul style="list-style-type: none"> ■ Click the Use User's Default Inbox option to determine the user's queue length directly from the contents of each user's inbox

Option or Check Box	Available Selections
---------------------	----------------------

- | | |
|--|--|
| | <ul style="list-style-type: none"> ■ Click the Use Saved Search option to determine the user's queue length by applying a saved search to the user's inbox. Click the browse button to select a saved search from the Expression Binding dialog box. Select from the public Saved Searches available on the connected My webMethods Server. The number of assigned task instances will be determined by the results of the selected saved search. For more information about working with saved searches, see <i>webMethods BPM Process Development Help</i>. ■ Select the Check Availability Against User Calendar check box to assign the task to the next available user when the user with the shortest queue length is not available. ■ Select the Ignore Other Task Types check box if you want the queue length determination to be made for this task type only. |
|--|--|

For more information about queue length distribution, see [“About Queue Length Task Distribution” on page 135](#). For more information about user availability by calendar, see [“Working with User and Business Calendars” on page 237](#) and [“About User Calendars and Task Distribution Management” on page 135](#).

Note:

If you select round-robin or queue length distribution, the assignee list must contain *only* roles, security roles, groups, or services that return role or group values. Individual user names cannot be in the assignee list for these distribution management methods. Specified roles must be static roles; distribution management does not support LDAP query, database, or rule-based roles.

5. Save the task to apply your changes.

About Round-Robin Task Distribution

Round-robin task distribution is a method for distributing instances of a task as equally as possible among a task's assignees. You can configure a task assignment to use round-robin distribution as described in [“Configuring Distribution Management for a Task Assignment” on page 133](#).

Note:

For round-robin, the assignee list must contain *only* roles, security roles, groups, or services that return role or group values. Individual user names cannot be in the assignee list for the round-robin distribution management methods.

Round-robin distribution considers the users that are members of a role or group in the assignee list when making task assignments. For example, when a task assignee list contains a role with multiple users, instances of the task are evenly distributed among all users in the role. The

distribution steps through the list of users and then repeats the sequence — thus the name, "round-robin."

When the assignee list contains two or more roles, tasks are evenly distributed among all roles, then users within roles.

For example, if the assignee list contains three roles with two users in each role:

- Task instance 1 is assigned to Role1 User1, Role 2 User1, and Role 3 User 1
- Task instance 2 is assigned to Role1 User2, Role 2 User2, Role 3 User 2
- Task instance 3 is assigned to Role1 User1, Role 2 User1, Role 3 User 1
- Task instance 4 is assigned to Role1 User2, Role 2 User2, Role 3 User 2

And so on. Other considerations for round-robin distribution include:

- When a role or group contains another (nested) role or group, the nested object is evaluated in sequence with all other users in the assignee list, just as if it were a user entry.
- When an assignee list contains one or more security roles that are linked to My webMethods Server roles, those security roles are handled exactly like other roles.

About Queue Length Task Distribution

Queue length task distribution is a method for distributing instances of a task based on the number of task instances already assigned to role or group-based task assignees. You can configure a task assignment to use queue length distribution as described in [“Configuring Distribution Management for a Task Assignment” on page 133](#).

Note:

For queue length, the assignee list must contain *only* My webMethods Server roles, security roles, groups, or services that return role or group values. Individual user names cannot be in the assignee list for the queue length distribution management methods

Generally, queue length distribution assigns a task to the user that has the fewest number of tasks (that is, the shortest queue length).

About User Calendars and Task Distribution Management

For round-robin and queue length distribution management methods, you can further refine task assignment by selecting the **Check Availability Against User Calendar** option; if the logically selected user is unavailable, the assignment evaluation process moves on to the next logical user. The user's availability determined using both My webMethods Server business calendars and a user's personal calendar (assuming these calendars are defined and configured).

When availability checking is selected, the calendar checking process is as follows:

The next working day is determined by applying the task queue time to the user's business calendar (as defined in My webMethods). For example, suppose that the user's business calendar defines

working days as Monday through Friday, 9 A.M. - 5 P.M. In this case, the next working day is calculated as following:

Task Queue Time	Result in the following Next Working Day assignment:
Friday, 8 A.M.	Friday
Friday, 11 A.M.	Friday
Friday, 5:01 P.M.	Monday (the Friday business day has ended, so the next business day is now Monday)

All tasks queued after 5:00 P.M. Friday and before 5:00 P.M. Monday will be given a working day of Monday. Any holidays configured in the business calendar are also considered.

When a task is being assigned to or accepted for another user, the system checks the user's personal calendar to determine if the user is available on this working day, taking into consideration only Out of the Office and Busy types of calendar events that are *scheduled for the entire day*. All other event types and durations are ignored. For example if a user has scheduled meetings marked as Busy during the day, the task is still assigned to the user if there are any available times between the meetings.

You can also view a user's personal calendar on the Task List Management page in My webMethods when you manually assign a task or accept a task for another user.

For more information about creating and working with business calendars and personal user calendars, see the PDF publication *webMethods Task Engine User's Guide*.

Working with Task Events

You can define the behavior of a task when certain task event types occur by configuring one or more task events. With each task event you configure, you specify:

- The event type that will trigger an action (for example, the task is updated or completed).
- Any conditions that you want to apply to the event type (for example, the presence of certain business data values).
- The type of action you want to occur, such as set the task priority, or send an e-mail notification.

You can create two or more task events and place them in different control sets; this enables you to select different events and conditional behavior for different task implementations in the run-time environment. For more information about control sets, see [“Working with Task Control Sets” on page 148](#).

Performance Tip

Quite often, task events are used to trigger an action that updates the task. In this case, each event is a relatively large transaction and it is much more efficient to create multiple update actions inside a single event whenever possible.

For example, you might need to update several pieces of business data for a task when it is queued from a process. If you need to update more than a single property, avoid creating multiple task events, and instead create a single task event with several Set Business Data Field actions within it.

Creating a Task Event

After you create a task you can create one or more task events.

➤ To create a task event

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. Click the **Events** tab. Task events are displayed in the list in the Events area.
3. Click the **Add** button in the Events area. A New Event entry appears in the assignment list.
4. The task is automatically saved with the new event, using the default event name.

You can now carry out the following event configuration activities:

- Modify the event name, description, and control set, as described in [“Configuring Event General Information” on page 137](#).
 - Specify an event type and conditions as described in [“Configuring a Task Event Type” on page 138](#).
 - Specify a pre-defined action that will be triggered by the event as described in [“Configuring Simple Task Actions” on page 140](#) (event actions are not available for any of the Filter Inbox event types).
- or
- Define an advanced action expression that will triggered by the event as described in [“Configuring Advanced Task Expressions” on page 243](#).

Configuring Event General Information

➤ To configure event general information

1. Open the task in the task editor by double-clicking the task in the Solutions view.
2. Click the **Events** tab of the task editor and click the event you want to work with in the Events list, or double-click the event in the Events area on the **Overview** tab.
3. Expand the General Information area if it is not already visible.

4. Specify the following values:

- **Event Name** — Type the display name for the event. You must save the task to see this name applied in the Events list. This name is also displayed in My webMethods. Event names cannot contain the following characters: < > ' ' & | \ / ;
- **Event Description** — Type a description of the event. This text is also displayed in My webMethods.
- **Task Control Set** — Type a new control set name, or click an existing control set name from the drop-down list. For more information on control sets, see [“Working with Task Control Sets” on page 148](#).

Configuring a Task Event Type

You can specify one or more event types for a task and associate an action that occur when the event type is triggered in the run-time environment. You can create two or more task events, and place them in different control sets; this enables you to select different event behavior for different task implementations in the run-time environment.

- For more information about control sets, see [“Working with Task Control Sets” on page 148](#).
- For information about how to create a task event, see [“Creating a Task Event” on page 137](#).

➤ To configure a task event type

1. Open the task in the task editor by double-clicking the task in the Solutions view.
2. Click the **Events** tab of the task editor and click the event you want to work with in the Events list, or double-click the event in the Events area on the **Overview** tab.
3. Expand the Event Type area if it is not already visible.
4. In the **Event Type** box, select the type of event you want to use. The following selections are available:
 - Select **Advanced** to create an expression to define a custom event type (for more information, see [“Configuring Advanced Task Expressions” on page 243](#)). You cannot create advanced expressions for filtering or scheduling event types.
 - The following event types define a simple change in the task: **Accepted, Assigned, Canceled, Comment Added, Comment Deleted, Comment Updated, Completed, Delegated, Description Updated, Name Updated, On Error, Priority Updated, Queued, Reassigned, Resumed, Suspended, Task Deleted, and Task Updated**. A matching occurrence of any of these event types triggers the event.
 - **Business Data Updated** — Specify a business data field from the task's business data. The event is triggered when the specified field is modified.

- **Custom ID Updated** — This event triggers when the custom ID value (part of the Task Info) is updated.
 - **Date/Time Event** — Specify parameters to define the date/time event. The event is triggered when the specified date and time is matched. For more information about the date and time settings, see [“Configuring a Date/Time Event” on page 142](#).
 - **Expiration Date Reached** — The event is triggered when the task's specified expiration date (part of Task Info) has occurred.
 - **Expiration Date Updated** — The event is triggered when the task's specified expiration date (part of Task Info) is updated.
 - **Filter Inbox - Accepted Tasks** — This filter shows only tasks that are accepted by this user or not yet accepted by any other user. No additional configuration or extra conditions are needed, although you can define them optionally. No actions are associated with this event type.
 - **Filter Inbox - Custom Filter** — Specify extra conditions to define the Inbox Filter conditions you want to apply. The condition should return true to show the task in the user inbox. No actions are associated with this event type. For more information, see [“Filtering a Custom Inbox with a Task Event” on page 144](#).
 - **Filter Inbox - User Activity** — This filter shows only tasks the user has not yet worked on. This filter requires the selection of a custom business data field or a service to provide a list of users who have worked on the task. You must also configure this field or service to be updated manually by the Task Details portlet or another event action. No actions are associated with this event type.
5. If you want to define additional conditions for the selected event type, add them to the Extra conditions list as described in [“Configuring Simple Task Conditions” on page 145](#).
 6. You must now define an action that will occur when the task event type is matched, as described in [“Configuring a Task Event Action” on page 139](#).

Configuring a Task Event Action

You must specify an action that occurs when the task event type is triggered in the run-time environment. For information about how to create a task event, see [“Creating a Task Event” on page 137](#).

➤ To configure a task event action

1. Open the task in the task editor by double-clicking the task in the Solutions view.
2. Click the **Events** tab of the task editor and click the event you want to work with in the Events list, or double-click the event in the Events area on the **Overview** tab.
3. Expand the Event Actions area if it is not already visible.

4. In the **Action Type** box, select List of Simple Actions if you want to create actions from the list of simple actions (default), or select Advanced Action Expression to use the advanced action expression editor. For more information, see:
 - [“Configuring Simple Task Actions” on page 140](#)
 - [“Configuring Advanced Task Expressions” on page 243](#)
5. Save the task.

Configuring Simple Task Actions

You can specify the actions that govern the behavior of a task when a task event is triggered in the run-time environment by choosing from a list of predefined (simple) actions.

Note:

If you find you cannot create the action you want with the simple task actions that are available, you can create an advanced (custom) action expression; for more information, see [“Configuring Advanced Task Expressions” on page 243](#).

➤ To configure simple task actions

1. Open the task in the task editor by double-clicking the task in the Solutions view.
2. Click the **Events** tab of the task editor and click the event you want to work with in the Events list, or double-click the event in the Events area on the **Overview** tab.
3. Expand the Event Actions area if it is not already visible.
4. Click **Add** to add an action. In the Task Action Selection dialog box, select one of the following actions:
 - **Accept** — Causes the task to be accepted by the specified user when the task is queued. Do one of the following:
 - Type a user ID in the **User ID** box.
 - Click the browse button next to the box to select a user ID binding value from the task business data or task information.
 - Click the browse button next to the box to select a service that has been added to the task; the output value of the service is used to define user ID. For more information about adding a service to a task, see [“Adding a Service to a Task” on page 114](#).
 - **Assign to Users** — Assigns the task to one or more of the specified users, groups, or roles. Click the browse button to specify an assignee. The method of defining assignees is the same as that described in [“Configuring Assignees for a Task Assignment” on page 123](#).

- Select the **Append to assignee list** check box to add the assignees in the action to the existing task assignees.
- Clear the **Append to assignee list** check box to replace the existing task assignees with the assignees in the action (default).
- **Cancel Task** — Places the task in Canceled status.
- **Complete Task** — Places the task in Completed status.
- **Complete Parent Task** — Places the parent task in Completed status. This action is available only when a parent task is defined on the Overview tab of the task editor (that is, the task is a collaboration task).
- **Expire Task** — Places the task in Expired status.
- **Invoke Service** — Executes the specified service. For specific information about this action, see [“Adding a Service to a Task” on page 114](#).
- **Queue Child Task** — Starts a collaboration task created from the specified task type, and named with the specified task name (a Browse button is available to define the task name from a business data field or from a service). This action is available only when a task is named as a parent task by one or more other tasks. It is available in both the parent and child tasks.
- **Re-assign** — Re-assigns the task to a specified entry in the task assignee list.
- **Resume Task** — Resumes a suspended task.
- **Send Notification** — Executes the specified task notification.
- **Set Business Data Date Field** — Use this action to set the values for a date field selected from the data available in the task. The data field, start date, and business calendar values must be selected as a binding to existing task data; you cannot type a value. If the task is the child of a parent task, you can specify a binding to a value contained in the parent task business data; this is useful to update a child task with data from the parent task.
- **Set Business Data Field** — Use this action to set a value for a data field selected from data available in the task. The value you set can be selected as a binding to existing task data or typed as a value. If the task is the child of a parent task, you can specify a binding to a value contained in the parent task business data; this is useful to update a child task with data from the parent task.
- **Set Parent Business Data Date Field** — Use this action to set the values for a date field in a parent task. The data field, start date, and business calendar values must be selected as a binding to existing parent or child task data; you cannot type a value. This action is available only when the task is the child of a parent task, and is typically used to update a parent task date field to reflect a child task date field.
- **Set Parent Business Data Field** — Use this action to set a value for a data field in a parent task; the data field you select must be selected as a binding to existing parent task data; you cannot type a value. The value you set for the data field can be selected as a binding to existing parent or child task data or typed as a value. This action is available only when

the task is the child of a parent task, and is typically used to update a parent task business data field to reflect a child task business data field.

- **Set Task Custom ID** — Sets the task custom ID to the specified value.
- **Set Task Description** — Sets the task description to the specified value.
- **Set Task Expiration** — Sets the expiration date for the task based on the defined date and time values. Note that this action does not place the task in Expired status, it only specifies a point in time when the task is considered expired. You can create an event type of Expiration Date Reached with an Expire Task action to actually place the task in Expired status.
- **Set Task Name** — Sets the task name to the specified value.
- **Set Task Priority** — Sets the priority of the task to the specified value.
- **Suspend Task** — Places the task in Suspended status.

5. Click **OK** to add the action.
6. Repeat steps 4-5 if you want to add another action.
7. Save the task.

Configuring a Date/Time Event

You can define a date/time event for a task, as described in [“Configuring a Task Event Type” on page 138](#).

» To define a date/time event type

1. Select a Date/Time event type as described in [“Configuring a Task Event Type” on page 138](#). The controls for defining the event type appear in the Date/Time Event area.
2. Specify a start date for the date/time event by clicking the browse button next to the **Choose Start Date** label. The start date is the base date that determines how the date/time event is evaluated. For example, if you want an event to trigger on a specific date, click that date as the start date. If you want the event to trigger after a specified number of days, the interval is determined beginning with the start date.

Note:

If you are using a defined date from the Task Info (such as creation date), click the browse button and select it. If you want to use a custom date, you must first create a custom business data field defining the date. For more information on defining custom business data fields, see [“Adding A Business Data Field with the Editor” on page 94](#) and [“Adding Task Business Data Fields to a Page or View” on page 102](#).

3. If you want a business calendar to calculate the date/time interval, select the **Use Business Calendar** check box. If you select this check box, click the browse button next to the check box to specify a corporate (business) calendar defined in My webMethods Server. For more information about business calendars, see [“Working with User and Business Calendars” on page 237](#).
 - You can also click a business data field to use as a business calendar; in this case, when you pick a business data field for calendar, the field value is expected to contain the Lookup Name of the desired calendar, as defined on the Business Calendars administration page in My webMethods.
 - If you do not use a business calendar, any date/time interval will be calculated using all days of the week (including weekends and holidays).
4. If you leave the **Days After Start Date** value set to zero, the event will trigger when the start date is matched.

Note:

Date/time events are not precise; the approximate accuracy is about 1 minute. For example, if you configure a date/time event to trigger exactly at the task creation date, the actual event may be triggered up to 1 minute later than the task is queued

If you want the event to trigger after a certain number of days, do one of the following:

- Type a positive integer value in the **Days After Start Date** box (0 to 365).
- Click the browse button next to the box to select a binding value from the task business data or task information.
- Click the browse button next to the box to select a service that has been added to the task; the output value of the service is used to define the Days After Start Date value. For more information about adding a service to a task, see [“Adding a Service to a Task” on page 114](#).

Note:

When you define a Task Date/Time event and base the StartDate on the output of a service, it will call the service every time the scheduler runs and for each task instance being tested (typically every Active task for this task type).

5. If necessary, fine tune the trigger time by specifying a positive or negative offset from the **Hours After Start Date** and **Minutes After Start Date** boxes. For example, to trigger the event at 10 P.M. (22:00), you could count to the next day and then set a negative offset of -2 hours, or you could count to the previous day and set an offset of +22 hours.

You can also click the browse button next to the box to select a binding value or service, as described in step 4.

6. Continue defining the event actions as described in [“Configuring a Task Event Action” on page 139](#).

Filtering a Custom Inbox with a Task Event

You can use the Filter Inbox - Custom Filter task event to apply display filtering to any custom task inbox you create. For example, you can configure the event so that the custom inbox displays only tasks with a status of Active and a priority of High. Custom filtering overrides any runtime search criteria specified in the task SearchBar portlet. For example, if you define filtering to show only tasks with priority = High, and the user sets the Priority field = Medium in the Search For window and clicks **Search**, the resulting search will return no results.

Note:

The task event filter mechanism is not available for use with My Inbox or Task List Management. Also, it is recommended for use only in custom task inboxes that are implemented with the standard search provider. If used in conjunction with the indexed search provider, the performance and memory advantages of indexed searching will be lost.

