Task Development Help

webMethods Designer

Version 7.2

December 2008

webMethods

This document applies to webMethods Designer Version 7.2 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 © 2006—2008 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, United States of America, and/or their suppliers. All rights reserved.

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

Document ID: DES-TDH-72-20081212

Table of Contents

About This Guide

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 webMethods 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 user interface.
Narrow font	Identifies storage locations for services on webMethods Integration Server, using the convention *folder.subfolder:service*.
UPPERCASE	Identifies keyboard keys. Keys you must press simultaneously are joined with a plus sign (+).
Italic	Identifies variables for which you must supply values specific to your own situation or environment. Identifies new terms the first time they occur in the text.
`Monospace font`	Identifies text you must type or messages displayed by the system.
{ }	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 (...).

Additional Information

You can find additional information about webMethods products at the locations described below.

webMethods Central Documentation Directory

During product installation, you can download the webMethods product documentation to a single directory called "_documentation." This directory is located by default under the webMethods installation directory.

webMethods Advantage Bookshelf

The webMethods Advantage Web site at http://advantage.webmethods.com provides you with important sources of information about webMethods products:

- **Troubleshooting Information.** The webMethods Knowledge Base provides troubleshooting information for many webMethods products.
- **Documentation Feedback.** To provide feedback on webMethods documentation, go to the Documentation Feedback Form on the webMethods Bookshelf.
- **Additional Documentation.** You can find documentation for all webMethods products on the webMethods Bookshelf.

Software AG Developer Community

Additional articles, demos, and tutorials are available on the webMethods portion of the Software AG Developer Community. The various Developer Communities feature technical information, useful resources, and online discussion forums, moderated by Software AG professionals, to help you do more with webMethods technology.

With the Software AG Developer Communities, you can:

- Use the online discussion forums to exchange best practices and chat with other experts.
- Expand your knowledge with product documentation, code samples, articles, online seminars and tutorials.
- Link to external sites on open standards and many Web technology topics.
- See how other customers are streamlining their operations with technology from Software AG.

1 Task Development Help

webMethods Task Development

webMethods Designer 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.

Task development is covered in the following online help topics. For additional information on working with tasks, see "Other Resources for Task Development" on page 12.

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 applications and Web services.
- The Java Platform, Enterprise Edition, formerly known as Java 2 Platform, Enterprise Edition/J2EE (http://java.sun.com/javaee/).
- Java Server Faces (JSF) technology (http://java.sun.com/javaee/javaserverfaces/).
- 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.

Related Topics

- "webMethods Task Development" on page 11
- "Other Resources for Task Development" on page 12
- "Setting Task Development Preferences" on page 14

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 locations:

Product Guides

- The *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 built-in services that are available for interacting with tasks from a remote server.

These PDF files are available in the _documentation directory of your webMethods installation; note that the _documentation directory is installed as a separate component in the webMethods Installer and may not be present. You can install the directory using the Installer or you can download these files from the Advantage Bookshelf Web site.

Web resources

You can find links to various Web-based Software AG product solutions, such as tutorials, examples, and technical articles, located in the webMethods Designer Online Help Overview.

Related Topics

- "Software AG Information Resources" on page 13
- "Software AG Training Resources" on page 13
- "webMethods Task Development" on page 11
- "Task Development Prerequisites" on page 12

Software AG Information Resources

The _documentation directory of your webMethods installation provides an excellent repository of product-related information. For example, if you are interested in an overview of the webMethods product suite or business process management, we recommend examining the following Software AG documents:

- *Understanding My webMethods Product Suite*
- *Getting Started with Business Process Management*

Note: The _documentation directory is installed as a separate component in the webMethods Installer. If the _documentation directory is not installed on your system, then you can install the directory by using the webMethods Installer, or you can download any document of interest from the Advantage Bookshelf Web site located at https://advantage.webmethods.com/bookshelf_cross_component.

Additionally, you might find the Software AG publication, *BPM for Dummies: A Reference Guide for the Rest of Us*, of particular interest. This book addresses the business management and information technology sides of BPM, and delves into the process-centric foundation upon which BPM is built. You can download the entire book at no cost from http://www.softwareag.com/corporate/res/books/default.asp.

Related Topics

- "Software AG Training Resources" on page 13
- "Other Resources for Task Development" on page 12
- "Task Development Prerequisites" on page 12

Software AG Training Resources

You might choose to benefit from Software AG training before you use webMethods products, although training is not a prerequisite. Software AG training provides extensive coverage of webMethods Designer functionality, with hands-on sessions and numerous exercises delivered by experienced Software AG instructors. For more information about available training courses and to schedule training for your organization, visit the webMethods Advantage Training center at https://advantage.webmethods.com/training.

Related Topics

- "Software AG Information Resources" on page 13
- "Other Resources for Task Development" on page 12
- "Task Development Prerequisites" on page 12

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, click 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	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. 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. 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 '_').

Preference	Description
Always use GUID for new Task Type ID	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: 1436DB52-6E63-311A-5D03-DD3E141EF483. When this preference is not selected, you have several choices when you are creating a new task. You can: ■ Manually request that Designer generate a GUID for the task type ID. ■ Type your own unique task identification. ■ Accept a task type ID based on the project name and task type name. In the latter case, a default name of [ProjectName].[TaskTypeName] is entered in the task ID box. As the task type name must be unique, this will create a unique task type ID. If you want, 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 '_'). This preference is ignored if the automatic task creation preference (above) is selected; in this case, a GUID is always automatically generated. This check box is selected by default.
Create Task View Portlet for new Task	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.
Create Task Start Portlet for new Task	Select this preference to always create a task start portlet 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.
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.

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.

Related Topics

- “Creating Tasks” on page 25
- “Working with Task Applications” on page 17
- “Task Overview” on page 20
- “webMethods Task Development” on page 11

Working with Task Applications

All tasks must be created within an existing task application project. The following topics provide additional information on understanding and working with task application projects.

For general information about task applications, see "Task Application Overview" on page 17. For specific information, see the following topics:

- "Creating a Task Application Project" on page 18
- "Deleting a Task Application Project" on page 19

Task Application Overview

A *task application* is a portlet application project (a type of dynamic web project). A task application project contains the following major components:

- One or more *tasks*, which contain:
  - One or more *portlets* that implement the task, which contain:
    - One or more *views* seen by the task user.

In addition, a task also contains supporting components such as assignments, events, rule sets, and notifications.

All tasks must be created within a task application project. The number of tasks in a task application project is determined primarily by the complexity of the human activity you are addressing, as well as by the amount of granularity you want to provide to your task structure. To deploy a task to My webMethods Server, 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 could 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.

Related Topics

- "Creating a Task Application Project" on page 18
- "Deleting a Task Application Project" on page 19
- "Task Overview" on page 20
- "Creating Tasks" on page 25

Creating a Task Application Project

A task must be created in an existing task application project, which is a type of portlet application project. For additional information about creating portlet application projects, see "Creating a Portlet Application" in the *webMethods Composite Application Framework Help*.

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

- On the My webMethods 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>Composite Applications>Portlet Application Project**.
- By clicking the **New>Other** command anywhere it is available, then expanding the **Software AG>Composite Applications** 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**.

To create a new task application project

1. Start the portlet application project wizard as described above.
2. When the Dynamic Web Project page appears, type a unique name for the task application project.
3. A task application project is created like any other portlet application project. You can click **Finish** to create the task application project using the default values, or you can click the **Next** button to specify custom values.

Related Topics

- "Deleting a Task Application Project" on page 19
- "Task Application Overview" on page 17
- "Task Overview" on page 20
- "Creating Tasks" on page 25

Deleting a Task Application Project

Task application projects can be deleted in webMethods Designer; deleted task application projects cannot be recovered.