➤ To define a Filter Inbox - Custom Filter event type

1. Open the task in the task editor by double-clicking the task in the Solutions view.
2. Click the **Events** tab of the task editor and click the event you want to work with in the Events list, or double-click the event in the Events area on the **Overview** tab.
3. Expand the Event Type area if it is not already visible.
4. In the **Event Type** box, select the **Filter Inbox - Custom Filter** event.
5. In the extra conditions area, define the Inbox Filter conditions you want to apply. The condition must return “true” to display the task in the custom inbox. If you define multiple conditions, specify how you want them to be evaluated:
 - ALL of the conditions are true
 - ANY of the conditions is true
 - ONLY ONE of the conditions is true

If you create multiple conditions, use the **Up** and **Down** buttons to define their order. Careful consideration can help to improve performance. For example, if you are filtering for tasks with status = active and priority = HIGH, placing the condition for HIGH priority first will likely result in a smaller set of matches than the active status condition.

Note:

No actions are associated with this event type.

6. Save the task.

Creating a Task Notification Event

You can create a task notification that My webMethods users can subscribe to, as described in [“Working with Task Notifications” on page 185](#). By default, the user must be both assigned to the task and subscribed to the notification to receive the notification.

Note:

You can remove the “assigned to” requirement for users. For more information, see [“Sending Task Notifications to Non-Assigned Users” on page 194](#).

After you create a task notification, you must create a task notification event to trigger the notification; without a task notification event, the notification will never be sent. A task notification can be triggered as a single result in an event action, or it can be combined with other actions.

Note:

You must create at least one task notification for a task type before you can create a task notification event. For more information, see [“Creating a Task Notification” on page 191](#).

➤ To create a task notification event

1. Create a task event as described in [“Creating a Task Event” on page 137](#). You can create a notification action for all event types except filter events.
2. If you want to create additional conditions to apply to the event type, define them in the Extra Conditions area. For more information, see [“Configuring Simple Task Conditions” on page 145](#).
3. In the Event Actions area, click **Add** to open the Task Action Selection dialog box.
4. Click **Send Notification**.
5. In the **Notification** box, select the task notification you want to use (if the list is empty, you must first create a task notification as described in [“Creating a Task Notification” on page 191](#)).
6. Click **OK**.
7. Save the task.
8. Publish the task application project containing the task to make the task notification and subscription available to My webMethods users.

Configuring Simple Task Conditions

You can specify the conditions that govern the assignment or event actions of a task; when the task assignment or event is triggered in the run-time environment, it will be subject to these conditions.

Note:

If you do not define any conditions, the task assignment or event action occurs depending on the basic Assignment Options or Event Type setting. For more information, see [“Configuring Task Assignment Options and Conditions” on page 132](#) and [“Configuring a Task Event Type” on page 138](#). If you find you cannot create the condition you want with the simple task conditions that are available, you can create an advanced (custom) condition expression; for more information, see [“Configuring Advanced Task Expressions” on page 243](#).

➤ To configure simple task conditions

1. Open the task in the task editor by double-clicking the task in the Solutions view.
 - For assignments, click the **Assignments** tab and click the assignment you want to work with. Expand the Assignment Options and Conditions area if it is not already visible. In the **Condition Type** box, click Joined List of Simple Conditions. The Simple conditions editor appears immediately below the list.
 - For events, click the **Events** tab and click the event you want to work with. Expand the Event Type area if it is not already visible. In the **Event Type** box, click any of the available event types except Advanced. The Extra conditions editor appears immediately below (for some event types, an additional options editor may appear as well).

2. Specify the following condition settings:

Note:

The **Join** options are disabled unless two or more conditions are defined.

- **Join** — Click one of these three options to specify how the conditions will be evaluated:
 - **ALL are true** — All of the listed conditions must be true to trigger the assignment.
 - **ANY is true** — Any of the listed conditions must be true to trigger the assignment (that is, it can be one, two, or any subset of the condition list).
 - **ONE only is true** — A single listed condition must be true to trigger the assignment.
3. Click **Add** to add a condition. The Add Simple Condition dialog box appears.
 - In the **Field** box, click the browse button and select the task data field you want to base the condition on.
 - In the **Operation** box, select the operator you want to apply to the condition.
 - In the **Value** box, type a value you want the Field value to be evaluated against, or click the browse button to select a task data field.
 4. Click **OK** to add the condition.
 5. Repeat steps 2-4 if you want to add another condition.

6. Save the task to apply your changes.

Modifying Simple Task Conditions

You can modify and delete the conditions that govern task assignments and events.

➤ To modify a simple task condition

1. Open the task in the task editor by double-clicking the task in the Solutions view.
 - For assignments, click the **Assignments** tab and click the assignment you want to work with. Expand the Assignment Options and Conditions area if it is not already visible.
 - For events, click the **Events** tab and click the event you want to work with. Expand the Event Type area if it is not already visible.
2. In the **Conditions** box, select the simple condition you want to modify.
3. Click **Edit**. The Edit Simple Condition dialog box appears.
4. Modify the condition settings as described in [“Configuring Simple Task Conditions” on page 145](#).
5. Save the task to apply your changes.

Deleting Simple Task Conditions

➤ To delete a simple task assignment condition

1. Open the task in the task editor by double-clicking the task in the Solutions view.
 - For assignments, click the **Assignments** tab and click the assignment you want to work with. Expand the Assignment Options and Conditions area if it is not already visible.
 - For events, click the **Events** tab and click the event you want to work with. Expand the Event Type area if it is not already visible.
2. In the **Conditions** box, select the simple condition you want to delete.
3. Click **Delete**. The condition is removed from the task.
4. Save the task to apply your changes.

Rearranging the Order of Simple Task Conditions

You can rearrange the order of the simple conditions that govern task assignments and events.

When the Any join option is applied, conditions are evaluated in order, beginning at the top of the condition list, until a condition is matched. To speed up performance, you can, for example, move the conditions most likely to be matched to the top of the list, and place the least likely matches at the bottom.

Because you can call a service as part of a condition definition, you may want to create several conditions that search for simple matches, and if they fail, you can place a service with a more sophisticated matching algorithm at the bottom of the condition list.

➤ To rearrange the order of a simple task condition

1. Open the task in the task editor by double-clicking the task in the Solutions view.
 - For assignments, click the **Assignments** tab and click the assignment you want to work with. Expand the Assignment Options and Conditions area if it is not already visible.
 - For events, click the **Events** tab and click the event you want to work with. Expand the Event Type area if it is not already visible.
2. In the **Conditions** box, select the simple condition you want to move.
3. Click the **Up** or **Down** button to relocate the selected condition in the list. Although this functionality is available for all join options (All, Any, or One), it has significance only for the Any join option.
4. Save the task to apply your changes.

Working with Task Control Sets

You can define one or more control sets that are associated with a task, although there is not a requirement for a task to have a control set. Control sets are designed to enable you to create different combinations of assignments and events that can be selected for different implementations of a task.

For example, suppose you have the following scenario:

- You receive incoming orders from two differently branded web sites.
- The basic tasks of the order handling process itself are the same after the order is entered.
- The assignees and actions for each task vary depending on the originating web site.

To address this scenario, you can configure a task that defines the basic order handling user interface and behavior, and then you can define a separate control set for each set of assignees and events.

When the task is integrated into a process, you can specify which control set applies to the task when it runs within that particular process. This enables you to create "basic" tasks and then add control sets that adapt them for specific usages.

- Each assignment and event you create can be added to a control set; you can define as many control sets as necessary, containing any combination of assignments and/or events.
- An assignment or event can belong to only one control set. If you want the same assignment or event behavior in more than one control set, you must create a new assignment or event that copies the behavior of the original and associate it with another control set.
- When an assignment or event belongs to no control set, it is always executed when the task is invoked. If you want an assignment or event to always apply when the task is invoked, do not assign it to a control set (as opposed to duplicating it in all control sets).
- If no control sets are defined, then all of the task's assignments and events are executed whenever the task is invoked.

For information on how to specify a control set when adding a task to a process, see *webMethods BPM Process Development Help*.

Viewing Task Control Sets

You can view task control set availability in the following locations:

- **Overview tab** — Open the task in the task editor by double-clicking the task in the Solutions view, then click the **Overview** tab. The Assignments and Events areas each contain a sortable column labeled **Task Control Set**. This column indicates which control set the assignment or event is associated with. Click the column heading to sort the assignments or events by control set.
- **Assignments and Events tabs** — Open the task editor and click the **Assignments** or **Events** tab. The General area contains a box labeled **Task Control Set**. This box indicates which control set the assignment or event is associated with.

Creating a Task Control Set

You can create a control set and associate task assignments and events with the control set.

➤ To create a task control set

1. Open the task in the task editor by double-clicking the task in the Solutions view.
 - For assignments, click the **Assignments** tab and click the assignment you want to work with. Expand the General area if it is not already visible.
 - For events, click the **Events** tab and click the event you want to work with. Expand the General area if it is not already visible.

2. In the **Task Control Set** box, type the name of the control set you want to create. Control set names cannot contain the following characters: < > ' ' & | \ / ;)
3. Save the task.
4. The assignment or event you are working with is now associated with the control set you created, and is available for use with other assignments and events.

Associating an Assignment or Event with a Control Set

➤ To associate a task assignment or event with a control set

1. Open the task in the task editor by double-clicking the task in the Solutions view.
 - For assignments, click the **Assignments** tab and click the assignment you want to work with. Expand the General area if it is not already visible.
 - For events, click the **Events** tab and click the event you want to work with. Expand the General area if it is not already visible.
2. In the **Task Control Set** box, select the name of the control set.
3. Save the task.
4. The assignment or event you are working with is now associated with control set you selected.

Removing an Assignment or Event from a Control Set

➤ To remove a task assignment or event from a control set

1. Open the task in the task editor by double-clicking the task in the Solutions view.
 - For assignments, click the **Assignments** tab and click the assignment you want to work with. Expand the General area if it is not already visible.
 - For events, click the **Events** tab and click the event you want to work with. Expand the General area if it is not already visible.
2. In the **Task Control Set** box, delete the name of the control set, or select the empty entry in the drop-down list.
3. Save the task.
4. The assignment or event you are working with is now removed from control set you selected.

Note:

If you remove all assignments and events from a control set, the control set is removed from the task. A task cannot support an empty control set.

Changing the Task Type Name

You can change the task type name of a task after the task has been created. The task type name identifies the task in the Solutions view and in other locations in Software AG Designer, and anywhere the Task Type name is displayed in My webMethods. For more information, see [“About Task Names and Task Type IDs” on page 31](#).

If the task you want to rename has already been published to My webMethods Server, you must republish the modified task to see the changes in My webMethods.

Note:

Although it is possible to create task types with duplicate names, you must take additional steps if you apply a task type name that is already in use. For more information, see [“About Task Names and Task Type IDs” on page 31](#), especially the section .

➤ To change the task type name

1. Open the task in the task editor by double-clicking the task in the Solutions view.
2. Click the **Overview** tab.
3. In the General Information area, type the new name for the task in the **Task Type Name** box.
4. Save the task. In most cases, you will want to change the task type inbox name as described in [“Renaming a Task Root Page” on page 181](#).

Note:

If the task type is used in a process, consider changing the task step name in the process as well. The task type and task step names do not have to match, as the task is programmatically tracked by the Task Type ID number. However, best practices recommend that you keep the display names synchronized. Note that in the Process Designer, the task name is displayed on both the **General** tab and on the **Implementation** tab.

Using Gadgets to Render the Task Detail Page in webMethods Business Console

Instead of using Composite Application Framework portlets, you can use gadgets to render the task detail page in webMethods Business Console. To do so, you add a gadget to an existing task in Designer.

Before adding a gadget to a task, you must:

1. Create the task in a task application project. For more information about creating tasks, see [“Creating Tasks” on page 39](#).
2. Create one or more gadgets in the task application project. For more information about creating gadgets for webMethods Business Console, see *webMethods CAF and OpenUI Development Help*.

➤ To add a gadget to a task

1. Open the task in the task editor by double-clicking the task in the Solutions view.
2. On the **Overview** tab, in the User interfaces area, click **Gadgets**, and then click **Attach**.
3. In the **Gadgets** field of the Attach Task Detail Gadget wizard, select a task detail gadget from the drop-down list.
4. Click **OK**.

Designer displays the gadget in the Gadgets folder.

Note:

You can attach only one gadget to a task. If you want to attach another gadget, you must first delete the previously attached gadget from the Gadgets folder.

9 Updating Tasks

■ Task Update Overview	154
■ Updating Tasks Applying Optional Settings	155

Task Update Overview

After a task instance starts running in the run time, you can update the task instance over the life cycle of the task. This can be done manually, through the task user interface in My webMethods, or programmatically, through the action of both internal and external APIs:

- Java APIs that are available within the task portlet application. For more information, see [“Available Java APIs” on page 154](#).
- Services that are included as part of the WmTaskClient package installed on Integration Server:
 - Built-in services
 - RESTful services
 - Web services

For more information about the services in the WmTaskClient package, see [“Working with the Task Engine APIs” on page 259](#).

About Task Engine Optional Settings

In addition, there are several Task Engine optional settings that apply to task updates in the following areas:

- Synchronization of task updates in a cluster environment.
- Task locking during an update.
- Task locking timeout and duration.
- Updating a completed task.

Be sure to review the behavior of these optional settings regarding your environment. For more information, see [“Updating Tasks Applying Optional Settings” on page 155](#).

Available Java APIs

When you are working with task portlet applications that run in My webMethods Server/Task Engine, best results are obtained by using the embedded Task Engine APIs as much as possible. These APIs are created when you create a task application portlet.

These APIs exist as Java classes in the package `com.webmethods.caf.faces.data.task.impl`. For more information, see the Javadoc pages for this package and its classes, available on the Empower Product Support Website.

In some cases, it is not possible to use these APIs (for example, from an application that is external to My webMethods Server, from a business process model, or from an IS service). In these instances, you can use the Task Engine APIs from the WmTaskClient package. For more information, see [“Working with the Task Engine APIs” on page 259](#)).

Updating Tasks Applying Optional Settings

Several optional Task Engine settings apply when you update tasks with the available Web and built-in services:

Note:

In addition to the following options, you are advised to review all of the available options described in the PDF publication *webMethods Task Engine User's Guide*.

- **-Dtask.update.sync.** Cluster synchronization is not enabled by default. This option helps to solve the problem that when a task is updated, there is a small (up to 2 second) delay until other nodes in the cluster reflect the change. This is normally not an issue for standard task interfaces, but it can present a problem for remote Task Engine Web service APIs, as well as services in the WmTaskClient package. For more information, see the PDF publication *webMethods Task Engine User's Guide*.
- **-Dtask.lock.wait.timeout and -Dtask.lock.valid.time.** These two options specify both the time that the Task Engine will wait to obtain a lock on a task instance (default=10 seconds), as well as the duration of the lock (default=60 seconds). For more information, see the PDF publication *webMethods Task Engine User's Guide*.
- **-Dupdate.completed.task.** This option enables a remote service to update tasks regardless of task status. By default, you can update only tasks in Active, Error, and Suspended state. For more information, see the topic "Task Update Behavior" in the PDF publication *webMethods Task Engine User's Guide*.

10 Working with Task Searches

■ Task Search Overview	158
■ Implementing Case-Insensitive Searching	158
■ About Indexed Searches	159
■ About HPSTRA Searches	160
■ Search Considerations with Task Type Inboxes	161
■ Implementing an Indexed Search Content Provider for Existing Task Types	161
■ Searching Indexed Data in a Custom Task Inbox	165
■ Implementing a Standard Search Provider for Existing Task Types	167
■ Considerations When Publishing Tasks with Indexed Data	168

Task Search Overview

When you create a new task type, you can select from the following task search providers:

- *Default search provider*
- *Task indexed search provider*
- *HPSTRA search provider*

You can select the indexed task search provider when you create a new task, using the **Select search content provider** drop-down list on the New Task wizard. By default, the task is created with a standard search provider. You can change the search provider type at any time after a task has been created.

The standard search provider enables the task user to search across all business data in all the task instances available in the inbox by executing a complete in-memory scan of all of the business data fields for those task instances. If the number of task instances is relatively small, this search executes quickly. However, when a large number of tasks are present, a standard (full) search can take significantly longer.

The indexed task search provider enables you to optimize search performance. This provider searches only business data fields that are specified as indexed, making the search much faster.

The HPSTRA search provider enables you to search tasks data, indexed in an Elasticsearch store. In addition to standard task searches, this search provider ensures high-performance full-text search on the task information and mapped task data fields for a large number of tasks.

Implementing Case-Insensitive Searching

By default, searches for custom task fields are case-sensitive. For example, a search for the term `Order` will return different results than a search for `order`. If you have a task application that requires case-insensitive searches, a public API method (`caseInsensitive`) is available to enable individual custom search fields to provide case-insensitive behavior.

Note:

Prior to carrying out the following procedure, you must first create your task application as described in [“Creating a New Task with the Task Wizard” on page 40](#), and then add custom search fields to it as described in [“Creating Custom Search Fields from Task Business Data” on page 104](#).

You must carry out this procedure for each field that requires case-insensitive behavior.

➤ To define a field as case-insensitive

1. In the Project Explorer view, expand the task application project you want to work with.
2. Open the task inbox results default view Java file in the editor. For example:

```
com.webmethods.caf.<taskName>_inboxresults >
```

```
<taskName>_InboxResultsDefaultviewView.java
```

3. Locate:

```
private static final String[][] TASKSEARCHPROVIDER_PROPERTY_BINDINGS =
new String[][]
```

and add the following binding:

```
{"#{taskSearchProvider.searchQuery.<fieldName>.caseInsensitive}", "true"},
```

4. Save the file.

Alternative Approach

As an alternative to the method above, you can modify the task search provider Java file, as follows:

1. Open the task search provider Java file in the editor. For example:

```
com.webmethods.caf.<taskName>_inboxresults >
<taskName>_SearchProvider.java
```

2. Add `<fieldName>.setCaseInsensitive(true);` to the get action for the field you want to apply case-insensitive behavior to. For example, if you are working with the field `FirstName`:

```
public com.webmethods.caf.faces.data.task.impl.TaskSearchQueryTerm
getFirstName() {
    if (FirstName == null) {
        FirstName = new
            com.webmethods.caf.faces.data.task.impl.TaskSearchQueryTerm();
        FirstName.setCaseInsensitive(true);
    }
    return errorMessage;
}
```

3. Save the file.

About Indexed Searches

To optimize searching when many tasks are present, you can configure the task application that contains the task to use an indexed search provider to identify specific non-list business data fields as *indexed fields*, rather than use the standard search.

When a task type is enabled for indexed searches/indexed business data fields, an indexed field table is created in the run-time database of the Task Engine when the task application is published to the runtime. In this case, a search considers only the indexed business data fields in the table and ignores all other fields, resulting in a much faster search.

Obviously, an indexed field search can return only the business data fields which are indexed. If a field is not indexed, a search for that field will return no results.

Task developers are advised to give careful thought as to which search scheme is implemented for a given task type.