Important! When you delete a task application project, all tasks within that project will be deleted and cannot be recovered.

To delete a task application project

1 Locate the task application project you want to delete in the Solutions view.

2 Right-click the task application project and click **Delete**.

If the task application project has previously been published to My webMethods Server, you may also want to delete the tasks contained in the task application project in My webMethods:

To delete a task application project from My webMethods Server

1 Log in to My webMethods Server with administrator privileges.

2 Navigate to **Administration>Business>Tasks>Task Engine Administration**.

3 If a task application contains two or more task types, sort the task list by clicking the **Task Application** column to group together all of the task types in that application.

Important! When you delete a task type, all task instances of that task type are deleted regardless of the task status, as well as all run-time objects associated with the task type; for example, all events, assignments, and pages. This action can delete currently running tasks with a status of Active. Exercise caution when deleting Active tasks.

4 Select the task types you want to delete and click **Delete**.

Note: This action does not remove the actual task application (*.war file) from My webMethods Server. Complete removal of the task application can be done using the Install Administration functionality of My webMethods Server (for more information, see "Uninstalling Server Components" in the *My webMethods Server Administrator's Guide*).

Related Topics

- "Task Application Overview" on page 17
- "Creating a Task Application Project" on page 18
- "Task Overview" on page 20
- "Creating Tasks" on page 25

Task Overview

When you create a *task* within webMethods 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 *process* developed and deployed with webMethods Designer.

When you publish a task application project from Designer to My webMethods Server, 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 viewed, modified, or used to start a run-time instance of a task.

The various component views for a task are implemented within the Designer environment as *portlets*, and a task can contain a single portlet, or two or more portlets. A task can be added to a process model, also created in Designer.

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 "Creating a Portlet Application" in the *webMethods Composite Application Framework Help*.

As you develop a task, you can preview the task within Designer to see how your user interface will appear to the task user. In addition, Designer contains a preview server that enables you to actually run the task (or any of its component portlets or pages) to test its behavior before you deploy it to a run-time server. For more information, see "Previewing a Portlet View" on page 55, "Running a Task Portlet View in the Designer Preview Server" on page 56, and "Working with Task Pages" on page 131.

Related Topics

- "About Task Characteristics" on page 21
- "About Task Status" on page 22
- "About Task Names" on page 23
- "Working with Task Applications" on page 17
- "Creating Tasks" on page 25
- "Working with Task Portlets" on page 33
- "Configuring Tasks" on page 65
- "Working with Task Notifications" on page 134

About Task Characteristics

When you create a task, you define how the task will appear and behave in the run-time environment of My webMethods 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.
- Information displayed to the user.
- Controls available to the user.
- One or more rule 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; see the *webMethods Task Engine API and Service Reference*, a PDF document available in the _documentation directory of your webMethods installation.

By defining task assignments at design time, a run-time task can be routed to one or more My webMethods roles or users based on those assignment definitions. Each task runs discretely within the My webMethods run-time, and can be monitored like any other run-time instance, and can be modified by a My webMethods Server administrator.

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 59 for design-time procedures, or the *webMethods Task Engine User's Guide* for run-time procedures, also available in the _documentation directory of your webMethods installation.

Related Topics

- "About Task Status" on page 22
- "About Task Names" on page 23
- "Task Overview" on page 20
- "Creating Tasks" on page 25
- "Configuring Tasks" on page 65

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 to users 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 *webMethods Task Engine User's Guide*.

Related Topics

- "About Task Characteristics" on page 21
- "About Task Names" on page 23

About Task Names

Tasks are identified by a *task type name* and by a *task type ID* (identification). These values must be unique within the Eclipse workspace you are working with. If the task type name or task type ID is not unique with the workspace, webMethods 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 any tasks you create are uniquely named and identified regardless of the workspace in which it is located. If you create tasks in different workspaces that share a task type name or task ID, you will experience problems with task operation when the tasks are published to the same My webMethods Server.

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

The task type name is displayed 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 117.

Task Type ID—This name is used by both My webMethods and webMethods Designer to identify a task programmatically. It cannot be modified.

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 14).

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.
- Manually request that Designer generate a GUID for the task type ID.
- Type your own unique task identification.

When you implement a new task from a process model, a GUID is always automatically generated.

About Task Step Names

The task step name (label) is used to create the task name when you implement a new task from within the process modeler. When you add a new task step to a process, you first add an activity step, and then change the step type to Task.

However, the step retains the default name of "Activity." Unless you rename the task step with a unique name before you implement the task step as a task, the task will either be created with the name "Activity," or, if a task named "Activity" already exists, the Task wizard will open to enable you to type a new task name. In this case, *be sure to rename the task step to use the new task 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 number. However, best practices recommend that you keep the display names synchronized. In the process editor, you can modify the task name of the activity step on the process canvas and on the **General** page of the Properties view. You can also click an existing task name on the **Implementation** tab; for more information, see "Configuring Task Steps" in the *webMethods Process Development Help*.

Related Topics

- "Changing the Task Type Name" on page 117
- "Creating Tasks" on page 25
- "Creating a New Task with the Task Wizard" on page 26
- "Adding a Task to a Process" on page 29
- "Configuring Tasks" on page 65
- "Task Overview" on page 20
- "Working with Task Portlets" on page 33
- "Working with Task Notifications" on page 134

Creating Tasks

A task can only exist as part of a task application project; for information on creating a task application project, see "Creating a Task Application Project" on page 18.

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

- On the My webMethods 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 starting the Task wizard from the **New>Other** menu command anywhere it is available.
- 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 **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 document or Java types to define input or output data, ensure that the document or Java types have been created and are available.

Related Topics

- "Creating a New Task with the Task Wizard" on page 26
- "Adding a Task to a Process" on page 29
- "Configuring Tasks" on page 65
- "Task Overview" on page 20
- "Working with Task Portlets" on page 33
- "Working with Task Notifications" on page 134

Creating a New Task with the Task Wizard

Use this process to create a new task on the My webMethods 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 29.

To create a new task with the Task wizard

1 Start the Task wizard as described in "Creating Tasks" on page 25.

> **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.

- 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 advanced options 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 want to work with the advanced options, click **Next**. Otherwise click **Finish** to create the task.

4 If you click **Next**, the Task Information page appears. Do the following:

a Type a task type name for the task. The task type name must be unique.

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.
- Type a unique task identifier.
- Click **Generate** to have Designer create one for you.

For more information about the GUID preference, see "Setting Task Development Preferences" on page 14; for more information about task type names, see "About Task Names" on page 23.

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:

Value	Description
Generate default Task View portlet (required)	Your task *must* contain a Task View portlet; otherwise, no task information will appear when the task is opened in My webMethods. If you do not select this check box now, you must add a Task View portlet after you create the task. For more information, see "Accessing the New Portlet Wizard" on page 46. This option can be enabled by default depending on the Task Development preference settings (see "Setting Task Development Preferences" on page 14).
Enable User Routing	Select this check box to provide the ability for a user to forward a task in the user's inbox to another user. This ability appears as a functional privilege that can be assigned to a user role. If you do not select this check box now, you can still add this functionality later by clicking the **Update** button in the Task Editor and selecting the appropriate wizard.
Auto Accept Task upon Modification	Select this check box to cause a task to be automatically accepted by a user when the user modifies the task. When this check box is not selected, the user must manually accept a task before updating it. You can change this setting later, as described in "Modifying Task Client Options" on page 113.
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 40. This option can be enabled by default depending on the Task Development preference settings (see "Setting Task Development Preferences" on page 14).
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 36 and "About the Task Inbox Results Portlet" on page 37. This option can be enabled by default depending on the Task Development preference settings (see "Setting Task Development Preferences" on page 14).
Accept Task on Open	Select this check box to cause a task to be automatically accepted by the first user to open the task.