- For more information about working with indexed business data fields in a task type, see [“About Indexed Business Data Fields” on page 89](#).
- For more information about creating a task type with an indexed search content provider, see [“Creating a New Task with the Task Wizard” on page 40](#).
- For information about searching for indexed data fields in a custom task inbox (including searching multiple inboxes), see [“Searching Indexed Data in a Custom Task Inbox” on page 165](#).
- For more information about publishing task types with indexed fields to the runtime environment, see [“Considerations When Publishing Tasks with Indexed Data” on page 168](#).
- For general information about the task search content providers, see:
 - [“Task Search Content Provider” on page 255](#)
 - [“Task Indexed Search Content Provider” on page 254](#)

About HPSTRA Searches

When you enable a task type for HPSTRA-indexing, all task information fields and mapped business data fields are indexed in the Elasticsearch store. Business data fields are mapped for HPSTRA indexing by default, but at task design time, you can select which fields to exclude from the index. All task business data is stored in the Task Engine database, regardless of your HPSTRA mapping.

At task runtime, when you search My Inbox, Task Engine queries the Task Engine database. However, when searching a custom task inbox that uses the HPSTRA search provider with a HPSTRA-enabled task, Task Engine queries the Elasticsearch store. Enabling HPSTRA-indexing for a task type does not automatically add the HPSTRA search provider to the generated task inbox. You can enable a task type for HPSTRA-indexing, persist task information and business data fields to an Elasticsearch store, and use a custom solution to search and retrieve the task data.

By default, the `includeTaskData` property for the search provider in a HPSTRA task inbox is set to `false`, and with this setting the search retrieves only the task information fields. To include task business data in the search results for custom task inboxes, you can set the `includeTaskData` property to `true`. When the property is set to `true`, Task Engine retrieves the task information from the Elasticsearch store, and the task business data - from the Task Engine database. Search results are retrieved faster than those from the default search provider, even though the results include all matching fields, including the fields not mapped for Elasticsearch indexing.

Example: Searching in HPSTRA Inboxes

At runtime, the HPSTRA task inboxes provide an input field for full text searches. Full text search results are affected by the indexing type that you select when enabling HPSTRA for a task type, as described in [“Modifying Task Business Data Fields” on page 96](#).

The following search scenarios apply for a HPSTRA-enabled task business data field with the field name 'Customer Name', and value 'Maria Anders'.

When the business data field is indexed with type 'analyzed':

- The Elasticsearch index contains 'maria', 'anders'
- A full text search for 'Anders' returns one result
- A search filter with search term 'Anders' returns zero results

When the business data field is indexed with type 'not_analyzed':

- The Elasticsearch index contains no index value for the field
- A full text search for 'Anders' returns zero results
- A search filter with search term 'Anders' returns zero results
- A search filter with search term 'Maria Anders' returns one result

When the business data field is indexed with type 'stored':

- The Elasticsearch index contains no index value for the field
- A full text search for 'Anders' returns zero results
- A search filter with search term 'Anders' returns zero results
- A search filter with search term 'Maria Anders' returns zero results

However, you can retrieve the value of the 'Customer Name' field within the result set for another query, for example, when selecting all customers from a city.

For more information about Elasticsearch index types, see the Elasticsearch documentation.

Search Considerations with Task Type Inboxes

It is possible to configure the properties of a task type inbox to specify a limit to the number of tasks returned in the search results, or to select the **No Maximum** check box, which effectively returns all matching tasks (note that this check box is not available in My Inbox or Task List Management).

When selected, the maximum is subject to the `task.max.results` option (default = 1000 tasks); for more information, see “Limit the Number of Tasks Returned to a Results List” in the PDF publication *webMethods Task Engine User’s Guide*.

When the task type associated with the task type inbox is configured with an indexed search provider, clearing or selecting the **No Maximum** check box has no effect; all search results are always returned, and the value defined in the `task.max.results` option is ignored.

Implementing an Indexed Search Content Provider for Existing Task Types

There are two cases where you might want to change from a standard search content provider to an indexed search content provider:

- You have created a task with a standard search content provider, but now want to use an indexed search content provider.
- You have imported tasks from Designer Version 8.0 SP2 or earlier into your current workspace; earlier versions did not offer indexed business data, and you want to add this capability.

Note:

You also have the option of implementing both search types. The indexed search provider is part of the task inbox results portlet, so it is possible to create more than one custom inbox portlet and implement a different type of search in each one.

➤ **To implement an indexed search content provider in an existing task type that uses the standard search content provider**

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. On the **Overview** tab, expand the User Interfaces area if it is not already visible.
3. Expand the Task Inbox Results portlet and click the default view.
4. Click the **Update** button in the User Interfaces area. The Task UI Update wizard appears with the current properties for the selected element displayed.
5. Click the **Update inbox to support indexable fields for selected task UI page** option.
6. Click **Next**.
7. If you want to open the view in the view editor after applying the indexable field option, select the **Open updated page in Editor** check box.
8. Click **Finish**. The changes to the task are saved automatically.

If the task you are working with is implemented with the default portlets and user interface configurations (that is, no customization work has been done on it), no further work should be necessary. Otherwise, *you are strongly advised* to check the following areas before you publish the task type:

- [“Changing the Search Provider Base Class” on page 162](#)
- [“Setting the Sorting and Value Bindings for Custom Inbox Business Data Columns” on page 163](#)
- [“Setting the Sorting Bindings for Custom Inbox TaskInfo Columns” on page 164](#)
- [“Setting the Bindings Configuration for Task Search Provider” on page 165](#)

Changing the Search Provider Base Class

The Update Task Client Options wizard attempts to modify the name of the `<TaskName>SearchProvider` class in the `com\webmethods\caf\<taskType>inboxresults` package:

- It changes the class name from `TaskInboxSearchContentProvider` to `TaskInboxIndexedSearchContentProvider`.
- For `CustomInboxSearchQuery`, it changes the class name from `TaskInboxSearchContentProvider.InboxSearchQuery` to `TaskInboxIndexedSearchContentProvider.InboxSearchQuery`.

If the task application has been customized, the wizard may not be able to make these changes. If you have customized packages, locate these class names in your custom Java code and change them as noted above.

Setting the Sorting and Value Bindings for Custom Inbox Business Data Columns

The Update Task Client Options wizard attempts to modify the sorting and value bindings configuration for custom inbox columns that display task business data.

- The sorting value is used in the search results table column "Sort" property, which defines a binding expression for sorting
- The value binding is used anywhere in the controls to display the value of business data field

The following table shows an example of sorting and binding values:

Business Data Column	Standard Searches (Value and Sorting)	Indexed Searches
<code>order.orderDetails.orderNo</code>	<code>#{row.taskData.order.orderDetails.orderNo}</code>	<code>#{row.taskData["orderNo"]}</code>
Where "orderNo" is the database field name given to the business data field in the index configuration.		

If the task application has been customized, the wizard may not be able to make these changes. If you have customized packages, locate these sorting and binding values and change them as noted above.

The binding expression to be used for either Task Info or Task Business data fields can be viewed on the **Data Binding** page of the Properties view; to do so:

1. Open the default view of the Task Inbox Results portlet (or equivalent customized portlet) in the task editor.
2. In the Bindings view, select **Managed Beans > [TaskName][PortletName]/default > Task Search Provider > Run-time Info > [ColumnName]**.
3. Click the **Data Binding** page of the Properties view.

Setting the Sorting Bindings for Custom Inbox TaskInfo Columns

The Update Task Client Options wizard attempts to modify the sorting bindings configuration for custom inbox columns that display TaskInfo fields (value bindings configuration for TaskInfo fields does not change)

The configuration of sorting binding expressions is different depending on the type of Task Search Provider that is used. For some TaskInfo fields, the data being returned undergoes additional translation before it is displayed in the inbox table. For example, the value for priority is translated to the displayed priority name (medium > Medium). The same translated values are used for table sorting.

For tasks using the standard search, this sorting is done by the table control at the custom inbox application level. When an indexed search is used, the sorting is done directly in the database; therefore, sorting on translated values is not possible, and the binding expression for the table column "sort" property needs to be modified to use the original field value instead of the translated value.

The following table lists the result table columns that require and support modifications:

Results Table Column	Standard Search Expression	Indexed Search Expression
DelegatedTo	<code>{activePageBean.taskDisplayProvider.delegatedToCurrentUser}</code>	NOT SUPPORTED
DelegatedFrom	<code>{activePageBean.taskDisplayProvider.delegatedFromCurrentUser}</code>	NOT SUPPORTED
TaskID	<code>{activePageBean.taskDisplayProvider.taskIDNumber}</code>	<code>{row.taskInfo.taskID}</code>
Priority	<code>{activePageBean.taskDisplayProvider.priorityDisplayName}</code>	<code>{row.taskInfo.priority}</code>
CreatedBy	<code>{activePageBean.taskDisplayProvider.createdByDisplayName}</code>	<code>{row.taskInfo.createdBy}</code>
LastModifiedBy	<code>{activePageBean.taskDisplayProvider.lastModifiedDisplayName}</code>	<code>{row.taskInfo.lastModifiedBy}</code>
TaskTypeName	<code>{activePageBean.taskDisplayProvider.taskTypeDisplayName}</code>	<code>{row.taskInfo.taskTypeID}</code>
AssignedToList	NOT SUPPORTED	NOT SUPPORTED

Results Table Column	Standard Search Expression	Indexed Search Expression
AcceptedByList	NOT SUPPORTED	NOT SUPPORTED

If the task application has been customized, the wizard may not be able to make these changes. If you have customized packages, locate these sorting and binding values and change them as noted above.

These values can be viewed on the **Sorting** page of the Properties view; to do so:

1. Open the default view of the Task Inbox Results portlet (or equivalent customized portlet) in the task editor.
2. In the task editor, expand the Search Results Table, and click the column you want to work with.
3. Click the **Sorting** page of the Properties view

Setting the Bindings Configuration for Task Search Provider

The Update Task Client Options wizard attempts to modify the bindings configuration for the paging size property of the search provider to the page size used by table control. By default, the table control is configured to use the binding `#{activePreferencesBean.initialPageSize}` for the **Rows** property field; this sets the page size for the table, but it could be customized, in which case the wizard will not be able to modify it.

It is important to set paging size property of the search provider to either the same value (or a greater value) than the one used by the table. If you specify a greater value, there will be an additional performance overhead for fetching items in the database for each paging operation.

This value can be viewed on the **General** page of the Properties view; to do so:

1. Open the default view of the Task Inbox Results portlet (or equivalent customized portlet) in the task editor.
2. In the task editor, expand the Search Results Table, and click the column you want to work with.
3. Click the **Sorting** page of the Properties view.

Searching Indexed Data in a Custom Task Inbox

When you create a custom task inbox (also known as a task type inbox), you can add a Task Inbox Results portlet to the portlet application to display search results.

For example, you can use the following search provider bindings expressions:

```
{ "#{taskSearchProvider.searchQuery.myField.fields}", "MyIndex"},
{ "#{taskSearchProvider.searchQuery.myField.operator}", "="},
{ "#{taskSearchProvider.searchQuery.myField.value}",
  "#{activePageBean.activeSearchQuery.refineFields[\"myField\"]}" },
```

where “MyIndex” is the indexed field database name, and “myField” is the search term.

Note:

By default, searches for custom task fields are case-sensitive. If you want to enable case-insensitive searches, see [“Implementing Case-Insensitive Searching” on page 158](#).

Searching Indexed Data in Multiple Task Inboxes

By default, searches in the custom task inbox are limited to the task type associated with the custom task inbox. If the task type uses a standard search provider, you can configure the Task Inbox Results portlet to search all task types. For more information, see [“Task Client Option Definitions” on page 170](#).

However, if the task type uses an indexed business data search provider, you must use the following procedure to extend the search to other task types.

Note:

This procedure can also be applied to task types with a standard search provider if you want to limit the number of task types in the search.

If you plan to use the “search all task types” feature with task types that have indexed business data, be aware of the following points:

- For consistent search results, all task types with indexed data must use identical indexed field names. For example, indexed field names such as `Order_number`, `order_number`, and `orderNumber`, are seen by the Task Engine as three different fields. A search against any one of them will not include results from the other two fields. Task developers must take care to ensure that indexed field names are consistent across task types. For example:

Suppose the task type “OrderClaim” contains a field “OrderZipCode”, and the task type “CustomerClaim” contains a field “CustomerZipCode”. If you want to search both task types together for the zip code value, you must ensure that these differently named fields have the same index name (such as “ZipCode”).

- Similarly, best practices dictate that all indexed data fields of the same name contain information of the same type. For example, an indexed field named `SalesPriority` might have some entries in numerical format (“3”) and some in text format (“Medium”). For consistent results, avoid this situation.
- If the named indexed data field is used for different field types in other task types, all tasks containing the indexed data field name will be included in the search results, regardless of the field type.

Important:

Do not select the task option **Search all task types when searching for task type inbox results** when implementing the following procedure. Doing so will cause a search to fail and will generate an error. For more information about this option, see [“Modifying Client Options for a Task” on page 171](#) and [“Task Client Option Definitions” on page 170](#).

Note:

With the standard search provider, the API supports multiple fields per term, and it is possible to modify the Task Search Results view .java file to specify multiple search fields per term. *This is not supported for the indexed search provider; you are limited to one field per term.*

➤ **To search for an indexed business data field across multiple task types**

1. In Designer, add the indexed data field to the task search interface, as described in [“Creating Custom Search Fields from Task Business Data” on page 104](#).
2. Click **Navigate > Open Type**.
3. In the Open Type dialog box, type the task type name that you want work with.
4. In the resulting list, double-click the entry for:

```
taskTypeNameInboxResultsDefaultviewView
```

5. In the resulting text editor, locate the following method in the .java file:

```
private static final String[][] TASKSEARCHPROVIDER_PROPERTY_BINDINGS
```

6. Locate the task type ID value definition containing the current task type ID:

```
{"#{taskSearchProvider.searchQuery.taskTypeID.value}", "taskTypeID1"},
```

7. To specify additional task type IDs to include in the search, add the task type IDs for those task type to the task type ID value definition, separated by commas. For example:

```
{"#{taskSearchProvider.searchQuery.taskTypeID.value}", "taskTypeID1, taskTypeID2, taskTypeID3"},
```

Important:

Do not remove the task type ID value definition. With standard searches, removing this element enables a search of all task types. However, with an indexed business data field search, removing this element will cause an error.

8. Save the file. The task type is now enabled for searching for the indexed business data field in the specified task types.

Implementing a Standard Search Provider for Existing Task Types

There may be cases where you want to switch back to a standard search content provider after you have implemented an indexed search content provider.

Note:

You also have the option of implementing both search types. The indexed search provider is part of the task inbox results portlet, so it is possible to create more than one custom inbox portlet and implement a different type of search in each one.

If such cases, you must manually convert the search provider settings in your task type from indexed to standard; to do so, you must *reverse* the changes described in the following procedures:

- [“Changing the Search Provider Base Class” on page 162](#)
- [“Setting the Sorting and Value Bindings for Custom Inbox Business Data Columns” on page 163](#)
- [“Setting the Sorting Bindings for Custom Inbox TaskInfo Columns” on page 164](#)
- [“Setting the Bindings Configuration for Task Search Provider” on page 165](#)

Considerations When Publishing Tasks with Indexed Data

The presence of indexed business data in a task type has a distinct impact on the run-time environment when you publish a task type that is enabled for indexed searches. *You must be aware of these considerations before you begin publishing indexed search tasks to the runtime to avoid negative affects.*

Indexed business data fields are stored in a database table in the run-time environment. These table entries are searched when an business data search is executed by the user. There is one database table for each task type that you publish to the runtime.

When a task type with indexed business data is published for the first time, the Task Engine creates the database table with one column for each indexed field in the task type. You must be aware of the following important issues when you publish the task type thereafter:

- If you have modified the task type interface, assignments, events, and so on, but have made no changes to the indexed data fields, there is no impact in the run-time environment.
- If you have made any change to the indexed business data (for example, add or remove a field, or change a field type or attribute), the Task Engine detects this automatically, drops the existing index table for the task type (and all its data), and creates a new (empty) table.

To populate the newly created table with data from any existing task instances in your system, you must run the task reindexing procedure as described in the PDF publication *webMethods Task Engine User’s Guide*. This means that *for the time period between publishing the modified task type and completion of the reindexing procedure, any searches run from a task instance of this task type will return incomplete results or no results at all.*

You are advised to consider the impact of publishing task types with indexed business data very carefully. For production environments, you must publish updates to task types with modified indexed business data and reindex business data only during scheduled maintenance periods.

- For more information about task business data reindexing, see *webMethods Task Engine User’s Guide*.
- For more information about indexed business data, see [“About Indexed Business Data Fields” on page 89](#).

11 Working with Task Client Options

■ About Task Client Options	170
■ Task Client Option Definitions	170
■ Modifying Client Options for a Task	171

About Task Client Options

The run-time behavior of a task client running in My webMethods is determined in part by the settings for several task client options.

For a list of the available task client options, see [“Task Client Option Definitions” on page 170](#); for information about setting task client options, see [“Modifying Client Options for a Task” on page 171](#)

Task Client Option Definitions

The following task client options are available (for information about setting these task client options, see [“Modifying Client Options for a Task” on page 171](#)):

- For the Default view of the Task Inbox Search Results portlet:
 - **Auto-accept task upon opening in task type inbox results** (disabled by default). When this option is enabled, a task is automatically accepted by the user when the task is opened from a task type inbox results (does not apply to My Inbox). You can enable this option when you create a task, and you can modify this option by updating the default view of the Task Inbox Results portlet, if one is included in the task.
 - **Show completed tasks in the task type inbox results** (disabled by default). When this option is enabled, completed tasks continue to be visible in the task type inbox results (does not apply to My Inbox). When disabled, completed tasks are removed from the inbox results upon completion. Applies to custom task type inboxes only. Set this option by updating the default view of the Task Inbox Results portlet.
 - **Search all task types when searching for task type inbox results** (disabled by default). When this option is enabled, a task search includes all task types, not just the task type associated with the inbox. Applies to custom task type inboxes only. Set this option by updating the default view of the Task Inbox Results portlet.

Important:

Do not select this option if you have enabled the task type to use an indexed business data search provider.

If the task type uses a standard search provider and you want to include only some, but not all, of the available task types, leave this option unselected and restrict the search to a specific set of task types by using the procedure described in [“Searching Indexed Data in a Custom Task Inbox” on page 165](#).

- **Hide tasks accepted by others in the task type inbox results** (disabled by default). When this option is enabled, tasks accepted by other users do not appear in the task type inbox results (does not apply to My Inbox). Applies to custom task type inboxes only. Set this option by updating the default view of the Task Inbox Results portlet.
- For the Default view of the Task Details portlet:
 - **Auto-accept task upon modification** (disabled by default). When this option is enabled, the task is automatically accepted by the user when the user modifies the task, and released

immediately after the modification. You can enable this option when you create a task, and you can modify this option by updating the default view of the Task Details portlet.

- **Require task acceptance to permit modification** (enabled by default). When this option is enabled, the user is forced to accept a task before applying any modifications to the task. You can modify this option by updating the default view of the Task Details portlet.
- **Enable User Routing** (disabled by default). When this option is enabled, the user can assign an open task instance in the user's inbox to another user, group, or role. This ability appears on the task Data View tab. You can enable this option when you create a task, and you can modify this option by updating the default view of the Task Details portlet.

Modifying Client Options for a Task

➤ To modify the task view client options

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. On the **Overview** tab, expand the User Interfaces area if it is not already visible.
3. Refer to the descriptions above and click the default view you want to work with (applies only to the Task Details and Task Inbox Search Results portlet default views).
4. Click the **Update** button in the User Interfaces area. The Task UI Update wizard appears with the current properties for the selected element displayed.
5. Click the **Modify task client options for selected task UI page** option.
6. Click **Next**. The Task Client Options page appears.
7. Select or clear the check boxes for the available task client options as required. For more information about the available options, see [“Task Client Option Definitions” on page 170](#).
8. If you want to open the view in the view editor after applying the business data changes, select the **Open updated page in Editor** check box.
9. Click **Finish**. The changes to the task are saved automatically.

12 Working with Tasks in the Bindings View

■ About Tasks in the Bindings View	174
■ About Available Task Binding Properties	174
■ About Available Collaboration Task Bindings	175
■ About Task Indexed Search Content Provider Binding Properties	175
■ Modifying Task Binding Properties	176

About Tasks in the Bindings View

When you open a portlet view or a page view in the editor, you can examine a wide range of task properties and associated data with the Bindings view. The Bindings view is populated automatically when you open a view or page in the editor. When you click an item in the Bindings view hierarchy, the properties for that item are displayed in the Properties view.

You can modify task-specific property elements, as described in [“Modifying Task Binding Properties” on page 176](#). For more information about the available binding properties for a task, see [“About Available Task Binding Properties” on page 174](#).

For more information about working with the Bindings view, see *webMethods CAF and OpenUI Development Help*.

Note:

You can also use the Bindings view to extend task functionality with custom Java code. For more information, see [“Extending a Task with Custom Java Code” on page 116](#).

About Available Task Binding Properties

When you open a portlet view or page view in the view editor, you can click an item in the Binding view hierarchy to view and modify the properties for that item in the Properties view. All task business data is available in the Bindings view, as well as key elements such as the Task Search Provider (for the Inbox Results portlet). You can also set a number of behavioral properties for the task, such as:

- **Auto Accept** (default = false). When the data binding value for this property is set to true, the task is automatically accepted by the user when the user modifies the task, and released immediately after the modification. This property is the same as the Task UI Update wizard option, "Auto-accept task upon modification." Applies to the default view of the Task Details portlet.
- **Need Accept to Update** (default = true). When the data binding for this property is set to true, the user must accept a task before any actions or modifications can be done. This property is the same as the Task UI Update wizard option, "Require task acceptance to permit modification." Applies to the default view of the Task Details portlet.
- **Ad-hoc Routing** (default = false). When the data binding for this property is set to true, the user can assign the currently open task in the user's inbox to another user, group, or role. This ability appears on the task Data View tab. This property is the same as the Task UI Update wizard option, "Enable user routing." Applies to the default view of the Task Details portlet.
- **Maximum Users can Accept** (default = 1). This value specifies the maximum number of My webMethods users that can accept a specific task instance. You might want to enable multiple, simultaneous acceptances for a task used for voting purposes, for example, or where multiple approvals are needed. When the task is accepted by the number of user specified here, no other users can accept the task. This option is available in the Binding view only. Applies to the default view of the Task Details portlet.

- **Do Not Show Accepted By Others** (default = false). When the data binding for this property is set to true, the inbox search results will not contain any tasks accepted by other users. This property is the same as the task wizard option, "Hide tasks accepted by others in the task type inbox results." Applies to the default view of the Task Inbox Search Results portlet.

For the procedure to modify these binding properties, see [“Modifying Task Binding Properties” on page 176](#).

You can also modify several of the above properties in the task editor as described in [“Modifying Client Options for a Task” on page 171](#).

About Available Collaboration Task Bindings

When you open a collaboration task type in the task editor, you have access to the following bindings that are specific to collaboration tasks:

- Parent Task ID
- Collaboration Process ID
- Collaboration Step ID

To view these bindings, expand the **Task Info** entry in the Bindings view.

About Task Indexed Search Content Provider Binding Properties

When you create a task inbox results portlet, a task search content provider is created for the portlet. You have your choice of using a standard (full) search content provider, or an indexed search content provider. Regardless of which one is in use, the listing in the Bindings view is **Task Search Provider**. This provider is located in the Bindings view tree at:

Managed Beans > [TaskName]InboxResults/default > Task Search Provider

The contents of the task search content provider appears as follows:

- If you have implemented a standard task search content provider, all of the task business data fields appear in the **Task Search Provider > Business Data** entry of the Bindings view.
- If you have implemented an indexed task search content provider, only task business data fields that are marked as indexed fields appear in the **Task Search Provider > task data (indexed content)** entry of the Bindings view. Note that if you have added task business data to the task, but you have not marked any of the business data fields as indexed, the **task data (indexed content)** entry will have no contents.


For more information about indexed business data, see [“About Indexed Business Data Fields” on page 89](#).

Modifying Task Binding Properties

The most commonly modified binding properties (shown as examples below) are described in [“About Available Task Binding Properties” on page 174](#). Use the following procedure to modify these and other binding properties.

> To modify binding properties

1. In the Solutions view, navigate to and expand the task you want to work with, then double-click the portlet view to open it in the portlet view editor. All of the data available within the view is displayed in the Bindings view.

2. Click  in the Bindings view toolbar if you want to view all the available binding properties.

3. Locate the property element you want to work with; for example:

The following properties are found by opening the default view in the Task View portlet or the Details page and browsing to:

- For the View portlet: **Managed Beans > [TaskName]View/default > [TaskName]**

- For the Details Page: **Managed Beans > [TaskName]/default > [TaskName]**

- **adhocRouting**


- **Auto Accept**

- **Need Accept to Update**

- **Maximum Users can Accept**

The following property is found by opening the Task Inbox Results portlet default view and browsing to **Managed Beans > [TaskName]Results/default > Task Search Provider > Search Query**:

- **Do Not Show Accepted By Others**

4. Click the property element you want to work with to display its properties in the Properties view.
5. In the Properties view, click **Data Bindings**, and select the desired value from the data binding list or click  to select a binding expression.
6. Save your modifications.

A small orange icon appears on these properties  to indicate that the value for the given property is bound to a configuration or initialization value. For example, the Auto Accept property is bound

to a value of true or false. You can also place the cursor over the binding property to see a tool tip with the property description and binding value.

Note:

You can also modify several of the above bindings on the **Overview** tab of the task editor as described in [“Modifying Client Options for a Task”](#) on page 171.

13 Working with Task Pages

■ Task Pages Overview	180
■ Accessing a Task Page	180
■ Renaming a Task Root Page	181
■ Renaming a Task Page	181
■ Hiding and Displaying a Task Page	182

Task Pages Overview

Although you can modify and display user interfaces by working with task portlet views, you can also work with task pages displayed in the User Interfaces area of the task editor.

The Page entries contain a page for each major component of the user interface, with each page corresponding to a page presentation in the run-time environment. For example, suppose you create a task with the following portlets:

- Task View
- Task Details View
- Task Audit View
- Task Comments
- Task Collaboration

By default, the following pages are available:

- Task Application Root Page
- Inbox Page
- Start Page
- View Page

To access a task page, see [“Accessing a Task Page” on page 180](#). You can run any available page in a server, as described in [“Running a Task Page” on page 78](#).

You can also:

- Rename a page as described in [“Renaming a Task Page” on page 181](#).
- Hide or display a page as described in [“Hiding and Displaying a Task Page” on page 182](#).

Accessing a Task Page

➤ To access a task page

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. Click the **Overview** tab.
3. On the **Overview** tab, expand the User Interfaces area if it is not already visible.
4. Expand the Pages entry in the list to view the available pages.

Renaming a Task Root Page

You can modify the name of the task root page. This setting determines the text displayed in the Navigate panel in My webMethods, under **Applications > Monitoring > Business > Tasks**. To rename the inbox page, or any other page, see [“Renaming a Task Page” on page 181](#).

➤ To rename a task root page

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. Click the **Overview** tab.
3. On the **Overview** tab, expand the User Interfaces area if it is not already visible.
4. Expand the Pages entry in the list to view the available pages.
5. Select the Task Application Root Page entry and click **Edit**. The Page Properties dialog box appears.
6. Type a new root page name in the **Page Name** box.
7. Ensure that the **Show in MWS Taxonomy** check box is selected if you want the page to appear in the user's inbox or in the custom task inbox. For more information on this option, see [“Hiding and Displaying a Task Page” on page 182](#).
8. Click **OK**. The task is automatically saved to record the changes.

Renaming a Task Page

You can modify the name of the following task pages displayed in the User Interfaces area of the task editor:

- Task Details page
- Task Start page
- Task Inbox page

➤ To rename a task page

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. Click the **Overview** tab.
3. On the **Overview** tab, expand the User Interfaces area if it is not already visible.

4. Expand the Pages entry in the list to view the available pages.
5. Select the page you want to work with and click **Edit**. The Page Properties dialog box appears.
6. The text in the **Page Name** box is displayed at the top of the page when the page appears in the run-time environment. Type a new page name.
7. Ensure that the **Show in MWS Taxonomy** check box is selected if you want the page to appear in the user's inbox or in the custom task inbox. For more information on this option, see [“Hiding and Displaying a Task Page” on page 182](#).
8. Click **OK**. The task is automatically saved to record the changes.
9. After the modified task is published to My webMethods Server, a page with the previous name still exists. You must log on to My webMethods Server as sysadmin and manually delete the previous page, as described in the PDF publication *webMethods Task Engine User's Guide*.

Hiding and Displaying a Task Page

You can configure the following task pages to be hidden or displayed in My webMethods; these pages are available in the User Interfaces area of the task editor:

- Task Details page
- Task Start page
- Task Inbox page

➤ To hide or display a task page

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. Click the **Overview** tab.
3. On the **Overview** tab, expand the User Interfaces area if it is not already visible.
4. Expand the Pages entry in the list to view the available pages.
5. Select the page you want to work with and click **Edit**.
6. In the Page Properties dialog box, do one of the following:
 - Select the **Show in MWS Taxonomy** check box if you want the page to appear in the user's inbox or in the custom task inbox. In general, select this check box for the default Inbox page, but do not select it for the Details page, or the Start page (if one is present). If you have added custom pages to your task, select this check box as required.

- Clear the **Show in MWS Taxonomy** check box if you do not want the page to appear in the user's inbox or in the custom task inbox.
7. Click **OK**. The task is automatically saved to record the changes.

14 Working with Task Notifications

■ About Task Notifications	186
■ Task Notification Overview	186
■ Understanding Task Notification Behavior	187
■ Limitations to Subscription Access	188
■ About Task E-mail Notification Replies	189
■ About the Task E-mail Listener	190
■ Accessing the Task Notification Wizard	191
■ Creating a Task Notification	191
■ Adding a Task Action Link to a Notification	192
■ Extracting Data from a Task Notification Return E-mail	193
■ Modifying and Deleting a Task Notification	194
■ Sending Task Notifications to Non-Assigned Users	194

About Task Notifications

You can create a task e-mail notification to notify users of changes to or conditions applying to a task, for example:

- A task is assigned to a user
- A task is cancelled.
- A large loan request or merchandise order is received.
- A task has not been acted upon within a certain time period.

In addition, you can configure the e-mail notification to contain a Task Action Link; this control enables the notified My webMethods Server user to respond to the task without having to have a connection to My webMethods.

For general information about task notifications, see [“Task Notification Overview” on page 186](#). For specific information, see the following topics:

- [“Understanding Task Notification Behavior” on page 187](#)
- [“Limitations to Subscription Access” on page 188](#)
- [“About Task E-mail Notification Replies” on page 189](#)
- [“Accessing the Task Notification Wizard” on page 191](#)
- [“Creating a Task Notification” on page 191](#)
- [“Adding a Task Action Link to a Notification” on page 192](#)
- [“Extracting Data from a Task Notification Return E-mail” on page 193](#)
- [“Modifying and Deleting a Task Notification” on page 194](#)
- [“Sending Task Notifications to Non-Assigned Users” on page 194](#)
- [“About Task Notifications” on page 186](#)

Task Notification Overview

Software AG Designer enables you to create one or more *task notifications* in a task application. With proper permission, run-time users can then subscribe to available notifications for the tasks in their inbox, enabling them to receive an e-mail notification when a specified task-related event occurs. Because task events can be grouped into task control sets, you have great flexibility in designing your notification model. A My webMethods user with Task List Management access and proper permissions can also subscribe a user, group, or role to a task notification.

For additional information about notification subscriptions, see [“Limitations to Subscription Access” on page 188](#).

To be able to open a task from a task notification e-mail, the task notification recipient must have a network connection to the My webMethods Server instance that sent the notification.

For situations where a network connection is not available, you can configure the notification e-mail so that the My webMethods Server user receiving the notification can interact with a task running in the Task Engine without requiring a connection to My webMethods Server. For more information, see [“About Task E-mail Notification Replies” on page 189](#).

As an example, you might want to create a task notification for any of the following conditions:

- A task is assigned to a user.
- A task is cancelled.
- A large loan request or merchandise order is received.
- A task has not been acted upon within a certain time period.

Task notifications are triggered by the conditions you define in a task event; when the task event condition is matched, you configure the event action to publish the notification. Such events are referred to as *task notification events*.

When a task notification event is triggered, notification is provided by e-mail to all assigned users who are subscribed to the notification. The notification is sent to the user's e-mail account defined in the My webMethods user profile.

- For more information about when and to whom task notifications are sent, see [“Understanding Task Notification Behavior” on page 187](#).
- For more information about task notification events, see [“Creating a Task Notification Event” on page 145](#).

Understanding Task Notification Behavior

Task notifications are primarily intended to send e-mail messages to My webMethods users who are both assigned to a task and subscribed to a notification associated with that task. In this case, to receive and interact with task notifications, all of the following conditions must be true. The My webMethods user:

- Must be assigned to the task, either directly or as a member of a user, group, or role, or have the task delegated to them.
- Must be subscribed to a notification associated with the task, either as an individual user, or as a member of a group or role. If the user is to self-subscribe, the user must be granted the **Subscribe to Task** permission for the task application.
- Must have a valid e-mail address defined as part of their My webMethods user profile.

In addition, the user must be granted, at a minimum, permission to view the task, otherwise the user will not be able to access the task even though a notification is received. If the user is to interact with the task, the user must also be granted those task application permissions.

Note:

You can remove the “assigned to” requirement for users. In this case, all subscribed users receive a notification regardless if they are assigned to the task or not. For more information, see [“Sending Task Notifications to Non-Assigned Users” on page 194](#).

Notification e-mails are sent to all users who fulfill the notification conditions. For example, if three users are assigned to the task and are subscribed to a notification in the task, all three users will receive notification e-mails.

When a task is delegated to another user, the delegation is equivalent to assignment for the purpose of task notification. The delegated user must fulfill all the other conditions to receive notifications. That is, the delegated user must be subscribed to the notification, have a valid e-mail address, and must be granted the permissions to view or interact with the task.

Important:

A special condition applies to My webMethods administrators, or any user who is assigned the **Task Administration** permission.

Any user who has the **Task Administration** permission *and* is subscribed to a task notification will *always* receive task notifications from the task, whether the user is assigned to the task or not.

The reason for this behavior is that it is possible to create and publish a task type that contains no assignments, which means that there will be no task assignees when the task is queued (started). Any notification event that is triggered while the task is unassigned (including the Queued event) will never generate a notification according to the default notification conditions.

To enable task administrators to be aware of event occurrences within these unassigned tasks, a task administrator can subscribe to the task's notifications and receive notification e-mails without having to be assigned to the task.

Limitations to Subscription Access

When working with subscriptions in My webMethods, a user can be granted the **Subscribe to a Task** functional permission. This enables the user to manage task subscriptions on the Task Subscription page, which is an embedded component in My Inbox. The user accesses the Task Subscriptions page from either of two locations:

- A Subscriptions link located on the Inbox window of My Inbox.
- A Subscriptions link located in the Task Results window of the custom task inbox.

A user must have access permissions to My Inbox to manage subscriptions from either location. If a user is granted permission to the custom task inbox, but does not have access permission to My Inbox, when the user clicks the Subscriptions link in the custom task inbox, the error message "You are not authorized to view the resource" occurs.

Note:

This behavior also applies to the Scheduled Delegations link. For more information, see ["Limitations to Scheduled Delegation Access" on page 198](#).

There are several ways to address this situation:

- You can provide the user with My Inbox access permission. Doing so will enable access to all aspects of My Inbox, not just subscription management.

- You can set the `Rendered` property of the subscription portlet simple link control to `false` (or remove the control entirely) on the Default view of the task Inbox Results portlet. This will remove the Subscriptions link from the custom task inbox display. The user will have no access to subscription management.
- You can copy the existing Task Subscription portlet included in My Inbox and create and publish your own custom subscription page for the user to access. To access the default subscription page, log in to My webMethods as sysadmin and navigate to the page by either of these methods:
 - By using the page alias: `/webm.apps.workflow.inbox.subscription`
 - By clicking **Folders > My webMethods Applications > Fabric Tasks > Monitoring > Business > Tasks > My Inbox**.

You can then modify the custom task inbox Subscription link to point to the custom page you created and published.

About Task E-mail Notification Replies

When you create a task e-mail notification, it is possible to enable the My webMethods Server user receiving the notification to interact with the originating task (running in the Task Engine) without requiring a connection to My webMethods Server. This is accomplished as follows:

- A Task Action Link control enables you to include and define reply links in a task notification e-mail; each reply link specifies a reply e-mail address, subject text, body text, and, if desired, an action that is to be executed on the task after it is received.
- A Task Engine e-mail listener monitors a specified POP3 e-mail account for messages on a scheduled basis. Notification reply e-mails are downloaded, processed, and any embedded task action requests are executed. The task notification reply e-mail actions are logged and displayed on the task's Audit tab.

For example, suppose a business process has an activity step that determines if an order is approved or denied. The task developer can create a task notification based on an automated denial of the order (for example, the order amount exceeds the limit authorized for the customer).

This task notification is e-mailed to the account manager for that customer, who is subscribed to the notification. The task developer includes basic task business data in the task notification (for example, the order amount and reason for denial) and includes a Task Action Link control in the notification e-mail.

This control specifies an e-mail account for receiving a reply from the task notification subscriber, as well as an action that invokes a predefined task dataflow action. In this case, the link triggers a service that overrides the denial and approves the order. The link appears in the account manager's e-mail as "Override this denial and approve the order" The task developer can add additional links for other responses.