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 rule sets. For more information on configuring a task, see "Configuring Tasks" on page 65.

Related Topics

- “Creating a New Task from an Integration Server Document Type” on page 28
- “Adding a Task to a Process” on page 29
- “About Task Names” on page 23
- “Configuring Tasks” on page 65
- “Task Overview” on page 20
- “Working with Task Portlets” on page 33
- “Working with Task Notifications” on page 134
- “Setting Task Development Preferences” on page 14

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 **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 26.
4. When the new task opens in the task editor, you can view the business data as described in “Viewing Business Data” on page 71.

For more information about working with business data, see “Configuring Business Data (Task Inputs and Outputs)” on page 65.

Related Topics

- “Adding a Task to a Process” on page 29
- “About Task Names” on page 23
- “Configuring Tasks” on page 65
- “Task Overview” on page 20
- “Working with Task Portlets” on page 33
- “Working with Task Notifications” on page 134
- “Setting Task Development Preferences” on page 14

Adding a Task to a Process

As you develop a process within webMethods 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 29.
- You can select an existing task and add it to the process, as described in "Adding an Existing Task to a Process" on page 31.

If you want to create a new task on the My webMethods perspective, see "Creating a New Task with the Task Wizard" on page 26.

Related Topics

- "Creating a New Task from an Integration Server Document Type" on page 28
- "Configuring Tasks" on page 65
- "Task Overview" on page 20
- "Working with Task Portlets" on page 33
- "Creating Tasks" on page 25

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 31.

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, it 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 dialog boxes will appear. For more information about this preference, see "Setting Task Development Preferences" on page 14.

  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 14. 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 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 26.

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 33.

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 rule sets. For more information on configuring a task, see "Configuring Tasks" on page 65.

Related Topics

- "Adding an Existing Task to a Process" on page 31
- "Creating a New Task with the Task Wizard" on page 26
- "Adding a Task to a Process" on page 29
- "About Task Names" on page 23
- "Configuring Tasks" on page 65
- "Task Overview" on page 20
- "Working with Task Notifications" on page 134

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 31.
- 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 32.

Related Topics

- "Creating a New Task from a Task Step in a Process" on page 29
- "Adding a Task to a Process" on page 29

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. Open the process you want to work with; for example, locate the process in the Solutions view and double click it to open it in the process editor,
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 Design 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.

Related Topics

- "Adding a Task by Selecting from a Task List" on page 32
- "Adding a Task to a Process" on page 29
- "Creating Tasks" on page 25
- "Configuring Tasks" on page 65
- "Task Overview" on page 20
- "Working with Task Portlets" on page 33

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 a 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 rule sets are available for the task, click the one you want to use in the **Task Rule Set** box.
8. Save the process.

Related Topics

- "Adding a Task by Drag and Drop" on page 31
- "Adding a Task to a Process" on page 29
- "Creating Tasks" on page 25
- "Adding a Task to a Process" on page 29
- "Configuring Tasks" on page 65
- "Task Overview" on page 20
- "Working with Task Portlets" on page 33

Working with Task Portlets

Because the contents of a task are essentially a specialized type of portlet, 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 53.
- 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" on page 144.
- 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 91.
- Bind a security (user) role to a control, as described in "Binding a Security Role to a User Interface Control" on page 63.
- Add your own Java methods.

For more information about general procedures and concepts for working with portlets, see "Creating a Portlet Application" and "Portlet Applications" in the *webMethods Composite Application Framework Help*.

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

Related Topics

- "Default Task Portlet Overview" on page 34
- "Viewing and Opening Task Portlets" on page 41
- "Modifying the Default Task Portlets" on page 42
- "Default Task Portlet Overview" on page 34
- "Running a Task Portlet in the Designer Preview Server" on page 42
- "Adding a New Portlet to a Task" on page 44
- "Accessing the New Portlet View Wizard" on page 53
- "Previewing a Portlet View" on page 55
- "Running a Task Portlet View in the Designer Preview Server" on page 56
- "Deleting Task Portlets and Portlet Views" on page 57
- "Working with Task Privileges" on page 86
- "Working with Security Roles" on page 59

Default Task Portlet Overview

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

When you create a task with the task wizard, a task view portlet is selected for creation by default (this behavior can be disabled in the Designer preferences). You can specify that other portlets are created by default as well; for more information, see "Setting Task Development Preferences" on page 14.

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.

If you do not choose to add any default portlets at creation time, you can add them later as described in "Adding a New Portlet to a Task" on page 44.

Related Topics

- "About the Task View Portlet" on page 35
- "About the Task Inbox Bar Portlet" on page 36
- "About the Task Inbox Results Portlet" on page 37
- "About the Task Start Portlet" on page 40
- "Working with Task Portlets" on page 33
- "Viewing and Opening Task Portlets" on page 41
- "Modifying the Default Task Portlets" on page 42
- "Running a Task Portlet in the Designer Preview Server" on page 42
- "Accessing the New Portlet Wizard" on page 46
- "Accessing the New Portlet View Wizard" on page 53
- "Previewing a Portlet View" on page 55
- "Running a Task Portlet View in the Designer Preview Server" on page 56
- "Deleting Task Portlets and Portlet Views" on page 57
- "Working with Task Privileges" on page 86
- "Working with Security Roles" on page 59

About the Task View Portlet

The Task View portlet contains the components that are displayed on the task's **Data View** tab in My webMethods. The **Data View** tab appears when the run-time user opens the task for the first time. Every task must contain a task view portlet; if one does not exist, nothing will appear when the user opens the task. When you create a task with the task wizard, a task view portlet is selected for creation by default (this behavior can be disabled in the Designer preferences).

If you choose not to create a task view portlet when the task is created, you must add a task view portlet to the task manually as described in "Adding a New Portlet to a Task" on page 44. You might want to do this if you intend to build your own task view starting with an empty design canvas.

The default task view portlet contains the following components:

- A **Task Business Data** panel that contains:
  - A formatted message panel for displaying any JSF context messages.
  - An empty property subgroup for you to add controls to.
  - A **Toggle Task Info** button that enables the user to view or hide the following task information:
    - **Name**—The name of the task as typed by the task creator.
    - **Description**—The description of the task as typed by the task creator.
    - **Created on/by**—The date and time the task was created and the name of the user who created it.
    - **Last modified on/by**—The date and time the task was last modified and the name of the user who modified it.
    - **Expires on**—The expiration date of the task as typed by the task creator.
    - **Status**—The current status of the task.
  - A Select Principal(s) dialog box to enable the user to specify roles and users to which the task can be assigned

The following buttons are also included:

- **Complete**—Updates all task information from the data fields on the page and applies a status of Completed to the task. No further work can be performed on the task.
- **Submit**—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).
- **Assign to Users** —Opens the Select Principal(s) dialog box.
- **Accept**—Accepts the task for the current user, enabling a user to work on the task. This button is visible only if the task is not marked for auto-acceptance and is not yet accepted.

- **Release**—Releases an accepted task, indicating it is no longer accepted by the current user and enabling it to be accepted by another user. This button is visible only if the task is not marked for auto-acceptance and is accepted.
- **Return**—Returns the user to the previous page.

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

Related Topics

- "About the Task Inbox Bar Portlet" on page 36
- "About the Task Inbox Results Portlet" on page 37
- "About the Task Start Portlet" on page 40
- "Default Task Portlet Overview" on page 34

About the Task Inbox Bar Portlet

The Task Inbox Bar portlet defines the user interface for the **Basic** tab portion of 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.
- 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**—Search by task identification number.
  - **Priority**—Search by task priority.
  - **Accepted Only**—Limits the search to tasks accepted by the user.
- A **Date Ranges** panel that contains the following controls:
  - **Created**—Provides a list of time intervals (last 24 hours, previous week, etc.).
  - **Expiration**—Provides a list of time intervals (last 24 hours, previous week, etc.).
  - **Last Updated**—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 143.

Related Topics

- "About the Task View Portlet" on page 35
- "About the Task Inbox Results Portlet" on page 37
- "About the Task Start Portlet" on page 40
- "Default Task Portlet Overview" on page 34

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.
- The *edit view* defines the formatting controls available to the user for the inbox results.

  The end user accesses the edit view by clicking the  Edit icon in the Inbox Results toolbar.

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 143.

Related Topics

- "About the Task View Portlet" on page 35
- "About the Task Inbox Bar Portlet" on page 36
- "About the Task Start Portlet" on page 40
- "Default Task Portlet Overview" on page 34

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 or roles that the task is assigned to.
- **Accepted By**—Name of user who last accepted the task.
- The following links are included
  - **Subscribe**—takes the user to the Task Subscriptions page.
  - **Scheduled Delegations**—Takes the user to the Tasks Scheduled Delegations page.

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 *webMethods Task Engine User's Guide*.

Related Topics

- "About the Inbox Results Edit View" on page 39
- "About the Task View Portlet" on page 35

- "About the Task Inbox Bar Portlet" on page 36
- "About the Inbox Results Default View" on page 37
- "About the Task Start Portlet" on page 40
- "Default Task Portlet Overview" on page 34

About the Inbox Results Edit View

You access the inbox results edit view in My webMethods by clicking the  Edit icon in the Inbox Results toolbar. 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.

Related Topics

- "About the Inbox Results Default View" on page 37
- "About the Task View Portlet" on page 35
- "About the Task Inbox Bar Portlet" on page 36
- "About the Inbox Results Default View" on page 37
- "About the Task Start Portlet" on page 40
- "Default Task Portlet Overview" on page 34

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 the *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 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.

Related Topics

- "About the Task View Portlet" on page 35
- "About the Task Inbox Bar Portlet" on page 36
- "About the Task Inbox Results Portlet" on page 37
- "Default Task Portlet Overview" on page 34

Viewing and Opening Task Portlets

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.

Related Topics

- "Working with Task Portlets" on page 33
- "Default Task Portlet Overview" on page 34
- "Modifying the Default Task Portlets" on page 42
- "Running a Task Portlet in the Designer Preview Server" on page 42
- "Accessing the New Portlet Wizard" on page 46
- "Accessing the New Portlet View Wizard" on page 53
- "Previewing a Portlet View" on page 55
- "Running a Task Portlet View in the Designer Preview Server" on page 56
- "Deleting Task Portlets and Portlet Views" on page 57
- "Working with Task Privileges" on page 86
- "Working with Security Roles" on page 59

Modifying the Default Task Portlets

You can modify the default task portlets just as you would modify any other portlet created in webMethods Designer. You can add or remove controls, change the layout, add functionality, and so on. For more information about working with portlets in general, see these topics in the CAF Development online help:

- "Getting Started with Portlet Application Development"—this topic provides links to topics about working with portlets in general.
- "Views in webMethods CAF"—this topic provides links to topics describing how to carry out specific portal development tasks, such as:
  - "Using the Portlet Application Configuration Editor"
  - "Using the Design Canvas"

Related Topics

- "About the Task View Portlet" on page 35
- "About the Task Inbox Bar Portlet" on page 36
- "About the Task Inbox Results Portlet" on page 37
- "About the Task Start Portlet" on page 40
- "Working with Task Portlets" on page 33
- "Default Task Portlet Overview" on page 34
- "Viewing and Opening Task Portlets" on page 41
- "Running a Task Portlet in the Designer Preview Server" on page 42
- "Accessing the New Portlet Wizard" on page 46
- "Accessing the New Portlet View Wizard" on page 53
- "Previewing a Portlet View" on page 55
- "Running a Task Portlet View in the Designer Preview Server" on page 56
- "Deleting Task Portlets and Portlet Views" on page 57
- "Working with Task Privileges" on page 86
- "Working with Security Roles" on page 59

Running a Task Portlet in the Designer Preview Server

You can run a portlet in the webMethods Designer preview server at any time, enabling you to test the functionality of the portlet before it is published to the run-time environment. When you run a portlet in the Designer preview server, you see all of the subgroups, fields, controls, and other components of the view as they will appear in the run-time environment.

All buttons, controls, and programmed functions of the portlet behave as they will in the My webMethods Server run-time environment.

To run a task portlet in the preview 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. In the Console view, progress messages appear as the portlet is started.
6. The portlet appears in its own tab in the editor area, and you can interact with the controls in the interface.
7. 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. The modified portlet is automatically republished to the preview server.

Related Topics

- "Working with Task Portlets" on page 33
- "Viewing and Opening Task Portlets" on page 41
- "Modifying the Default Task Portlets" on page 42
- "Default Task Portlet Overview" on page 34
- "Accessing the New Portlet Wizard" on page 46
- "Accessing the New Portlet View Wizard" on page 53
- "Previewing a Portlet View" on page 55
- "Running a Task Portlet View in the Designer Preview Server" on page 56
- "Deleting Task Portlets and Portlet Views" on page 57
- "Working with Task Privileges" on page 86
- "Working with Security Roles" on page 59

Adding a New Portlet to a Task

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

- When you create a task, you can specify the creation of several default portlets within the task, such as TaskView, TaskInboxBar, and TaskInboxResults. If you did not add the default portlets at creation time, you can add any of these default portlets after you create the task.
- After you create the task, you may want to replace the default portlet with a customized portlet.

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 "Creating a Portlet Application" in the *webMethods Composite Application Framework Help*. There are certain specific procedures that pertain to portlets within a task. For a description of the available task portlet templates, see "Working with Task Portlet Templates" on page 46.

To add a new portlet to a task

1. Open the portlet wizard as described in "Accessing the New Portlet Wizard" on page 46.
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 template in the **Portlet Type** box.

   Note: Other portlets appear in the **Portlet Type** box, but only Task View, Task Start, Task Inbox Bar, and Task Inbox Results portlets can be added to a task.

   Important! If you create a new Task View or Task Start portlet in a task that contains an existing Task View or Task Start portlet, the new portlet will automatically replace the existing one. For information about reverting to the previous portlet, see "Modifying the Task View Definition" on page 48.
4. When a task portlet template is selected, additional controls appear on the page:
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 (does not apply to all templates):

Value	Description
Use AJAX Controls	Select this check box to enable the use of AJAX (Asynchronous Java Script and XML) controls in the portlet.
Enable User Routing	Select this check box to provide the ability for a user to assign a task in the user's inbox to another user. This ability appears as a functional privilege that can be provided to a user 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.
Auto Accept Task upon Modification	Select this check box to cause a task to be automatically accepted by a user when the user updates the task. When this check box is not selected, the user must manually accept a task before updating it. You can modify this setting later, as described in "Modifying Task Client Options" on page 113.

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.

7 Click **Finish** or click **Next** to continue configuration.

8 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**.

9 On the Create Portlet page, specify any text you want to appear at deployment time. Click **Next**.

10 On the Create Portlet page, specify the portlet class values you want to apply, or accept the default values. Click **Next**.

11 On the Create Portlet page, specify any resource bundle values you want to apply, or accept the default values.

12 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.

Related Topics

- "Accessing the New Portlet Wizard" on page 46
- "Working with Task Portlet Templates" on page 46
- "Modifying the Task View Definition" on page 48
- "Modifying the Task Details Page Definition" on page 49
- "Modifying the Task Start Definition" on page 50
- "Working with Task Privileges" on page 86

- "Working with Security Roles" on page 59
- "Modifying Task Client Options" on page 113
- "Accessing the New Portlet View Wizard" on page 53

Accessing the New Portlet Wizard

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

- On the My webMethods 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 44.