When the account manager receives the e-mail and decides to override the denial, he clicks the "Override this denial and approve the order" link. A reply e-mail is automatically created and sent (by way of the account manager's e-mail client) to the e-mail account specified in the link.

The e-mail listener monitoring this account downloads the account manager's reply, deletes the e-mail from the monitored account, and executes the action included by the task designer (in this case, the service that overrides the denial and approves the order). The task notification subscriber does not need to be connected to My webMethods Server to carry out this action.

This is a one-time occurrence; if the subscriber clicks the link again, the resulting e-mail will be downloaded by the e-mail listener, but will be ignored. Security measures are in place to prevent e-mail spoofing and the substitution of non-defined actions.

Note:

Task notifications are sent to the e-mail address recorded in the user's My webMethods Server profile. Some e-mail service providers may configure their server to remove, alter, or otherwise disable the URL contained in the "Click here to open task" link to comply with security requirements. In this case the link may be missing or inoperative in the delivered e-mail message.

For more information about the configuration and behavior of task notification e-mail replies, see:

- ["Adding a Task Action Link to a Notification" on page 192](#)
- ["About the Task E-mail Listener" on page 190](#)
- ["Extracting Data from a Task Notification Return E-mail" on page 193](#)

About the Task E-mail Listener

The ability to receive e-mail replies from a Task Action Link embedded in a task notification requires you to configure a task e-mail listener in My webMethods Server. To do so, you must:

- Identify or set up a POP3 e-mail account that will be used to receive replies from the task notification.
- Configure the Task Engine to monitor this e-mail account for reply e-mails.

For more information about the configuration and behavior of task notification e-mail replies, see the PDF publication *webMethods Task Engine User's Guide*.

You provide the e-mail server and user account information, and set a monitoring interval (for example, every 5 minutes). You can configure multiple e-mail listeners to monitor different mailboxes on different schedules. When the listener checks the specified account, it downloads any e-mails in the monitored account.

For each downloaded e-mail, the e-mail listener:

1. Deletes the e-mail from the server.
2. Parses the e-mail body looking for encoded action request. If a request is not found or is malformed, an error is logged in the Task Engine logs and the e-mail is ignored.
3. Determines the task instance for given taskID specified in the request.
4. Checks that the security GUID contained in the request matches the one stored on the task (the latter is generated during task notification publishing). If these values do not match, an error is recorded in the logs and in the task audit and the e-mail is ignored

5. Determines the binding expression of the action from the action ID from the request (this is again stored in the task instance during task notification publishing).
6. Executes the specified action for the given task and records an e-mail receive event in the task audit.

Accessing the Task Notification Wizard

You can access the task notification wizard in the following ways:

- On the UI Development perspective:
 - By clicking **File > New > Task Notification**.
 - By clicking the **New > Task Notification** arrow button in the main toolbar.
 - By clicking the New button on the main toolbar and clicking **Software AG > Composite Applications > Task Notification**.
- By clicking the **New > Other** command anywhere it is available, expanding the **Software AG/Composite Applications** folders and then clicking **Task Notification**.

By right-clicking a task's Task Notifications entry in the Solutions view and clicking **New Notification**.

Creating a Task Notification

➤ To create a task notification

1. Start the task notification wizard as described in [“Accessing the Task Notification Wizard” on page 191](#).
2. In the JSF Portlet View File page, specify the following information:
 - **Task**—Select the task you want the notification to apply to, or click **New** to create a new task.
 - **Subscription Name**—Type the name you want to apply to the notification, or accept the default name. This name appears in the subscription list seen by the My webMethods user.
 - **Notification Subject**—Type the text you want to appear in the subject line of the e-mail sent to the user; for example, "A new order approval task has been assigned to you."
3. Click **Next**.
4. On the second JSF Portlet View File page, accept the default value for the view file name, or type a new value.

From this point on, you can click **Next** to specify custom settings for the task notification, or click **Finish** to create the task notification with remaining default settings. The remaining pages in the wizard provide standard portlet configuration options.

When you click **Finish**, the task notification view opens in the view editor; you can modify the notification as required, and view the layout of the notification by clicking the **Preview** tab. For information about including a Task Action Link in a notification, see [“Adding a Task Action Link to a Notification” on page 192](#).

Important:

You must create a task notification event as described in [“Creating a Task Notification Event” on page 145](#), otherwise the notification will not be sent. In the run-time environment, you must also ensure that users are assigned to the task and subscribed to the notification. For information about subscribing to task notifications, see the PDF publication *webMethods Task Engine User’s Guide*.

Adding a Task Action Link to a Notification

You can configure the notification e-mail so that the My webMethods Server user receiving the notification can use it to interact with a task running in the Task Engine without requiring a connection to My webMethods Server. To do so, you must:

- Identify or set up a POP3 e-mail account that will be used to receive replies from the task notification.
- Configure the Task Engine to monitor this e-mail account for reply e-mails. For more information about the configuration and behavior of task notification reply e-mails, see the PDF publication *webMethods Task Engine User’s Guide*.
- Add a Task Action Link control to the notification interface and configure it to work with the correct e-mail account.

Note:

You can add multiple Task Action Links to your e-mail notification. Each link you include is constrained to a single action—although that action can be to execute a service that carries out a number of actions.

➤ **To add a Task Action Link to the notification interface**

1. In the Solutions view, expand **Tasks > [TaskApplicationProjectName] > TaskName] > Task Notifications**.
2. Open the task notification you want to work with in the editor.
3. In the editor, identify or create a container for the Task Action Link.
4. In the Palette view: **MWS Html > Control > Task > Task Action Link**.

5. Drag the Task Action Link control from the Palette view to the container in the editor.
6. Click the Task Action Link control in the editor and view its properties in the Properties view.
7. Define the following required properties:
 - On the **Display** tab, type a value for the **Label** field. This defines the link text displayed in the notification e-mail sent to the subscriber.
 - On the **Email** tab:
 - **Action**—a binding expression that specifies a data flow action that will be executed when the e-mail link is activated. For example, you can create a data flow action for the notification page bean that updates several task data fields.
 - **Email**—an e-mail address to which the e-mail response will be sent. This value can be an e-mail address or defined with a binding expression.
 - **Subject**—the text to be displayed in the Subject line of the return e-mail. This value can be a text string or defined with a binding expression.
 - **Body**—the text to be displayed in the body of the return e-mail. This value can be a text string or defined with a binding expression.

Note:

When you define the Action for the task action link, among the actions available to you is the ability to extract information from the return e-mail. For more information, see [“Extracting Data from a Task Notification Return E-mail” on page 193](#).

8. Save your modifications.

Extracting Data from a Task Notification Return E-mail

When you design a task notification, you can create a task notification action that sends an e-mail to a specified recipient. You can include a Task Action Link control in the e-mail message to enable the recipient to send an e-mail response back to the Task Engine. For more information, see:

- [“About Task E-mail Notification Replies” on page 189](#)
- [“Adding a Task Action Link to a Notification” on page 192](#)

As a task developer, you might want to extract certain information from the return e-mail message to use in further task processing, such as the sender, the date, attachments, body text, or anything else contained in the e-mail message.

For example, you might design the task notification e-mail so the recipient can approve or reject a purchase order using the task action link, and also enable the user to optionally enter a comment. When a comment is present, you want to extract the comment from the return e-mail message and include it in the task information, along with the user ID of the sender.

You can do all of this when you create the action that is executed when the return e-mail message is received. From the Java code of the action being executed, you can access the original Mime e-mail message object from the faces context under the "mimeType" bean name. From this object you can further access all e-mail properties and data:

```
javax.mail.internet.MimeMessage mimeType =  
resolveExpression("#{mimeType}");
```

Modifying and Deleting a Task Notification

➤ To modify a task notification

1. In the Solutions view, expand **[Solution] > [TaskApplicationProjectName] > Task Notifications**.
2. Locate the task notification you want to work with and do one of the following:
 - Double-click the task notification to open it in the view editor; click a control to examine and modify its properties in the Properties view.
 - To delete the task notification, right-click the notification and click **Delete**.
3. Save your modifications.
4. Publish the task application that contains the task to make the modified task application available to My webMethods users.
5. If you need to modify the event type that triggers the notification, see [“Configuring a Task Event Type” on page 138](#) for more information.

Sending Task Notifications to Non-Assigned Users

By default, the Task Engine sends e-mail notifications *only* to users who are both subscribed to the notification *and* assigned to the task. For example, if five people are subscribed to the task, but the task is assigned to only one user, only that assigned user will receive task notifications.

You can change this behavior on an individual task basis, so that the Task Engine sends e-mail notifications to *all* subscribed users, whether they are assigned to the task or not. As in the default notification behavior, you must assign permissions to all subscribers to view the task, and permission to interact with the task as needed.

To return to the default behavior, use the procedure below but set the value to `true`.

Important:

Use caution when changing this default setting. The subscribers are subscribing to notifications from a particular *task type*. This means that all subscribers will receive notifications from *all task instances started from that task type*. This could generate a substantial number of notification e-mails.

➤ **To send task notifications to unassigned and subscribed users**

1. You must first add the `TaskNotificationContentProvider` bean to the Faces Configuration Editor. In the Solutions view, expand **User Interfaces > TaskName**.
2. Double-click **Faces Configuration** to open the Faces Configuration Editor.
3. Click the **ManagedBean** tab, then click **Add**.
4. Make sure **Using an existing Java class** is selected and click **Browse**.
5. On the Select Type dialog box, type `TaskNotification` in the **Select Entries** field. The `TaskNotificationContentProvider` bean should appear in the Matching items window.
6. Click the `TaskNotificationContentProvider` bean to select it and then click **OK**.
7. Click **Next**. Provide a description if needed, and click **Finish**.
8. Next, you must modify the `setNotifyAssignedUsers` method. In the **Initialization** area of the editor, make sure **General class** is selected and then click **Add**.
9. In the Add Managed Property dialog box, select **notifyAssignedUsers** from the **Property name** drop-down list.
10. For the **Value** field, click **Edit**.
11. In the Edit Value dialog box, type `false` and click **OK**.
12. Save the task.

15 Working with Task Delegation

■ Task Delegation Overview	198
■ Limitations to Scheduled Delegation Access	198

Task Delegation Overview

The task delegation feature enables a user to delegate a task from an inbox (or from the Task List Management page) to another user, with a complete audit log of who the task was delegated from, who it was delegated to, and who did the delegating. The delegated task continues to be visible in the original inbox. Task delegations can be scheduled, to address situations such as employee vacation time.

Task delegation is an inherent feature of the Task Engine, and in general, a task developer does not have to take any special steps to enable the delegation feature.

You can read about the available task delegation controls that are included by default in these topics:

- [“About the Inbox Results Default View” on page 54.](#)
- [“About the Task Details View Portlet” on page 57.](#)
- [“About the Task Audit Portlet” on page 58.](#)

For more information about how task delegation works, see “Delegating Tasks” in the PDF publication *webMethods Task Engine User’s Guide*.

Some limitations apply to whether or not a particular user can access the Scheduled Delegations page. For more information, see [“Limitations to Scheduled Delegation Access” on page 198.](#)

Limitations to Scheduled Delegation Access

When working with task delegations in My webMethods, a user manages scheduled task delegations on the Scheduled Delegations page, which is an embedded component in My Inbox. The user accesses the Scheduled Delegations page from either of two locations:

- A Scheduled Delegations link located on the Inbox window of My Inbox.
- A Scheduled Delegations link located in the Task Results window of the custom task inbox.

A user must have access permissions to My Inbox to manage subscriptions from either location. If a user is granted permission to the custom task inbox, but does not have access permission to My Inbox, when the user clicks the Scheduled Delegations link in the custom task inbox, the error message “You are not authorized to view the resource” occurs.

Note:

This behavior also applies to the Subscriptions link. For more information, see [“Limitations to Subscription Access” on page 188.](#)

There are several ways to address this situation:

- You can provide the user with My Inbox access permission. Doing so will enable access to all aspects of My Inbox, not just scheduled delegation management.
- You can set the Rendered property of the scheduled delegation portlet simple link control to false (or remove the control entirely) on the Default view of the task Inbox Results portlet. This

will remove the Scheduled Delegations link from the custom task inbox display. The user will have no access to scheduled delegation management.

- You can copy the existing Scheduled Delegations portlet included in My Inbox and create and publish your own custom subscription page for the user to access. To access the default subscription page, log in to My webMethods as sysadmin and navigate to the page by either of these methods:
 - By using the page alias: /webm.apps.workflow.inbox.sched.delegation
 - By clicking **Folders > My webMethods Applications > Fabric Tasks > Monitoring > Business > Tasks > My Inbox**.

You can then modify the custom task inbox Scheduled Delegations link to point to the custom page you created and published.

16 Publishing Tasks

■ About Publishing Tasks	202
■ Publishing a Task	202

About Publishing Tasks

After you create a task, you must publish the application that contains it to the target run time so that it can run on the Task Engine.

For more information, see [“Publishing a Task” on page 202](#).

Publishing a Task

After you create a task in Software AG Designer, it remains available within the design-time environment of Designer but is not present in the run-time environment. To make the task available for a Task Engine instance, running on My webMethods Server or Integration Server, you must publish the task application project containing the task to the required run-time environment. When a task application is initially published, all tasks in the task application project are published to the run-time environment.

A project containing one or more tasks is published to the target runtime like any other portlet project. For more information about publishing portlet projects in general, see *webMethods CAF and OpenUI Development Help*.

Considerations When Publishing Task Types with Duplicate Names

In some instances, it may be necessary to create two or more task types with the same task type name. Programmatically, this is not a problem because the Task Engine tracks each task type by its task type ID, not its task name.

However, the existence of duplicate task type names in the My webMethods user interface will cause problems when you publish task types with duplicate names, unless you take the task editing precautions described below. Specifically, the most recently published task type will overwrite any existing task type with the same task type name.

Precautions


Whenever you create a task type with a duplicate name, before publishing the task type, you must use the task editor to manually rename the task application root page to make it unique among all tasks deployed to My webMethods Server; otherwise, the most recently deployed duplicate task type will overwrite the existing task application pages. This will also ensure that custom inbox page names will be unique. For information about modifying the task application root page, see [“Renaming a Task Root Page” on page 181](#).

Automatically Publishing Tasks in a Process

When you build a process containing one or more tasks, the task application project(s) containing those tasks can be automatically published depending on the setting for the **Automatically Deploy Task to Task Engine** option in the Process Designer preferences. If this preference is not enabled,

you must publish each task application project manually to make the tasks in it available to the process.

Note:

To view the preference used in the following procedure, you must first enable the advanced properties by clicking  in the main toolbar.

➤ **To enable automatic deployment of tasks**

1. In the main menu, click **Window > Preferences**.
2. In the preference list, expand **Software AG** and click **Process Development > Build and Upload**.
3. On the Build and Upload page, set the **Automatically Deploy Task to Task Engine** option as desired (**Always**, **Never**, or **Prompt** (default)).

When you click **Always**, all tasks contained in the process will be published to My webMethods Server each time you build and upload the process (note that in actuality, the task application project(s) containing the tasks are published). For more information about the task publishing process, see [“About Optimized Task Publishing” on page 203](#).

About Optimized Task Publishing

When a task application project is first published to the runtime environment, the publishing process can take a considerable amount of time, especially if the task application is a complex one with many tasks, custom interfaces, assignments, and events. Network capacity and traffic can also affect the publishing time.

To ensure that the subsequent publishing of task application projects is as rapid as possible, Software AG Designer optimizes this process by publishing only those portions of the task application project that have changed since the last publication. In most cases this optimization is completely transparent.

Note:

Optimized publishing applies only when publishing to My webMethods Server.

However, it is possible to modify portions of the task application project in the My webMethods Server run-time environment (although this is generally not recommended). If these changes are not applied to the task application project in Designer as well, the two versions will no longer be synchronized.

For instance, if the task application taxonomy is changed in My webMethods Server (for example, the inbox page is renamed), publication of the task application will not update the renamed page if that portion of the task application was not updated in Designer. You can force a publication of the entire task application project; for more information, see [“Forcing the Full Publication of a Task Application Project” on page 204](#).

Forcing the Full Publication of a Task Application Project

Software AG Designer optimizes the task application publication process by publishing only those portions of the project that have changed since the last publication.

To force the full publication of an entire task application, install the task application from My webMethods Server using the Install Administration functionality (for more information, see *Administering My webMethods Server*).

17 Deleting Tasks

■ Deleting a Task	206
■ Deleting a Task in My webMethods	206
■ Removing Tasks from Integration Server	206

Deleting a Task

Tasks can be deleted in Software AG Designer; deleted tasks cannot be recovered.

Note:

If the task you want to delete is the only task in the project, the easiest method is to delete the entire project from the Solutions or Navigator view. However, if you wish to retain the project, use the following procedure.

➤ **To delete a task**

1. Locate the task you want to delete with in the Solutions view.
2. Right-click the task and click **Delete**.
3. You may also want to delete the task in My webMethods if the task has been published to My webMethods Server. For more information, see the Task Engine documentation.

Deleting a Task in My webMethods

If you delete a task in Software AG Designer, it is possible that the task still exists on your instance of My webMethods Server if you have previously published the task to that server. Use the following procedure to remove the task from the server environment.

➤ **To delete a task from My webMethods Server**

1. Log in to My webMethods Server with administrator privileges.
2. Navigate to **Administration > Business > Tasks > Task Administration**.
3. Select the task you want to delete and click **Delete**.

Removing Tasks from Integration Server

If you have deleted a task application project in Designer and that task application has previously been published to Integration Server, you may also want to remove the tasks, contained in the application project from the Integration Server runtime. Removing a task application from an Integration Server runtime removes all tasks, that were created and published together as a single application project. You cannot select a single task type to remove from the runtime.

➤ **To delete tasks from Integration Server**

1. In the **Servers** view in Software AG Designer, right-click the Integration Server from which you want to remove task applications, and click **Add and Remove**.

2. In the **Add and Remove** dialog, select the task application that you want to delete and click **Remove** to move it to the **Available** list.
3. Click **Finish** to delete the task application and its task types from Integration Server.

18 Working with Task Attachments

■ Task Attachment Overview	210
■ Adding Attachment Capability to a Task	210
■ Sharing Attachments by Setting the Attachments Scope	211

Task Attachment Overview

Attachments can be added to a task in the following ways:

- By using the capabilities provided by the Comments portlet (see [“About the Task Comments Portlet” on page 59](#)).
- By adding attachment capability to the Task Details portlet default view (see [“Adding Attachment Capability to a Task” on page 210](#)).

The ability to add and delete attachments in a task is included by default with the Comments portlet. If you created a task with either a default or customizable Comments portlet, attachment handling capability is included in the **Comments** tab (for more information, see [“About the Task Comments Portlet” on page 59](#)).

When you add attachment capability to the Task Details portlet default view as described in [“Adding Attachment Capability to a Task” on page 210](#), you use the Attachments List Panel control.