Related Topics

- "Working with Task Portlet Templates" on page 46
- "Modifying the Task View Definition" on page 48
- "Modifying the Task Details Page Definition" on page 49
- "Modifying the Task Start Definition" on page 50
- "Working with Task Portlets" on page 33
- "Default Task Portlet Overview" on page 34

Working with Task Portlet Templates

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

⚠ **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. For more information, see "Modifying the Task View Definition" on page 48.

Template Name	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.
Task View (Default)	Use this template to add a default task view portlet. This template is equivalent to the default view portlet that is added to the task when the task is created (if you select the **Generate Default Task View Portlet** option at creation time).
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).
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).
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).

Related Topics

- "Adding a New Portlet to a Task" on page 44
- "Modifying the Task View Definition" on page 48
- "Modifying the Task Details Page Definition" on page 49
- "Modifying the Task Start Definition" on page 50
- "Accessing the New Portlet View Wizard" on page 53
- "Accessing the New Portlet Wizard" on page 46

Modifying the Task View Definition

If you add a new task view portlet in a task that contains an existing task view portlet, the new portlet will automatically replace the existing one. However, you must modify the task view definition if you want to revert to a previous task view portlet contained in the task. For information about modifying the task start portlet, see "Modifying the Task Start Definition" on page 50.

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
   ```

   webMethods Designer will always use the first view portlet in the definition.
5. Save your modifications.

You can view the results on the local preview server; for more information, see "Running a Task Portlet in the Designer Preview Server" on page 42. 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 138.

Related Topics

- "Modifying the Task Details Page Definition" on page 49
- "Modifying the Task Start Definition" on page 50
- "Adding a New Portlet to a Task" on page 44
- "Working with Task Portlet Templates" on page 46
- "Accessing the New Portlet View Wizard" on page 53
- "Modifying Task Client Options" on page 113

Modifying the Task Details Page Definition

By default, each new task is created with a task details page portlet that:

- Implements the Task Information display.
- Implements the **Task Audit** tab.
- Embeds a custom task details portlet in the **Task Data** tab.

This portlet uses the configuration from the taskDefinition.xml file to locate the task details portlet to use for Task Data tab. If you want to create and specify a custom task details portlet, you must modify the taskDetails.xml file.

Note: When a custom task details portlet is defined for this page, it will not implement the **Task Data** and **Task Audit** tabs that are included in the default portlet.

To modify the task details 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 taskDetails.xml file as follows:

   [*taskApplication*] > WebContent > WEB-INF > tasks > [*taskTypeID*] > taskDetails.xml
4. Double-click the taskDetails.xml file to open it in the editor and click the **Source** tab.
5. This XML file defines the task details portlet usage in the following elements (the portlet used is an "out of the box" portlet shipped with Designer):

   ```
   <wm_task_search___taskviews name="Task View">
   ...
   </wm_task_search___taskviews>
   ```

   This entry follows the format of:

   ```
   <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 to be used
   Display Name = the name that will be displayed in the design-time and run-time interfaces.
6. To redefine the task details portlet, do the following:
   - Substitute the new portlet project and name for the existing names.
   - Add or modify the portlet alias if desired.
   - Modify the display name for the portlet if desired.

For example:

```
<myportletapplication___mynewtaskdetails name="Task Details"
     alias="taskTypeID.task.app.newdetails.portlet">
          <layout name="titlebar" value="false"/>
          <layout name="border" value="false"/>
 </myportletapplication___mynewtaskdetails>
```

7 Save your modifications.

You can view the results on the local preview server; for more information, see "Running a Task Portlet in the Designer Preview Server" on page 42. 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 138.

Related Topics

- "Modifying the Task View Definition" on page 48
- "Modifying the Task Start Definition" on page 50
- "Adding a New Portlet to a Task" on page 44
- "Working with Task Portlet Templates" on page 46
- "Accessing the New Portlet View Wizard" on page 53
- "Modifying Task Client Options" on page 113

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 local preview server; for more information, see "Running a Task Portlet in the Designer Preview Server" on page 42. 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 138.

Related Topics

- "Modifying the Task View Definition" on page 48
- "Modifying the Task Details Page Definition" on page 49
- "Adding a New Portlet to a Task" on page 44
- "Working with Task Portlet Templates" on page 46
- "Accessing the New Portlet View Wizard" on page 53
- "Modifying Task Client Options" on page 113

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.

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 53:

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`.

   From this point on, you can click **Next** to specify custom settings for the view, or click **Finish** to create the view with remaining default settings.

4 Click **Finish** or click **Next** to continue configuration.

5 On the second page, specify the following:

Value	Description
For Portlet	This value is provided automatically as the name of the selected task portlet. If you have selected the wrong portlet, you can click the down arrow button to select from a list of all portlets in the task.
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**.

Value	Description
MIME Type	Enables you to specify the MIME type for View, Edit, and Help views.
Template	Enables you to select from a list of view templates. 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 "Working with Portlet View Templates" on page 54.

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 Portlet View" on page 55
- "Running a Task Portlet View in the Designer Preview Server" on page 56

Related Topics

- "Accessing the New Portlet View Wizard" on page 53
- "Working with Portlet View Templates" on page 54
- "Modifying Task Client Options" on page 113
- "Adding a New Portlet to a Task" on page 44
- "Deleting Task Portlets and Portlet Views" on page 57

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:

- On the My webMethods 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**.

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 52.

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 145.

Related Topics

- "Adding a New View to an Existing Task Portlet" on page 52
- "Working with Portlet View Templates" on page 54
- "Modifying Task Client Options" on page 113
- "Accessing the New Portlet Wizard" on page 46

Working with Portlet View Templates

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

⚠ **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. For more information, see "Modifying the Task View Definition" on page 48.

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.

Related Topics

- "Adding a New View to an Existing Task Portlet" on page 52
- "Accessing the New Portlet View Wizard" on page 53

- "Modifying Task Client Options" on page 113
- "Accessing the New Portlet Wizard" on page 46

Previewing a Portlet View

You can preview a portlet view in webMethods Designer 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 theMy webMethods Server Support option installed in Designer to use the preview server. If you have not already done so, shut down all webMethods applications, run the webMethods Installer, and select **Designer 7.2 > Composite Applications > My webMethods Server Support** to install the preview server functionality.

To preview a task portlet view

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.

Related Topics

- "Working with Task Portlets" on page 33
- "Viewing and Opening Task Portlets" on page 41

- "Modifying the Default Task Portlets" on page 42
- "Default Task Portlet Overview" on page 34
- "Running a Task Portlet in the Designer Preview Server" on page 42
- "Accessing the New Portlet Wizard" on page 46
- "Accessing the New Portlet View Wizard" on page 53
- "Running a Task Portlet View in the Designer Preview Server" on page 56
- "Deleting Task Portlets and Portlet Views" on page 57
- "Working with Task Privileges" on page 86
- "Working with Security Roles" on page 59

Running a Task Portlet View in the Designer Preview Server

You can run a portlet view in the webMethods Designer preview server at any time, enabling you to test the functionality of the view before it is published to the run-time environment. When you run a portlet view in the Designer preview server, you see all of the subgroups, fields, controls, and other components of the view as they will appear in the run-time environment.

All buttons, controls, and programmed functions of the portlet behave as they will in the My webMethods Server run-time environment.

To run a task portlet view in the preview 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. In the Console view, progress messages appear as the portlet is started.
6. The portlet appears in its own tab in the editor area, and you can interact with the controls in the interface.
7. 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 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.

Related Topics

- "Working with Task Portlets" on page 33
- "Viewing and Opening Task Portlets" on page 41
- "Modifying the Default Task Portlets" on page 42
- "Default Task Portlet Overview" on page 34
- "Running a Task Portlet in the Designer Preview Server" on page 42
- "Accessing the New Portlet Wizard" on page 46
- "Accessing the New Portlet View Wizard" on page 53
- "Previewing a Portlet View" on page 55
- "Deleting Task Portlets and Portlet Views" on page 57
- "Working with Task Privileges" on page 86
- "Working with Security Roles" on page 59

Deleting Task Portlets and Portlet Views

Task portlets and portlet views can be deleted in webMethods 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.

Related Topics

- “Working with Task Portlets” on page 33
- “Viewing and Opening Task Portlets” on page 41
- “Default Task Portlet Overview” on page 34
- “Running a Task Portlet in the Designer Preview Server” on page 42
- “Accessing the New Portlet Wizard” on page 46
- “Accessing the New Portlet View Wizard” on page 53
- “Previewing a Portlet View” on page 55
- “Running a Task Portlet View in the Designer Preview Server” on page 56
- “Working with Task Privileges” on page 86
- “Working with Security Roles” on page 59

Working with Security Roles

Security roles have two separate functions in webMethods 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.

Refer to the following related topics to view the available security roles for a task, or to add, delete, or modify a security role.

Related Topics

- "Viewing Task Security Roles" on page 59
- "Adding a Security Role to a Task" on page 60
- "Binding a Security Role to a User Interface Control" on page 63
- "Modifying and Deleting a Task Security Role" on page 64
- "Working with Task Privileges" on page 86
- "Working with Task Portlets" on page 33

Viewing Task Security Roles

To view task security roles

1. Open the portlet editor as described in "Viewing and Opening Task Portlets" on page 41.
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 60.

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 *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 91.

Related Topics

- "Adding a Security Role to a Task" on page 60
- "Binding a Security Role to a User Interface Control" on page 63
- "Modifying and Deleting a Task Security Role" on page 64
- "Working with Task Privileges" on page 86
- "Working with Security Roles" on page 59
- "Accessing the New Portlet View Wizard" on page 53

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 62.
- To provide a means of assigning a task to a My webMethods user role, see "Adding a Task Assignment Security Role" on page 61.

Related Topics

- "Viewing Task Security Roles" on page 59
- "Binding a Security Role to a User Interface Control" on page 63
- "Modifying and Deleting a Task Security Role" on page 64
- "Working with Task Privileges" on page 86
- "Working with Security Roles" on page 59

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 59).

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 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.

Related Topics

- "Adding a Security Role for Binding a Task Privilege to a User Interface Control" on page 62
- "Adding a Security Role to a Task" on page 60

- "Modifying and Deleting a Task Security Role" on page 64
- "Working with Security Roles" on page 59

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 59.

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 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 63.

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.

Related Topics

- "Adding a Task Assignment Security Role" on page 61
- "Adding a Security Role to a Task" on page 60

- "Modifying and Deleting a Task Security Role" on page 64
- "Working with Security Roles" on page 59
- "Working with Task Privileges" on page 86

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 62.

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 the Hide Expert Properties  button in the Properties view toolbar to show or hide all available properties.
5. Click the 

Expression Binding button 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 webMethods user role during run time. For more information about working with user roles in My webMethods, see the *webMethods Task Engine User's Guide*.

Related Topics

- "Viewing Task Security Roles" on page 59
- "Adding a Security Role to a Task" on page 60
- "Modifying and Deleting a Task Security Role" on page 64
- "Working with Task Privileges" on page 86
- "Working with Security Roles" on page 59

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 59.

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.

Related Topics

- "Viewing Task Security Roles" on page 59
- "Adding a Security Role to a Task" on page 60
- "Binding a Security Role to a User Interface Control" on page 63
- "Working with Security Roles" on page 59
- "Working with Task Privileges" on page 86

Configuring Tasks

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 a Java class.
- Create task assignments that will assign a task to a specified user 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.
- Create task events that apply an action when a specified change occurs is applied to the task; for example a change in the status, priority, business data, or assignment.
- Create one or more task rule 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 rule set; when you add the task to a business process in Developer, you can specify which rule set you want to apply to that particular instance of the task.

Related Topics

- "Configuring Business Data (Task Inputs and Outputs)" on page 65
- "Working with Services in Tasks" on page 80
- "Working with Task Privileges" on page 86
- "Working with Task Assignments" on page 92
- "Working with Task Events" on page 102
- "Working with Task Rule Sets" on page 110
- "Modifying Task Client Options" on page 113
- "Working with Tasks in the Bindings View" on page 114
- "Changing the Task Type Name" on page 117
- "Working with Task Expressions" on page 118
- "Creating a Task Notification Event" on page 130
- "webMethods Task Development" on page 11

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, as a Java type, or as a custom field. You have the opportunity to specify the documents, Java types, and fields you want to use at any time after you create a task. You can add business data to a task in the following ways:

- By adding documents, Java types, and custom fields with the task editor—for more information, see "Adding Business Data to a Task in the Task Editor" on page 67.

- 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 28.
- By creating step inputs and outputs for a task activity step using the process editor—for more information, see "Creating Step Inputs and Outputs" in the *webMethods 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 (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 or Java type may contain task input elements for "name," "price," and "quantity." Similarly, a Java type or document may define output elements such as "accept" and "escalate." Or, you might include both input and output elements in a single document or Java type.

You can create Integration Server documents in the following ways:

- In webMethods Designer, using the Designer ESB and Integration Server feature. For more information, see "Creating a Document Type" in the *webMethods Service Development Help* (this feature must be installed to view this help topic).
- In webMethods Developer. For more information about creating documents with webMethods Developer, see the *webMethods Developer User's Guide*.

In both cases, you must be connected to your designated Integration Server

Related Topics

- "Adding Business Data to a Task in the Task Editor" on page 67
- "Adding an Integration Server Document as Business Data" on page 68
- "Adding a Java Class as Business Data" on page 69
- "Adding a Custom Field Type as Business Data" on page 70
- "Viewing Business Data" on page 71
- "Modifying and Deleting Task Business Data" on page 72
- "Updating or Refreshing Task Business Data" on page 73
- "Adding Task Business Data Fields to a View" on page 77
- "Dragging and Dropping Task Business Data Fields to a View" on page 78
- "Creating Custom Search Fields from Task Business Data" on page 79
- "Configuring Tasks" on page 65

Adding Business Data to a Task in the Task Editor

After you create a task you can add business data in the task editor in the form of a document, a Java type, or a custom field, defining task inputs and outputs. You can also add business data to a task in the following ways:

- By adding business data to a task activity step using the process editor—for more information, see "Creating Step Inputs and Outputs" in the *webMethods Process Development Help*.
- 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 28.

To add business data to a task in the task editor

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. Click the **Overview** tab. Task business data is displayed in the Business Data area in the upper-right area of the editor.
3. Click the **Add** button in the Business Data area. The Field Type Selection dialog box appears.
4. In the display list on the left, click the type of business data you want to add.