For more information about JCR attachment providers, see "JCR Providers" in the *webMethods CAF and OpenUI Development Help*.

When attachments are available in a task, you can specify if those attachments are to be shared by setting a task attachments scope (see [“Sharing Attachments by Setting the Attachments Scope” on page 211](#)).

Adding Attachment Capability to a Task

You can add attachment and document handling capabilities to the tasks you create. You can provide users the ability to attach, view, update, and remove documents within a task.

Note:

A new attachment panel is added to the selected portlet view or Web application page each time you complete the following procedure.

➤ **To add attachment capability**

1. In the Solutions view, double-click the task to open it in the task editor.
2. Click the **Overview** tab and expand the User Interfaces area if it is not already visible.
3. Click the default view in the Task Details portlet, or the default view in the Queue New Task Portlet.

Note:

Although the options available on the Task Update wizard enable you to add attachment support to the Start page or portlet, this capability is provided primarily as a testing mechanism. By adding attachment support to the Start page or portlet, you can queue a standalone task in the runtime and specify an attachment at start time, enabling you test

attachment behavior and network connections. Unless you want to use this test capability, you would normally not add attachment support to a Start page or portlet.

4. Click the **Update** button in the User Interfaces area. The Task UI Update wizard appears with the current properties for the selected element displayed.
5. Click the **Add attachments support to selected task UI page** option.
6. Click **Next**. The Add Attachment Support panel appears.
7. If you want to open the view in the view editor after applying the business data changes, select the **Open updated page in Editor when finished** check box.
8. Click **Finish**. An attachments panel is added to the selected view.
9. The changes to the task are saved automatically.

Sharing Attachments by Setting the Attachments Scope

You can specify how task attachments are to be shared by setting the attachments scope within a task.

Note:

The sharing of attachments applies only to attachments on the **Comments** tab or to attachments added from an attachments list panel that has been added to Task Details portlet default view (see [“Adding Attachment Capability to a Task” on page 210](#)). Attachments on the **Content** tab are not subject to sharing.

» To set the attachments scope for a task

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. Click the **Overview** tab.
3. In the Task Attachments Scope list, select one of the following:
 - **Collaboration Process Instance.** Allows the task's attachments to be shared with the parent task and all other collaboration tasks running in a collaboration process instance.
 - **Process Instance and Task Control Set.** Allows the task's attachments to be shared with the parent task and all other collaboration tasks running in a collaboration process instance, as well as with all tasks that share the control set used by the task.
 - **Task Instance.** Restricts the task's attachments to the task only; attachments are not shared.
 - **Process Instance.** Allows the task's attachments to be shared with all other tasks running in the same business process instance.

4. Save your changes.

19 Working with Task Comments

■ Task Comments Overview	214
■ Setting a Scope for Task Comments	214

Task Comments Overview

The ability to add, modify, and delete comments in a task is included by default with the comments portlet. If you created a task with either a default or customizable comments portlet, commenting capability is included in the Comments tab (for more information, see [“About the Task Comments Portlet” on page 59](#)).

A task comments list provider is available in the Palette view: **Data > Task > Task Comments List Provider**.

When comments are available in a task, you can specify if those comments are to be shared by setting a task comments scope.

Setting a Scope for Task Comments

You can determine how task comments are shared and where they may be viewed by setting the task comment scope.

➤ To set the comments scope for a task

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. Click the **Overview** tab.
3. In the Task Comments Scope list, select one of the following:
 - **Collaboration Process Instance**—allows the task's comments to be shared with the parent task and all other collaboration tasks running in a collaboration process instance.
 - **Process Instance and Task Control Set**—allows the task's comments to be shared with the parent task and all other collaboration tasks running in a collaboration process instance, as well as with all tasks that share the control set used by the task.
 - **Task Instance**—restricts the task's comments to the task only; comments are not shared.
 - **Process Instance**—allows the task's comments to be shared with all other tasks running in the same business process instance.
4. Save your changes.

20 Working with Custom Task Inboxes

■ Custom Task Inbox Overview	216
■ About Creating a Custom Task Inbox	216
■ Modifying the Results List for a Custom Task Inbox	217
■ Example Modifying the Default Search Fields and Results	220
■ Example Adding an Indexed Business Data Search Field	221

Custom Task Inbox Overview

When a My webMethods user logs on, all of the tasks assigned to that user (either directly or from membership in a group or role) appear in the user's My Inbox panel. Aside from the standard page editing capability provided with the Options button, the My Inbox panel cannot be modified or customized.

However, with proper privileges, the user also has access to pages that display all of the tasks of a particular type assigned to that user. For example, if the user is assigned tasks of type "Order Approval," the user can also navigate to a page with a task type inbox that displays only Order Approval tasks. You can provide customized versions of these task-specific inboxes (also referred to as a *custom task inbox*).

When you create a task in Software AG Designer, you can choose to include several default user interface portlet views, as described in [“About Default Task Portlets” on page 50](#). You can customize these default views (or create entirely different views) to provide run-time users with specialized search and display capabilities based on the business data contained in the task.

For example, you may want to enable a run-time user to search through Order Approval tasks by order amount, customer name, or account number. If you include this business data as part of your task inputs, you can configure the search bar portlet view and the search results portlet view to provide access to these business data elements in the run-time inbox for that task type.

By default, a search executed in a custom task inbox applies only to the task type associated with the inbox. However, you can extend the search capabilities to include other task types:

- For task types using a standard search provider, you can select the task client option **Search all task types when searching for task type inbox results**, as described in [“Modifying Client Options for a Task” on page 171](#) and [“Task Client Option Definitions” on page 170](#). This applies the search to all task types.
- For task types using an indexed search provider, you can specify the task types to be included in the search, as described in [“Searching Indexed Data in a Custom Task Inbox” on page 165](#). This method can also be applied to task types with a standard search provider if you want to limit the number of task types in the search.

About Creating a Custom Task Inbox

When you create a task in Software AG Designer, you can choose to include several default user interface portlet views. In addition, you can further expand your interface modification options by choosing to expose additional portlets by selecting the **Customizable** check box when you create the task type in the Task wizard. For more information, see [“Creating a New Task with the Task Wizard” on page 40](#) and [“About Customizable Task Portlets” on page 57](#).

You can customize these views by removing existing interface components or by adding specific business data from within the task. The default portlet views include:

- **Task View Portlet**—This view displays the task details when the user opens the task (for example, from the My Inbox page or the Task List Management page). For example, you can

customize this view to include any of the business data elements within the task, or the ability to attach and view documents, or to type comments about the task.

- **Task Inbox Bar Portlet**—This view defines the search parameters available to the user. You can customize this view to add the ability to search tasks in this inbox using any of the business data elements contained within the task.
- **Task Inbox Results Portlet**—This view displays the results of the task inbox search action. For example, you can customize this view to add columns that display specific business data from within the task.

Note:

If you plan to use the Task Inbox Results Portlet to implement searches of task types that contain indexed business data fields, you must manually modify the .java file for the Task Inbox Results default view. For more information, see [“Searching Indexed Data in a Custom Task Inbox” on page 165](#).

- **Task Start Portlet**—This view is seen by run-time users who have task administration privileges to start a new task. You can customize this view to include any of the business data elements within the task.

You also have the option of creating a completely new portlet, as described in [“Accessing the New Portlet Wizard” on page 66](#), and adding to it a new portlet view, as described in [“Accessing the New Portlet View Wizard” on page 72](#). If you have already created the portlet but want to replace, rather than modify, the existing portlet view, you can do that as well. In either case, you can populate the new view with whatever controls and task business data you want to make available to the run-time user.

Modifying the Results List for a Custom Task Inbox

You modify the appearance and behavior of the results list in a custom task inbox by setting the following portlet preferences for the task Inbox Results portlet:

- [“Configuring the Results List Column Display in a Custom Inbox” on page 217](#)
- [“Configuring the Results List Page Size in a Custom Inbox” on page 218](#)
- [“About the Column Width Preference for the Results List” on page 219](#)
- [“Configuring the Results List Sorting Behavior in a Custom Inbox” on page 219](#)

Configuring the Results List Column Display in a Custom Inbox

When you create a new task type, all available columns are selected for display. You can modify the columns to be displayed by default when a user opens a custom task inbox. Each user has the ability to modify the columns displayed with the **Tools > Properties** menu after accessing the inbox. These values are persisted on a per-user basis by the default Scope setting for the preference, Value Stored Per User.

➤ **To modify the default column display in the task Inbox Results list**

1. In the Solutions view, locate and expand the task you want to work with.
2. Double-click any portlet to open the portlet editor.
3. Click the **Configuration** tab if it is not already selected.
4. In the **Portlet Application Configuration** list, locate the [*taskName*]InboxResults portlet and expand the **Preferences** node within it.
5. Select the columnDisplay preference.
6. Initially, the **Value** list of the **Portlet Preference** area is empty. This signifies that all available columns are displayed when the inbox is opened. Do one of the following:
 - If the **Value** list is empty:
 - Click **Add** to specify an individual column for display. When you add one or more values to the list, only the specified columns are displayed to the user.
 - Click the **Enter Value Here** entry in the list and type a column name. You can obtain the column name by opening the task in its custom inbox in My webMethods and viewing the **ID** fields on the **General** tab in the **Tools > Properties** view. For example, `taskPriorityColumn`.
 - If the **Value** list is populated, click **Add** to add another column, or select an existing entry and click **Delete** to remove a column. If all values are removed (that is, the the **Value** list is empty), all available columns are displayed.
7. Save your changes.

Configuring the Results List Page Size in a Custom Inbox

When you create a new task type, the page size (that is, the number of rows displayed in the results list) is set to 20 by default. You can modify the number of rows to be displayed by default when a user opens a custom task inbox. Each user has the ability to modify the number of rows displayed with the **Tools > Properties** menu after accessing the inbox. This value is persisted on a per-user basis by the default Scope setting for the preference, Value Stored Per User.

➤ To configure the number of rows displayed in the task Inbox Results list

1. In the Solutions view, locate and expand the task you want to work with.
2. Double-click any portlet to open the portlet editor.
3. Click the **Configuration** tab if it is not already selected.

4. In the **Portlet Application Configuration** list, locate the [*taskName*]InboxResults portlet and expand the **Preferences** node within it.
5. Select the initialPageSize preference. By default, value in the **Value** list of the **Portlet Preference** area is set to 20.
6. Click the existing value in the **Value** list and type a new value. The value you specify must be a positive integer.

Note:

In practice, there are no upper limits to the value you specify here. However, be aware that specifying a very high row count here can, depending on system resources, cause slow response times or out-of-memory errors each time the page is accessed or refreshed.

7. Save your changes.

About the Column Width Preference for the Results List

The portlet preferences for the task Inbox Results portlet include the columnWidths preference. While this preference is not configurable, it is advantageous to understand its purpose.

When you create a new task type with the Task wizard, the columns in the results list for the Inbox Results portlet are of fixed width. Using standard CAF development practices, you can add the Drag to Resize Columns control to the results list table to enable the user to resize columns.

If this control is not present, the columnWidths preference has no function. When the control is added, the columnWidths preference is used to persist the column width settings applied by the user. Do not apply any values to this preference.

Configuring the Results List Sorting Behavior in a Custom Inbox

Generally, to provide sorting for a table you must set the "Ordinal" sorting property of a column to non-zero value (default is zero, meaning it does not sort on this column). However, a different mechanism is used for the task Inbox Results portlet. The Edit view of the task InboxResults portlet contains controls that define, among other things, the sort order of the results, and whether they are sorted in ascending or descending order. By default, these are set to:

- Sort by = Task ID
- Sort order = ascending

A task user can change these settings in My webMethods using the controls described in [“About the Inbox Results Edit View” on page 55](#). These values are persisted on a per-user basis by the default Scope setting for the preference, Value Stored Per User.

Note:

Changing the sort properties for the Inbox Results portlet applies to the custom inbox and does not affect sorting in Task List Management and My Inbox (these settings can be modified by the user during runtime).

You can change the default settings for the SortBy and SortOrder properties as follows:

➤ **To change the default settings for sort by and sort order in the task Inbox Results list**

1. In the Solutions view, locate and expand the task you want to work with.
2. Double-click any portlet to open the portlet editor.
3. Click the **Configuration** tab if it is not already selected.
4. In the **Portlet Application Configuration** list, locate the [*taskName*]InboxResults portlet and expand the **Preferences** node within it.
5. To change the default Sort By column, select the initialSortBy preference; to change the default Sort Order, select the initialSortAscending preference.
6. In the **Value** list of the **Portlet Preference** area, select the existing value or values you want to replace, and click **Delete**.
7. Click **Add**.
8. Click the **Enter Value Here** entry in the list and type one of the following:
 - To set the initialSortBy preference, type the name of the column you want to use to define the sort order. Obtain the name from the **ID** field on the **General** tab in the Properties view for the table column control. For example, taskPriorityColumn. Although it is possible to enter multiple values here, only the initial value is evaluated.
 - To set the initialSortAscending preference, type false to sort in descending order, or true to sort in ascending order.
9. Save your changes.

Example Modifying the Default Search Fields and Results

This example applies to task types that use a standard search provider. If you want to search for indexed business data fields, see [“Example Adding an Indexed Business Data Search Field” on page 221](#).

For this example, assume that you are working with an order approval task, and the business data for that task includes an input field that specifies the order amount. The value of this content provider element is derived from an Integration Server document that serves as the task's input.

In this example, you want to enable the user to search for tasks by order amount, and also display an order amount column in the task inbox results.

» To modify the default search field and results

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. On the **Overview** tab, expand the User Interfaces area if it is not already visible.
3. Click the default view of the Task Inbox Search Bar portlet.
4. Add a custom business field (in this case, the order amount field) to the portlet view as described in [“Creating Custom Search Fields from Task Business Data” on page 104](#).
5. When you specify the custom business data field you want to work with, you can also specify that you want a column to be added to the search results table by selecting the **Add Column displaying this field to Inbox Search Results table** check box.
6. After completing the procedure described in [“Creating Custom Search Fields from Task Business Data” on page 104](#), click the Inbox page in the User Interfaces area of the task editor **Overview** tab.
7. Click **Run** to view and interact with the Inbox page as it will appear in the run-time environment. Note that the new search field is present, and a corresponding column is present in the search results.

Example Adding an Indexed Business Data Search Field

Task types can be created with indexed business data fields (for more information, see [“About Indexed Searches” on page 159](#)). When you create a custom task inbox, you can add an indexed business data search field to the task search interface.

Note:

Only one indexed business data field can be added.

In addition, you can configure the task type to search for this indexed business data field across two or more task types (that is, beyond the single task type associated with the custom task inbox).

» To add an indexed business data search field to the search interface

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. On the **Overview** tab, expand the User Interfaces area if it is not already visible.
3. Click the default view of the Task Inbox Search Bar portlet.

4. Add a custom indexed business data field to the portlet view as described in [“Creating Custom Search Fields from Task Business Data”](#) on page 104.
5. After completing the procedure described in [“Creating Custom Search Fields from Task Business Data”](#) on page 104, click the Inbox page in the User Interfaces area of the task editor **Overview** tab.
6. Click **Run** to view and interact with the Inbox page as it will appear in the run-time environment. Note that the new search field is present, and a corresponding column is present in the search results.

To configure the task type to search for this indexed business data field across two or more task types (that is, beyond the single task type associated with the custom task inbox), see [“Searching Indexed Data in a Custom Task Inbox”](#) on page 165.

21 Working with Task Workflows

■ Task Workflow Overview	224
■ Code Samples	225
■ Implementation Components	226
■ Working with the Sample Applications	226
■ Key Implementation Points	228

Task Workflow Overview

Tasks are closely integrated with webMethods business process models. A task can be added to a business process as a *user task activity*, also referred to as a *user task*. Each user task can represent a discrete human activity within the process, such as installing a piece of equipment, but in more advanced processes, you might want to create a series of user tasks, where each user task step represents one activity in a larger, overall procedure.

When you connect a series of tasks together in this way, you are creating a *task workflow*, also referred to as a *form flow*.

A task workflow enables you to break up a complicated procedure into a series of simpler user task steps. Task orchestration is done by the process model and not inside a single task that could become overly complex and difficult to deploy, maintain, use, and understand.

In a task workflow, you can use the more sophisticated logic of the process model to determine which task to start and the next task interface to display to the user. The process model can call additional services to implement any necessary business logic between tasks steps. As the task workflow progresses, the individual tasks open automatically; the user does not have to manually open each individual task.

In a typical task flow scenario, a previous step in a business process completes and then transitions to and starts (or *queues*) a user task as the next step in the process.

For example:

- Task1 presents an interface that enables the user to gather user personal information from an applicant to run a credit check.
- When the user completes this task, the task submits all the collected data back to the process.
- The process executes some business logic (most likely in the form of one or more services) to determine the applicant's credit score based on the submitted data.
- Depending on the result of the credit check, the process transitions to either a loan application task or to a "credit check failed" report task. The process presents the interface of the next task to the user automatically.

Basic Operation

From the viewpoint of the user, the task workflow appears as a seamless flow of task interfaces in My webMethods. As the user completes each task in the workflow, the next logical interface appears, thus eliminating the need for the user to locate each new task in My Inbox and then open it manually.

Operationally, this behavior is implemented with the following:

- A process component that waits for the next task of the business process (or for completion of the workflow). Upon notification, the waiting component then executes a user interface action, such as open the next task in the workflow, or transition to the next post-workflow activity in the process. The waiting component uses a Java API to wait for the next action to occur.

- A process step that notifies the waiting component. Within the workflow, the next workflow task uses a Java API to notify the waiting step. When the final step in the workflow completes, it waits for notification from a following step that invokes an Integration Server flow service to provide the notification. In both cases, the waiting step can then transition to the next activity.

Code Samples

To assist you with understanding how to implement a task workflow, you can examine and deploy a sample task application, process model, and Integration Server package that support a very simple loan application process. You can find the code samples on the TECHcommunity website at:

<http://techcommunity.softwareag.com/ecosystem/communities/public/webmethods/products/caf/codesamples>

The sample zip files contain the following items:

- **FormFlowProcessTasks.zip.** This file contains the FormFlowProcess Tasks task application that contains two tasks, each with the following portlets:
 - **StartLoanProcess.** The primary purpose of this portlet is to provide a My webMethods interface that enables the user to enter a loan process number and start a new loan process instance. The portlet transitions to the first task interface when the first task is instantiated by the process instance.
 - **TaskStep1Overview.** This is a standard Task Overview portlet that serves as a container for providing a formatted message panel for displaying any JSF context messages, as well as portlet include controls for any additional portlets in the task.
 - **TaskStep1Start.** A standard Start portlet that enables a user to start a new task on the Task Engine Administration page of My webMethods. If a task is used exclusively within a process (where it will be triggered only by the process), you do not need to include a task start portlet.
 - **TaskStep1View.** Also a standard portlet, this is identified as the Task Details portlet in the task editor. It presents editable and non-editable task information, as well as various controls and tabs. This is the portlet where you implement transition to the next task in the task workflow when the task is completed.
- **WmFormFlowTest.zip.** This file contains the Integration Server package with the required IS document “Loan” as well as the following:
 - startLoanProcess flow service to start loan application process.
 - formFlowComplete flow service to complete a task workflow when process ends.
- **FormFlowTest.zip.** This file contains the FormFlowProcess process model, a very simple process model that demonstrates a task workflow:



The start message step is subscribed to the WmFormFlowTest IS document named Loan. Task Step 1 and Task Step 2 are examples of a two-stage task workflow, and the process completes with a service activity that calls a notification service.