Related Topics

- "Adding an Integration Server Document as Business Data" on page 68
- "Adding a Java Class as Business Data" on page 69
- "Adding a Custom Field Type as Business Data" on page 70
- "Viewing Business Data" on page 71
- "Modifying and Deleting Task Business Data" on page 72
- "Updating or Refreshing Task Business Data" on page 73
- "Adding Task Business Data Fields to a View" on page 77
- "Dragging and Dropping Task Business Data Fields to a View" on page 78
- "Creating Custom Search Fields from Task Business Data" on page 79
- "Configuring Business Data (Task Inputs and Outputs)" on page 65

Adding an Integration Server Document as Business Data

To add an Integration Server document reference to a task as business data

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. Click the **Overview** tab. Task business data is displayed in the Business Data area in the upper-right area of the editor.
3. Click the **Add** button in the Business Data area. The Field Type Selection dialog box appears.
4. Select Document Reference as the field type.
5. Click **Browse** to specify the Integration Server document you want to add.
6. Type the display name you want to display for the business data.
7. Select the appropriate check box to specify the contents of the document:
   - If the field is an input of the task, select **Input**.
   - If the field is an output of the task, select **Output**.
   - If the field is a list of specified documents rather than a single document, select **List**.
8. Click **OK**. The selected document appears in the Business Data area.
9. Save the task.

Note: You can also add a document as business data by dragging it from the appropriate data browser (for example, the Package Navigator view, or the Metadata Library view). Drag the document and drop it onto the Business Data area. Then, with the newly added item selected, click **Edit** to specify the document properties.

You can now add the business data to a view as described in "Adding Task Business Data Fields to a View" on page 77 and "Dragging and Dropping Task Business Data Fields to a View" on page 78.

You can view the name and expression values for the document by clicking it in the Bindings view and then opening the Properties view.

Related Topics

- "Adding Business Data to a Task in the Task Editor" on page 67
- "Adding a Java Class as Business Data" on page 69
- "Adding a Custom Field Type as Business Data" on page 70
- "Viewing Business Data" on page 71
- "Modifying and Deleting Task Business Data" on page 72

- "Updating or Refreshing Task Business Data" on page 73
- "Adding Task Business Data Fields to a View" on page 77
- "Dragging and Dropping Task Business Data Fields to a View" on page 78
- "Creating Custom Search Fields from Task Business Data" on page 79
- "Configuring Business Data (Task Inputs and Outputs)" on page 65

Adding a Java Class as Business Data

Note: Custom Java type fields cannot be inputs or outputs of a task; these fields can only be used internally by the task. The selected Java types must implement the java.io.Serializable interface; otherwise, they will not be persisted as business data.

To add a Java class to a task as business data

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. Click the **Overview** tab. Task business data is displayed in the Business Data area in the upper-right area of the editor.
3. Click the **Add** button in the Business Data area. The Field Type Selection dialog box appears.
4. Select Java class as the field type.
5. Click **Browse** to select the Java class you want to add.
6. Type the display name you want to display for the business data.
7. Click **OK**. The selected Java class appears in the Business Data area.
8. Save the task.

You can now add the business data to a view as described in "Adding Task Business Data Fields to a View" on page 77 and "Dragging and Dropping Task Business Data Fields to a View" on page 78.

You can view the name and expression values for the Java type by clicking it in the Bindings view and then opening the Properties view.

Related Topics

- "Adding Business Data to a Task in the Task Editor" on page 67
- "Adding an Integration Server Document as Business Data" on page 68
- "Adding a Custom Field Type as Business Data" on page 70
- "Viewing Business Data" on page 71
- "Modifying and Deleting Task Business Data" on page 72

- "Updating or Refreshing Task Business Data" on page 73
- "Adding Task Business Data Fields to a View" on page 77
- "Dragging and Dropping Task Business Data Fields to a View" on page 78
- "Creating Custom Search Fields from Task Business Data" on page 79
- "Configuring Business Data (Task Inputs and Outputs)" on page 65

Adding a Custom Field Type as Business Data

To add a custom field to a task as business data

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. Click the **Overview** tab. Task business data is displayed in the Business Data area in the upper-right area of the editor.
3. Click the **Add** button in the Business Data area. The Field Type Selection dialog box appears.
4. Select the type of field you want to add, for example, string, Boolean, or float.
5. Type the display name you want to display for the business data.
6. Select the appropriate check box to specify how the field will be used:
   - If the field is to be an input element, select **Input**.
   - If the field is to be an output element, select **Output**.
   - If the field is to contain a list, select **List**.
7. Click **OK**. The selected field type appears in the Business Data area.
8. Save the task.

You can now add the business data to a view as described in "Adding Task Business Data Fields to a View" on page 77 and "Dragging and Dropping Task Business Data Fields to a View" on page 78.

You can view the name and expression values for the field by clicking it in the Bindings view and then opening the Properties view.

Related Topics

- "Adding Business Data to a Task in the Task Editor" on page 67
- "Adding an Integration Server Document as Business Data" on page 68
- "Adding a Java Class as Business Data" on page 69
- "Viewing Business Data" on page 71
- "Modifying and Deleting Task Business Data" on page 72

- "Updating or Refreshing Task Business Data" on page 73
- "Adding Task Business Data Fields to a View" on page 77
- "Dragging and Dropping Task Business Data Fields to a View" on page 78
- "Creating Custom Search Fields from Task Business Data" on page 79
- "Configuring Business Data (Task Inputs and Outputs)" on page 65

Viewing Business Data

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

- In the Business Data area of the **Overview** tab in the task editor, as described in "Adding Business Data to a Task in the Task Editor" on page 67.
- 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 task 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 from the pop-up menu, enabling you to delete an elements, as well as add data, or add an action, managed bean, or content provider, as appropriate.

Related Topics

- "Adding Business Data to a Task in the Task Editor" on page 67
- "Adding an Integration Server Document as Business Data" on page 68
- "Adding a Java Class as Business Data" on page 69
- "Adding a Custom Field Type as Business Data" on page 70
- "Modifying and Deleting Task Business Data" on page 72
- "Updating or Refreshing Task Business Data" on page 73
- "Adding Task Business Data Fields to a View" on page 77
- "Dragging and Dropping Task Business Data Fields to a View" on page 78
- "Creating Custom Search Fields from Task Business Data" on page 79
- "Configuring Business Data (Task Inputs and Outputs)" on page 65

Modifying and Deleting Task Business Data

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

To modify business data

1. In the Solutions view, open the task in the task editor by double-clicking the task name.
2. Click the **Overview** tab. Task business data is displayed in the Business Data area in the upper-right area of the editor.
3. Click the business data element you want to modify.
4. Click the **Edit** button in the Business Data area. The Field Type Selection dialog box appears with the current properties for the selected element displayed.
5. Modify the business data properties as required and click **OK**.
6. Save the task.

To delete business data

1. In the Business Data area of the task editor, click the business data element you want to delete.
2. Click the **Delete** button in the Business Data area.
3. Click **Yes** in the Delete confirmation dialog box.
4. Save the task.

Note: You can also add delete business data elements from the Bindings view. In the Bindings view, expand the Business Data entry and right-click the business data element your 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.

Related Topics

- "Adding Business Data to a Task in the Task Editor" on page 67
- "Adding an Integration Server Document as Business Data" on page 68
- "Adding a Java Class as Business Data" on page 69
- "Adding a Custom Field Type as Business Data" on page 70
- "Viewing Business Data" on page 71
- "Updating or Refreshing Task Business Data" on page 73
- "Adding Task Business Data Fields to a View" on page 77
- "Dragging and Dropping Task Business Data Fields to a View" on page 78
- "Creating Custom Search Fields from Task Business Data" on page 79
- "Configuring Business Data (Task Inputs and Outputs)" on page 65

Updating or Refreshing Task Business Data

As you work with a task, the task business data associated with the task can be changed from the task editor or from the process editor (for more information, see "Configuring Business Data (Task Inputs and Outputs)" on page 65. Similarly, an Integration Server document or local Java type that is associated with a task to provide business data 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 Task Editor" on page 74.
- To refresh the task business data from an Integration Server document, see "Refreshing Task Business Data from an Integration Server Document" on page 75
- 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 76.