Implementation Components

You implement a task workflow using these installed software components:

- A Java API `com.webmethods.portal.service.task.ITaskFormFlowService` included with Task Engine. To read the Java docs for the Java interface, see the *webMethods CAF and My webMethods Server Java API Reference*.
- A built-in Integration Server public service included with the WmTaskClient package `pub.task.taskclient:formFlowTaskNotify`. For more information about this and other task services, see the PDF publication *webMethods Task Engine API and Service Reference*. For general information about Task Engine services, see [“Working with the Task Engine APIs” on page 259](#).

Both of these publications are available in the webMethods Product Suite section of the [Software AG Documentation Web site](#).

Working with the Sample Applications

Before you can work with the sample applications you must have My webMethods Server, Integration Server, and Software AG Designer running, and you must:

- Import the process model into Software AG Designer and then build and upload the process to the Integration Server.
- Import the task application into Designer and then publish it to My webMethods Server.
- Unzip the WmFormFlowTest package, add it to the `/packages` directory of your Integration Server instance, and then activate the package with the Integration Server Administrator.
- Log in to My webMethods Server and access the StartLoanProcess portlet with the following URL: `[hostname]:[port]/start.loan.process`. For example: `localhost:8585/start.loan.process`.

At this point, you can demonstrate a very simple usage scenario with the installed applications:

1. Type a value into the **Loan Number** field in the Start Loan Process 2 window.
2. Click **Start Single Process**.
 - An instance of the FormFlowProcess business process starts, and immediately starts Task Step 1, while the StartLoanProcess portlet waits.

- When the StartLoanProcess portlet is notified that the user task is fully instantiated, the StartLoanProcess portlet redirects to the user task interface URL, which opens the task interface for the end user to interact with.
- In this sample, Task Step 1 has no functionality other than to display the Loan Number entered in the StartLoanProcess portlet.

3. Click **Complete**.

- The process instance starts the next user task in the sequence, Task Step 2, while Task Step 1 waits.
- When Task Step 1 is notified that the Task Step 2 is fully instantiated, Task Step 1 redirects to the URL of the Task Step 2 interface, and the second task interface opens.
- Again, Task Step 2 has no functionality other than to display the Loan Number entered in the StartLoanProcess portlet.

4. Click **Complete**.

This completes the task workflow, and the process instance transitions to the End service task:

- The service activity is configured to run the IS service formFlowComplete in the WmFormFlowTest package.
- The formFlowComplete service contains the WmTaskClient service `pub.task.taskclient:formFlowTaskNotify`, which enables you to pass a correlation ID, a result, as well as the Loan document.
- This step can also be configured to provide any required error handling
- With the completion of the End service activity, the original StartLoanProcess portlet interface appears in My webMethods, and the entire process can begin again. However, if necessary, the business process could continue on from the service step to additional downstream steps in the process.

This workflow can be extended with as many user tasks as necessary. In addition, the logic capabilities of both the process instance and the user task steps can be implemented to provide more advanced handling. For example:

- You can implement task events, task assignment, and the task control set functionality when you design the tasks to apply a certain level of conditional behavior within the tasks applications.
- The task activity steps themselves can be configured with process behavior such as transition looping, join logic, and If Condition transitions for further flexibility. For more information, see the *webMethods BPM Process Development Help* in Software AG Designer.
- Using the available process step logic, you can add service activity steps to the process model to provide any process work that is needed to support the task workflow.

Key Implementation Points

As you examine the sample implementation, make note of the following points:

Single path processing only

Software AG Designer enables you to create parallel data flows in a process model through the use of gateways and step joins and transitions. However, it is not possible to use a task workflow with parallel process branches where each branch might queue its own task instance. That is, you must avoid creating a situation where more than one task instance is running in parallel.

Task workflow correlation ID

A primary concept with the task workflow is that each object in the workflow must wait for the next task to fully instantiate before a transition occurs. This requires you to add a wait mechanism to each component in the workflow. You use a task workflow correlation ID to synchronize the waiting process component and the new task being queued.

This will ensure correct data flow through the process. The value of the task workflow correlation ID must be unique within the Process Engine environment.

Important:

The task workflow correlation ID is completely different from and unrelated to the standard document correlation ID often used in process implementation.

Prepare to wait

The first requirement in implementing the wait mechanism is to call:

```
TaskHelper.getTaskFormFlowService().waitPrepare(correlationID)
```

This returns a wait object to be used later to actually wait for notification with the correlation ID. It is very important to call `waitPrepare()` *before* starting a process or completing a task. Because these activities are executed in parallel, there is chance that the notification could be published *before* the wait is started and would then be missed.

Note:

When the very first activity step in your process model is a workflow task, you must implement the wait mechanism in the object that starts the process. For example, in the sample `StartLoanProcess` portlet, when the user clicks the **Start Loan Process** button, the first activity is to invoke `waitPrepare`. Then, the portlet starts the loan process. Example code can be found at: `FormFlowProcessTasks\src\com\webmethods\caf\startloanprocess\StartLoanProcessDefaultviewView.java`

Waiting for task instantiation

The next part of the code is to wait for the new task being queued:

```
TaskHelper.getTaskFormFlowService().wait(waitObject, timeoutinmillis)
```

Notification

After the task is fully instantiated by the Process Engine, the task must notify any objects that are waiting for it. This is accomplished by adding a Queued event to the task and implementing an Invoke Service task action for the event that calls the following API:

```
TaskHelper.getTaskFormFlowService().notify(correlationID, result, false)
```

where `result` can be any value to be passed back to the waiting component. In the example, the result passes the taskURL of new task being queued. The code be found at:

```
FormFlowProcessTasks\src\com\webmethods\caf\taskclient\TaskStep1RuleContext.java
```

The `Queued()` action is the service invoked by the Task Queued event.

When the `wait()` call returns, the code checks the result of `waitObject.getResult()` which returns information from new task being queued. In the sample, this is the taskURL of the task being started.

Completing a task and waiting for the next task in the workflow

The basic behavior here is similar to that of starting a new process. However, instead of starting a process, the code completes the task and then waits for the next task to be queued. See the code sample in:

```
FormFlowProcessTasks\src\com\webmethods\caf\taskstep1view\
TaskStep1ViewDefaultviewView.java
```

The `completeTask()` action method addresses the completion requirement.

Ending the workflow from the process

Eventually a workflow task does not transition to a another workflow task, but instead completes the business process. In this case, the original CAF application may be waiting to be notified, but there is no new task to publish a notification. In this case it is the process model that knows the task workflow is complete.

The `pub.task.taskclient:notifyFormFlow` service in the `WmTaskClient` package accommodates this use case, enabling you to pass the correlation ID and a result:

```
pub.task.taskclient:notifyFormFlow (correlationID, result)
```

This service serves the same purpose as the `notify()` method of the Java API. It notifies any object waiting for a correlationID and passes a result. The result can be any value that the waiting thread can interpret as a signal to finish the task workflow (for example, it could be NULL, or it could be specific hard-coded value).

For more information about this and other task services, see the PDF publication *webMethods Task Engine API and Service Reference*. For general information about Task Engine services, see [“Working with the Task Engine APIs” on page 259](#).

22 Working with Tasks in the Faces Configuration Editor

■ About Tasks and the Faces Configuration Editor	232
--	-----

About Tasks and the Faces Configuration Editor

Like other JSF application portlets, task portlets can be examined and modified with the Faces Configuration editor available within each project in the Solutions view. Among other things, you can define navigation rules for task views, and work with managed beans associated with a task.

To access the Faces Configuration editor in the Designer Solutions view: **User Interfaces > [Task Application] > Faces Configuration**.

23 Finding Task Applications in Software AG Designer

■ Searching for Task Applications	234
---	-----

Searching for Task Applications


After you create a task, there are several methods for locating it within Software AG Designer. You can:

- Find it in the Solutions view by expanding a solution folder and then expanding the Tasks folder.
- Find the task by using the webMethods search feature:
 1. With Designer running, press CTRL+H to open the Search dialog box.
 2. Click the **webMethods** tab (not **webMethodsAdvanced**).
 3. In the **Asset Type** box, select CAF Task Type.
 4. In the **Name or description contains** box, type the task search term.
 5. Click **Search** and view the result in the Search view.

You can drag a task from the Search view onto the process editor canvas.

Tip:

You can create a search with more specific criteria on the **webMethodsAdvanced** tab. Select CAF Task Type in the **Asset Type** box and then specify additional search criteria.

- Find a task from within a process that contains a task:
 1. Open the process in the process editor.
 2. In the process editor, click any task step to select it.
 3. In the Properties view, click the **Implementation** tab.
 4. Click the  button inside the **Task Information** area. The Tasks dialog box appears, displaying a list of all the task applications in your workspace.

24 Working with Task Metadata

■ Task Metadata Overview	236
--------------------------------	-----

Task Metadata Overview

Metadata is data that describes a webMethods asset such as a task; for example, all tasks have names and various properties, such as the **Task Type ID**. If an asset references other assets (for example, when a task step references a service), that asset will have metadata about such associations.

In Designer, metadata is created for tasks, processes, portlets, Integration Server documents, e-forms, Integration Server services, rule services, and Web services. You can search the available metadata to locate assets.

Local metadata that is stored on your computer is referred to as *workspace metadata*. Workspace metadata is stored in a database, on your computer, called the *workspace index*. For more information about searching for and working with local metadata, see *webMethods BPM and CAF Workspace Metadata Help*.

When CentraSite is installed as part of your Software AG installation, CentraSite serves as a central repository for metadata for each Designer asset. You can *publish* assets to CentraSite and *retract* assets from it. When you publish metadata to CentraSite, it is stored in the CentraSite repository, and it is available for use by other users who have access to CentraSite. For more information about search for and working with metadata in CentraSite, see *webMethods BPM Process Development Help*.

25 Working with User and Business Calendars

■ About Tasks and User and Business Calendars	238
---	-----

About Tasks and User and Business Calendars

Software AG Designer and My webMethods Servers support the use of business and personal user calendars to assist with task definition and behavior. Both business and user calendars are set up and configured in My webMethods Server; for more information, see the PDF publication *webMethods Task Engine User's Guide*.

- **Business calendars** define standard business days and hours for your business organization, including holidays, weekends, or any other times when your organization is not conducting business. For example, you might define your business calendar for normal business hours of Monday through Friday, 8:00 A.M. to 5:00 P.M. Eastern Standard time.

These business calendars are defined in My webMethods Server and can be specified when you define a task date/time event type, for example. This ensures that when counting days, only business days will be considered and that non-business days such as weekends and holidays are not included.

Note:

You can create as many business calendars as you want; for example, it is possible to create a business calendar for each user in your system, assigning each business calendar an alias equal to the user ID or domain name to aid in lookup.

- **User calendars** are maintained in a third-party application such as Microsoft Exchange or Lotus Notes, where the user maintains daily calendar events that define the user's availability. This enables you to determine if the user is available on a particular working day, taking into consideration only Out of the Office and Busy types of calendar events that are *scheduled for the entire day*. For additional information on checking personal calendars, see [“About User Calendars and Task Distribution Management” on page 135](#).

26 Importing Task Applications from Earlier Versions

■ Importing Task Applications Overview	240
■ Indexed Field Considerations When Importing into Version 8.2 or Later	240
■ Moving from JSF 1.x to JSF 2.x Support	240

Importing Task Applications Overview

If you plan to migrate existing task applications developed with earlier versions of Software AG DesignerTask Development, you should be aware of the following items:

- [“Indexed Field Considerations When Importing into Version 8.2 or Later” on page 240](#)
- [“Moving from JSF 1.x to JSF 2.x Support” on page 240](#)

Indexed Field Considerations When Importing into Version 8.2 or Later

Beginning with webMethods Product Suite Version 8.2, task types can be implemented with indexed business data fields, a feature unavailable in earlier versions. For more information about the advantages of working with indexed task business data, see [“About Indexed Business Data Fields” on page 89](#).

The run-time search behavior of task applications imported from earlier versions will continue to work as in earlier releases, with no modifications necessary. However, if you want to implement indexed business data in a task type imported from an earlier version to take advantage of faster, scalable business data searches, you must modify the task type in the task editor of Software AG Designer Version 8.2.

The task editor features an option in the Task Client Options wizard to assist with updating tasks to work with indexed business data; this option works best with task types that are implemented with the default components and user interfaces provided by the task editor at creation time.

However, the task editor also enables the task developer to customize the task type extensively; in these cases, the wizard may not be able to complete the conversion automatically and some additional manual configuration may be necessary.

Before you apply these procedures, you must import your 8.0 SP2 and earlier task projects and applications into Software AG Designer Version 8.2 or later. You do *not* need to apply these procedures if you are importing tasks from Version 8.2 or later.

To convert a standard search content provider to an indexed search content provider, follow the procedures in [“Implementing an Indexed Search Content Provider for Existing Task Types” on page 161](#).

Moving from JSF 1.x to JSF 2.x Support

Task application projects created in Task Development Help version 9.6 are implemented with JSF 1.x support only. These task application projects will continue to work as expected in version 9.7 and later.

However, if you want to take advantage of JSF 2.x functionality in your task project, you can migrate an older project to JSF 2.x after you have imported it. For more information, see [“Migrating Task Projects and .view Files to JSF 2.x Format” on page 24](#).

27 Working with Task Expressions

■ About Task Expressions	242
■ Task Expression Overview	242
■ Configuring Advanced Task Expressions	243
■ Building a Sample Task Expression	244
■ Viewing Sample Task Expression Syntax	245
■ About Condition Expressions	246
■ About Action Expressions	246
■ About Filter Expressions	247
■ About Expression Operators	247
■ About Date/Time Values in Expressions	249
■ About Expression Managed Beans	249

About Task Expressions

The task expression language is very flexible and can be used to:

- Detect conditions pertaining to task status, business data, and other attributes
- Update data in a task
- Invoke an action on a task
- Send notifications
- Invoke actions in a service or a managed bean associated with the task.

For general information about task expressions, see [“Task Expression Overview” on page 242](#). For specific information, see the following topics:

- [“Configuring Advanced Task Expressions” on page 243](#)
- [“Building a Sample Task Expression” on page 244](#)
- [“Viewing Sample Task Expression Syntax” on page 245](#)
- [“About Condition Expressions” on page 246](#)
- [“About Filter Expressions” on page 247](#)
- [“About Expression Operators” on page 247](#)
- [“About Date/Time Values in Expressions” on page 249](#)
- [“About Expression Managed Beans” on page 249](#)

Task Expression Overview

The task expression language used within Software AG Designer is proprietary and is essentially simplistic in nature. However, it does offer binding capability to a wide range of data within the task environment, and all the bindings are JSF binding expressions as well. You can find more information about working with the JSF expression language from various sources on the Web.

Task expressions fall into the following general categories:

- *Condition expressions* that must be matched for an assignment to be applied.
- *Action expressions* that carries out some user-defined activity.
- *Filter expressions* that define which tasks appear in the user's inbox.

In addition, you can also use expression terms to create bindings to run-time data associated with the task.

A basic expression statement consists of the following:

[A data field] [An operator] [A data field]

For example, the following condition expression determines if the current task status is "active:"

```
#{currentTask.taskInfo.status} == "active"
```

For more information on task status, see [“About Task Status” on page 30](#).

When an expression contains two or more elements, the elements must be separated by a semi-colon (;). For example:

```
#{currentTask.taskInfo.priority} = "high"; #{currentTask.applyChanges};
```

You can build advanced task expressions with the Edit Expression dialog box used for defining advanced conditions and event actions. For more information about advanced task expressions, see [“Configuring Advanced Task Expressions” on page 243](#).

For further information about creating task expressions, see the following topics:

- [“Configuring Advanced Task Expressions” on page 243](#)
- [“Building a Sample Task Expression” on page 244](#)
- [“Viewing Sample Task Expression Syntax” on page 245](#)
- [“About Condition Expressions” on page 246](#)
- [“About Filter Expressions” on page 247](#)
- [“About Expression Operators” on page 247](#)
- [“About Date/Time Values in Expressions” on page 249](#)
- [“About Expression Managed Beans” on page 249](#)

Configuring Advanced Task Expressions

If you cannot create the assignment or event expressions you want with the simple conditions and actions editors, you can create advanced (custom) syntax-based expressions to govern the behavior of a task.

➤ To configure advanced task expressions

1. Open the task in the task editor by double-clicking the task in the Solutions view.
 - For assignments, click the **Assignments** tab and click the assignment you want to work with. Expand the Assignment Options and Conditions area if it is not already visible. In the **Condition Type** box, select **Advanced condition expression**. The Advanced condition editor appears immediately below the list.
 - For events, click the **Events** tab and click the event you want to work with. Expand the Event Type area if it is not already visible. In the **Event Type** box, select **Advanced** (you might have to scroll the list up to find this entry). The Advanced event condition expression editor appears immediately below the list.

Note:

If you have already created simple conditions for a task assignment, they will appear in the advance condition editor in their text-based format. You must delete the simple conditions and save the task to clear this display

2. Click **Edit** to open the Edit Expression dialog box.
3. Create the advanced conditions expression you want to apply either by selecting the expression terms with the available buttons, or by typing in the expression directly. For more information about creating expressions, see [“Task Expression Overview” on page 242](#) and [“Building a Sample Task Expression” on page 244](#).
4. Save the task to apply your changes.

Building a Sample Task Expression

Although this sample creates a very simple condition expression, you can use the basic procedure to build advanced task expressions for both conditions and actions. The following condition expression determines if the current task status is "active:"

```
#{currentTask.taskInfo.status} == "active"
```


You select expression terms by clicking the **Add Field** and **Add Operator** buttons on the Edit Expression dialog box, and then selecting the data field or operator from the resulting lists. Various operators such as AND and OR enable you to combine data tests in a single condition.

- For more information about condition expressions, see [“About Condition Expressions” on page 246](#).
- For more information about action expressions, see [“About Action Expressions” on page 246](#).

➤ To create a simple condition expression

1. Open the advanced expression editor in the task editor as described in [“Configuring Advanced Task Expressions” on page 243](#).
2. Click the **Add Field** button in the Edit Expression dialog box.
3. In the Expression Binding dialog box, expand **Current Task State > Task Info**, and then click **Status**.
4. The expression element `#{currentTask.taskInfo.status}` appears in the box at the top of the dialog box.

Tip:

Click  in the tool bar to show all available properties. All expressions are checked for validity as they are created and syntax errors are displayed as they are typed. Click a field

or operator and click **OK** to add it to the Edit Expression dialog box. Near the top of the Add Field display is the entry **Managed Beans**. You can create your own custom Java methods for providing data and include them in the task managed beans. They will then be available for use in expressions from this location.

5. Click **OK**.
6. Place the cursor at the end of the expression in the Edit Expression dialog box.
7. Click the **Add Operator** button.
8. In the Choose Operator dialog box, click the "equals" operator (=).
9. Click **OK**.
10. Place the cursor at the end of the expression in the Edit Expression dialog box.
11. Click the **Add Field** button on the Edit Expression dialog box.
12. In the Expression Binding dialog box, expand **Task Status** and then click **Active**.
13. The expression element "active" appears in the box at the top of the dialog box.
14. Click **OK**.

Viewing Sample Task Expression Syntax

One way to see various implementations of expression language is to examine the syntax of the existing simple conditions available within Software AG Designer. This syntax is not visible by default; use the following procedure to view the syntax.

➤ To view the syntax of a simple condition or action

1. In the task editor, add a simple condition or action to an assignment or event.
2. Save the task.
3. Change the condition type or action type from "Simple" to "Advanced."
4. The syntax of the simple condition or action is displayed in the **Condition** box for conditions, and in the **Results** box for actions.
5. Click **Edit** to view the expression in its entirety, or to modify it as needed.

About Condition Expressions

You can create a condition expression to test for a wide range of task characteristics, including:

- Current status or previous status
- Priority
- Input and output values
- Escalation
- Dates and times associated with the task
- Current user

For example, this condition expression tests for a task status of active:

```
{currentTask.taskInfo.status} == "active"
```

The results are returned a simple Boolean true or false. This simple expression was built by clicking the **Add Field** and **Add Operator** buttons on the Edit Expression dialog box. For more information about building expressions, see [“Configuring Advanced Task Expressions” on page 243](#) and [“Building a Sample Task Expression” on page 244](#).

This more complex statement tests for the expiration of an active task:

```
{currentTask.taskInfo.status} == "active" && ({System.currentDateTime} +  
(ExpirationThresholdDays * 86400000)) > {currentTask.taskInfo.expireDate}
```

You can view the available data fields and operators by clicking the **Add Field** and **Add Operator** buttons on the Edit Expression dialog box.

About Action Expressions

You can create an action expression to apply various changes to a task. Because of the wide range of capability with the task expression language, you must specify that you want to apply a change to the task when you create an expression that directly affects some aspect of the task. Such changes include:

- Setting the task assignment
- Setting the task status
- Setting the task priority
- Resuming the task
- Setting dates and times associated with the task

When you update a task with an expression, first assign values and then *always* conclude the expression with the `applyChanges (Update)` action or the `applyChangesNoAccept` action. For example, this action expression sets the task priority to high:

```
#{currentTask.taskInfo.priority} = "high"; #{currentTask.applyChangesNoAccept};
```

This expression was built as follows:

- The `#{currentTask.taskInfo.priority}` portion was selected from the Add Field list.
- The "assignment" operator `=` was selected from the Add Operator list.
- The priority value "high" was selected from the Add Field list.
- The second element in the expression. `#{currentTask.applyChangesNoAccept}`, was chosen from the Add Field list.

If the `applyChangesNoAccept` action is not present, the action will not be applied to the task.

When setting the task status to Complete, the `applyChangesNoAccept` action is not required:

```
#{currentTask.completeTask}
```

About Filter Expressions

Two of the available filter actions contain hidden expressions that filter certain tasks from the task inbox of a user, group, or role. For reference, the following expressions are used:

- For the Filter Inbox - Accepted Tasks action:

```
(isEmpty #{currentTask.taskInfo.acceptedByList}) ||  
(#{currentTask.taskInfo.acceptedByList} contains #{currentUser.principalID})
```

This condition allows a user to see only tasks that have not yet been accepted, or tasks accepted by the user. This effectively filters out tasks accepted by other users.

- For the Filter Inbox - User Activity action:

```
(#{fieldExpr} doesNotContain #{currentUser.principalID} && #{fieldExpr}  
doesNotContain #{currentUser.principalURI} && #{fieldExpr}  
doesNotContain #{currentUser.principalDN})
```

This condition filters out tasks that the current user has already worked on. It requires you to select a custom business data field or service that returns a list of the users that have worked on the task. You must also configure the custom field or service to be updated by the Task Details portlet or by another event action.

About Expression Operators

You can view the available expression operators by clicking the **Add Operator** button on the Edit Expression dialog box. The following table lists the available operators:

Operator	Definition
+	Add
	Note:

Operator	Definition
	This operator works only for number types. It is not supported for strings and will generate an error when used with string values.
-	Subtract
*	Multiply
/	Divide
%	Modulus (Absolute value, or dividing without a remainder)
^	Exclusive OR
>	Greater than
<	Less than
==	Equals
=	Assignment (sets a value)
!=	Does not equal
>=	Greater than or equal
<=	Less than or equal
&&	Logical AND
	Logical OR
!	Logical NOT
Contains	The preceding string, collection, or object array contains the following string
Does Not Contain	The preceding string, collection, or object array does not contain the following string
Is Empty	The preceding collection, string object, or object array is empty
Not Empty	The preceding collection, string object, or object array is not empty
Matches	The preceding sting matches the following regular expression
Starts With	The preceding string starts with the following string
Ends With	The preceding string ends with the following string
Semi-colon	Statement separator

About Date/Time Values in Expressions

You can specify date and time values in an expression; for example:

```
#{System.currentTimeMillis}
```

or

```
#{currentTask.taskInfo.expireDate}
```

These date and time values are evaluated as follows:

If a given data element within an expression is a `java.util.Date` type, it is automatically converted into epoch time in milliseconds as a `java.lang.Long` type. Therefore, all arithmetic and comparison operators can be applied to that value, as it now exists as a long number.

About Expression Managed Beans

When a condition or action expression is evaluated, the Task Engine creates and makes available specialized managed beans. You can also create custom objects for use with expressions as long as they are defined as managed beans in the task application (just as you would when using binding expressions in a CAF application).

For example, additional beans are used when a task action publishes a notification or uses a business calendar to evaluate date/times; these beans are usually auto-generated when the corresponding data element or action is specified in the task expression editor. It is also possible to manually define them as managed beans in the application and use them in the rule expressions.

Following are managed beans that are automatically defined when evaluating expressions:

- `#{currentTask}`—this is an instance of the specific `TaskContentProvider` class for the task being assigned. That is, it is the same class that is used in the task details portlet. Any property of `TaskInfo` and `TaskBusinessData` or any action can be accessed from it using the same binding expressions used on a task details portlet when rooted from `#{currentTask}`.
- `#{currentUser}`—identifies the current user on behalf of whom these expressions are being evaluated. `#{currentUser}` is only available when a request originates from a user; for example, it is not available for task events, which are processed in the background. `#{currentUser}` is contained in the class `com.webmethods.caf.faces.data.dir.UserModel`, which provides access to basic and advanced attributes of a user object

Some examples of property binding expressions:

- `#{currentTask.taskInfo.name}`—name of the current task.
- `#{currentTask.taskInfo.assignedToList}`—list of principals to whom the task is assigned.
- `#{currentTask.taskData.order.orderNumber}`— the order number from a business data order document attached to the task.

Some examples of action binding expressions:

- `{currentTask.completeTask}`—completes a task (though it does not make sense to complete a task from assignment)
- `{currentTask.applyChangesNoAccept}`—updates the task instance by applying any changes specified to `TaskInfo` and `TaskBusinessData`
- `{currentGroup}`—an instance of `PortalUserModel` that represents the group principal to which task is to be assigned. This can be used to obtain group-specific attributes to define assignment conditions.
- `{currentRole}`—an instance of `PortalGroupModel` that represents the role principal to which task is to be assigned. This can be used to obtain role-specific attributes to define assignment conditions.
- `{currentUser}`—an instance of `PortalRoleModel` that represents the user principal to which task is to be assigned. This can be used to obtain user-specific attributes to define assignment conditions.
- `{<TaskName>RuleContext}`—This context object is always available for task rules expression evaluation (for both assignments and events). This is an object which is exposed in the bindings view under "Services" and which is usually used to expose custom actions/properties (like web service connectors) for task rules

Expressions For Task Events

In addition to `{currentTask}`, the Task Engine also exposes `{oldTask}` and `{newTask}` objects; these simply represent the task state before and after a task update. Having these is useful when an event is to be triggered upon some specific modification. For example, the following is a condition for Queued event:

```
{oldTask.taskInfo.status} == "new") && ({newTask.taskInfo.status} == "active")
```

This expression detects the change of status field from "new" to "active".

The `{currentUser}` object is not available for task events.

Any update actions applied to a task can be executed only against a `{currentTask}` object, but not against `{oldTask}` or `{newTask}`; the latter are essentially read-only containers of data.

For example, the following action expression updates the name of the task and some task business data, followed by an action to apply the changes.

```
{currentTask.taskInfo.name} = {currentTask.taskData.order.orderNumber};
```

```
{currentTask.taskData.orderValue} = ({currentTask.taskData.orderValue} * 100) + 10;
```

```
{currentTask.applyChangesNoAccept};
```

28 About Java Providers and Related Classes

■ Understanding Task Providers	252
--------------------------------------	-----

Understanding Task Providers

As you develop task portlet and web applications with webMethodsTask Development, you will find that much of the interaction between the user interface and the underlying task APIs is handled by various provider classes (also known as Service Provider Interfaces). These task providers are included as part of the webMethodsTask Development feature, and they are created automatically for the default user interfaces generated with the task creation wizard.

For more information about the task providers classes, see *webMethods CAF and My webMethods Server Java API Reference*

BPM Process Content Provider

Name	BPMProcessContentProvider
Use:	Manages collaboration tasks associated with a single task instance. Enables creation, retrieval, deletion, modification, and listing of collaboration tasks.
Interfaces with:	Task Collaboration page
Class name:	com.webmethods.caf.faces.data.bpm.BpmProcessContentProvider

Task Attachments Provider

Name	TaskAttachmentsProvider
Use:	Manages task attachments for a single task instance. Enables creation, retrieval, deletion, modification, and listing of attachments.
Interfaces with:	Typically Task Data View portlet when attachments support is added (or can be any other task portlet)
Class name:	com.webmethods.caf.faces.data.task.TaskAttachmentsProvider

Task Audit Content Provider

Name	TaskAuditContentProvider
Use:	Returns a list of audit entries associated with a single task instance.
Interfaces with:	Task Audit View portlet
Class name:	com.webmethods.caf.faces.data.task.impl.TaskAuditContentProvider

Task Audit Display Provider

Name	TaskAuditDisplayProvider
Use:	Facilitates display of task audit data associated with a single task instance.
Interfaces with:	Task Audit View portlet
Class name:	com.webmethods.caf.faces.data.task.TaskAuditDisplayProvider

Task Business Calendar Model

Name	TaskBusinessCalendarModel
Use:	Used indirectly inside Data/Time task events. Can also be used for calendar calculations in custom code (that is, it is not tied to any particular task portlet).
Interfaces with:	Various components.
Class name:	com.webmethods.caf.faces.data.task.TaskBusinessCalendarModel

Task Comment List Provider

Name	TaskCommentListProvider
Use:	Manages task comments for a single task instance. Enables creation, retrieval, deletion, modification, and listing of comments.
Interfaces with:	Task Comments page
Class name:	com.webmethods.caf.faces.data.task.TaskCommentsListProvider

Task Content Provider

Name	TaskContentProvider
Use:	Manages task business data content for a single task instance. Enables creation, retrieval, deletion, and modification of task business data. Note that there is an extended sub-class of this provider that provides additional functionality; see the Javadocs for details. In addition, when an inbox is generated, another extended sub-class is created as part of the task application to define the business data class to be used.

Interfaces with: Pages that interact with task business data.

Class name: com.webmethods.caf.faces.data.task.impl.TaskContentProvider

Task Display Provider

Name TaskDisplayProvider

Use: Facilitates display of task properties associated with a single task instance.

Interfaces with: Task Details View portlet.

Class name: com.webmethods.caf.faces.data.task.TaskDisplayProvider

Task Indexed Search Content Provider

Name TaskIndexedSearchContentProvider

Use: Handles task search queries from the user interface directed at the indexed field table in the task database. Implements a search using the specified terms and returns a list of matching tasks. If no indexed fields are specified in the task business data, this search provider will return no results regardless of the search terms. Note that there is an extended sub-class of this provider that provides additional functionality; see the Javadocs for details. In addition, when an inbox is generated, another extended sub-class is created as part of the task application to define the business data class to be used.

Interfaces with: Task Inbox Results page

Class name: com.webmethods.caf.faces.data.task.impl.TaskIndexedSearchContentProvider

Usage notes: When implementing sorting by column with this provider, the sort order specified for the table control columns must be defined using a binding expression that references the valid task indexed data fields that are present in the database; references to non-indexed data fields will fail. In addition, when this provider sorts on the fields that contain user names (for example, Created By, Last Updated By, or Accepted By), it does not sort on the full name of the user, but on the user ID (UID) value. For more information about indexed data fields, see [“About Indexed Business Data Fields” on page 89](#).

Task JCR Attachments Provider

Name	TaskJcrAttachmentsProvider
Use:	Provides the same functionality as the Task Attachments Provider (manages task attachments for a single task instance, enabling creation, retrieval, deletion, modification, and listing of attachments) via JCR APIs.
Interfaces with:	Various components.
Class name:	com.webmethods.caf.faces.data.task.TaskJcrAttachmentsProvider

Task Search Content Provider

Name	TaskSearchContentProvider
Use:	Handles task search queries from the user interface. Implements a search using the specified terms and returns a list of matching tasks. Note that there is an extended sub-class of this provider that provides additional functionality; see the Javadocs for details. In addition, when an inbox is generated, another extended sub-class is created as part of the task application to define the business data class to be used.
Interfaces with:	Task Inbox Results page
Class name:	com.webmethods.caf.faces.data.task.impl.TaskSearchContentProvider
Usage notes:	When implementing sorting by column with this provider, the sort order specified for the table control columns can be defined using any valid binding expression that references any available business data field, or even on a dynamically calculated value (for instance, a delegated/accepted column).

Non-Public Task Providers

The task provider classes in this section are presented for informational use only. They are not available for use by task developers.

Task Inbox Queue Workload Provider

Name	TaskInboxQueueWorkloadProvider
Use:	Used internally inside workload-based assignment rules. <i>Not for end user implementation.</i>

Interfaces with: Various components.

Class name: com.webmethods.caf.faces.data.task.TaskInboxQueueWorkloadProvider

Task Round Robin Workload Provider

Name TaskRoundRobinWorkloadProvider

Use: Used internally inside workload-based assignment rules. *Not for end user implementation.*

Interfaces with: Various components.

Class name: com.webmethods.caf.faces.data.task.TaskRoundRobinWorkloadProvider

Task Saved Search Content Provider

Name TaskSavedSearchContentProvider

Use: Used internally from the Task Chart portlet to load task saved searches. *Not for end user implementation.*

Interfaces with: Task Chart portlet.

Class name: com.webmethods.caf.faces.data.task.TaskSavedSearchContentProvider

Task Provider Utils

Name TaskProviderUtils

Use: Used internally to support various task provider functions. *Not for end user implementation.*

Interfaces with: Various components.

Class name: com.webmethods.caf.faces.data.task.TaskProviderUtils

29 Processing Task Change Events

■ Task Change Event Overview	258
■ Typical Use of Task Change Events	258

Task Change Event Overview

The webMethods Task Engine generates an internal task change event each time a change occurs to a task instance in the run time. For example, a change of status, a delegation, an assignment, or the addition or removal of a task attachment. These events are instances of `com.webmethods.portal.service.task.event.ITaskChangedEvent`.

Typical Use of Task Change Events

You can subscribe to these events and receive them in your application using `CustomTaskChangeHandler`. For example, suppose a user wants to store all the tasks in a custom data store for analysis. To do so, the user can first subscribe to task change events and listen to the events that contain all the information about the tasks. Then, this API can be used to extract the required information from the events and insert it into their custom data store.

The events can be subscribed by implementing the `ICustomTaskEventSubscriber` interface:

`com.webmethods.portal.mech.task.impl.CustomTaskChangeHandler.ICustomTaskEventSubscriber`

This interface contains two methods:

- `getDescription()` This returns a string that can be used to describe the subscriber (for usability only).
- `deliver(ITaskChangedEvent)` This method is called for each `ITaskChangedEvent`.

The subscribers are registered and maintained using `CustomTaskChangeHandler`:

`com.webmethods.portal.mech.task.impl.CustomTaskChangeHandler`

Which contains these methods:

- `addSubscriber(ICustomTaskEventSubscriber)` Adds the passed subscriber, and begins delivering events.
- `removeSubscriber(ICustomTaskEventSubscriber)` Stops event delivery and removes the subscriber.
- `ICustomTaskEventSubscriber[] getSubscribers()` Returns an array of all the current `ITaskChangedEvent` subscribers (for usability only).

For more information about these components, refer to the *webMethods CAF and My webMethods Server Java API Reference*.

30 Working with the Task Engine APIs

■ About the Task Engine APIs	260
------------------------------------	-----

About the Task Engine APIs

The webMethods Task Engine provides Application Programming Interfaces (APIs) that enable you to search and manipulate tasks from services or other applications outside of the My webMethods environment. These APIs are provided as Java classes, and as services in the WmTaskClient package.

The Java APIs are created when you create a task application portlet. For more information, see [“Available Java APIs” on page 154](#).

The following WmTaskClient APIs are available:

- **Built-in service API.** The WmTaskClient package contains built-in Java services that you can use as templates to create custom services in Designer that can create, modify, locate, queue, and delete tasks on the Task Engine to which the Integration Server is connected.
- **RESTful service API.** The WmTaskClient package contains RESTful services that you can use to create GET, DELETE, PUT, and POST requests for task instances, task audit information, task attachments, and task comments, routed through Integration Server or My webMethods Server.
- **Web service API.** webMethods Task Engine contains a Web service API that enables you to add, delete, locate, and update tasks on a webMethods Task Engine from a client program. The Task Engine hosts two versions of the task Web service, one version for .NET clients and the other for regular Web-service clients. Both versions of the Web service provide the same set of operations, however, the schemas for the underlying data structures differ slightly between the two.

For additional information about the available APIs, see these resources:

- Documentation for the Task Engine APIs described above is available in the PDF publication *webMethods Task Engine API and Service Reference*.
- For more information about implementing the Task Engine REST services in Integration Server, see the Integration Server PDF publication *REST Developer's Guide*.