Related Topics

- "Adding Business Data to a Task in the Task Editor" on page 67
- "Adding an Integration Server Document as Business Data" on page 68
- "Adding a Java Class as Business Data" on page 69
- "Adding a Custom Field Type as Business Data" on page 70
- "Viewing Business Data" on page 71
- "Modifying and Deleting Task Business Data" on page 72
- "Adding Task Business Data Fields to a View" on page 77
- "Dragging and Dropping Task Business Data Fields to a View" on page 78
- "Creating Custom Search Fields from Task Business Data" on page 79
- "Configuring Business Data (Task Inputs and Outputs)" on page 65

Refreshing Document Business Data in the Task 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 IS document or Java class.

Note: The **Refresh** button in the Task Editor is enabled only for Integration Server documents. It also regenerates any local Java types that are created from the source IS document. The **Refresh** button is not enabled for custom Java types; these are updated automatically when the Java source for such types is updated.

To refresh document business data

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

Related Topics

- "Updating or Refreshing Task Business Data" on page 73
- "Refreshing Task Business Data from an Integration Server Document" on page 75
- "Updating Business Data from a Task Step in a Process" on page 76
- "Adding Business Data to a Task in the Task Editor" on page 67
- "Adding an Integration Server Document as Business Data" on page 68
- "Adding a Java Class as Business Data" on page 69
- "Adding a Custom Field Type as Business Data" on page 70
- "Viewing Business Data" on page 71
- "Modifying and Deleting Task Business Data" on page 72
- "Adding Task Business Data Fields to a View" on page 77
- "Dragging and Dropping Task Business Data Fields to a View" on page 78
- "Creating Custom Search Fields from Task Business Data" on page 79
- "Configuring Business Data (Task Inputs and Outputs)" on page 65

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 for all tasks that use the IS document to the latest version of 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 **Sync Java Wrapper Types**.
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.

Related Topics

- "Updating or Refreshing Task Business Data" on page 73
- "Refreshing Document Business Data in the Task Editor" on page 74
- "Updating Business Data from a Task Step in a Process" on page 76
- "Adding Business Data to a Task in the Task Editor" on page 67
- "Adding an Integration Server Document as Business Data" on page 68
- "Adding a Java Class as Business Data" on page 69
- "Adding a Custom Field Type as Business Data" on page 70
- "Viewing Business Data" on page 71
- "Modifying and Deleting Task Business Data" on page 72

- "Updating or Refreshing Task Business Data" on page 73
- "Adding Task Business Data Fields to a View" on page 77
- "Dragging and Dropping Task Business Data Fields to a View" on page 78
- "Creating Custom Search Fields from Task Business Data" on page 79
- "Configuring Business Data (Task Inputs and Outputs)" on page 65

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 Task Editor" on page 74.

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 page 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 Task Design Editor**.

Related Topics

- "Updating or Refreshing Task Business Data" on page 73
- "Refreshing Document Business Data in the Task Editor" on page 74
- "Refreshing Task Business Data from an Integration Server Document" on page 75
- "Adding Business Data to a Task in the Task Editor" on page 67
- "Adding an Integration Server Document as Business Data" on page 68
- "Adding a Java Class as Business Data" on page 69
- "Adding a Custom Field Type as Business Data" on page 70

- "Viewing Business Data" on page 71
- "Modifying and Deleting Task Business Data" on page 72
- "Updating or Refreshing Task Business Data" on page 73
- "Adding Task Business Data Fields to a View" on page 77
- "Dragging and Dropping Task Business Data Fields to a View" on page 78
- "Creating Custom Search Fields from Task Business Data" on page 79
- "Configuring Business Data (Task Inputs and Outputs)" on page 65

Adding Task Business Data Fields to a View

You can add existing task business data fields to a task portlet view or a notification view using the task 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 View" on page 78.

To add business data fields to a task portlet view

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

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 to the task as described in "Adding Business Data to a Task in the Task Editor" on page 67.

3 On the **Overview** tab, expand the User Interfaces area if it is not already visible.

4 Click the task portlet view or notification view you want to add business data to; for example, the default view of the Task Details portlet.

Note: You cannot add business data to the Task Inbox Search Bar view or the Task Inbox Search Results edit view.

5 Click the **Update** button in the User Interfaces area. The Task UI Update dialog box 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 dialog box appears.

Note: This display is empty if there is no business data associated with the task, or if business data has been added but the task has not been saved.

8 Expand the table entries to locate and click the business data fields you want add.

9 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.

10 Click **Finish**. The business data fields are added to the view as follows:

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

11 The changes to the task are saved automatically.

Related Topics

- "Adding Business Data to a Task in the Task Editor" on page 67
- "Adding an Integration Server Document as Business Data" on page 68
- "Adding a Java Class as Business Data" on page 69
- "Adding a Custom Field Type as Business Data" on page 70
- "Viewing Business Data" on page 71
- "Modifying and Deleting Task Business Data" on page 72
- "Updating or Refreshing Task Business Data" on page 73
- "Dragging and Dropping Task Business Data Fields to a View" on page 78
- "Creating Custom Search Fields from Task Business Data" on page 79
- "Configuring Business Data (Task Inputs and Outputs)" on page 65

Dragging and Dropping Task Business Data Fields to a View

You can add existing task business data fields to a task portlet view by dragging and dropping, as described below. You can also add business data fields using the task editor as described in "Adding Task Business Data Fields to a View" on page 77.

To drag and drop business data fields to a task portlet view

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

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 to the task as described in "Adding Business Data to a Task in the Task Editor" on page 67.

3 On the **Overview** tab, expand the User Interfaces area if it is not already visible.

4 Double-click the task portlet 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.

Note: You can also open the task view editor from the Solutions view by expanding the task and the portlet containing the view, then by double clicking the view.

5 In the Bindings view, expand the **Managed Beans > [*TaskName*]View/default > [*Task Name*] > Business Data** entries to view the available business data fields.

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

7 Save the task.

Related Topics

- "Adding Business Data to a Task in the Task Editor" on page 67
- "Adding an Integration Server Document as Business Data" on page 68
- "Adding a Java Class as Business Data" on page 69
- "Adding a Custom Field Type as Business Data" on page 70
- "Viewing Business Data" on page 71
- "Modifying and Deleting Task Business Data" on page 72
- "Updating or Refreshing Task Business Data" on page 73
- "Adding Task Business Data Fields to a View" on page 77
- "Creating Custom Search Fields from Task Business Data" on page 79
- "Configuring Business Data (Task Inputs and Outputs)" on page 65

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.

To create a custom search field from a business data field

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

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 to the task as described in "Adding Business Data to a Task in the Task Editor" on page 67.

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 Bar portlet.

5 Click the **Update** button in the User Interfaces area. The Task UI Update dialog box appears with the current properties for the selected element displayed.

6 Click the **Implement task custom inbox bar search for task business data field(s)** option.

7 Click **Next**. The Add Searchable Fields dialog box appears.

Note: This display is empty if there is no business data associated with the task.

8 Click the business data field you want to work with to place it in the **Search Term Name** box.

9 Select the operator you want to apply to the field in the **Search Term Operator** box.

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 view in the view editor after applying the business data changes, select the **Open updated page in Editor** check box.

12 The changes to the task are saved automatically.

Related Topics

- "Adding Business Data to a Task in the Task Editor" on page 67
- "Adding an Integration Server Document as Business Data" on page 68
- "Adding a Java Class as Business Data" on page 69
- "Adding a Custom Field Type as Business Data" on page 70
- "Viewing Business Data" on page 71
- "Modifying and Deleting Task Business Data" on page 72
- "Updating or Refreshing Task Business Data" on page 73
- "Adding Task Business Data Fields to a View" on page 77
- "Dragging and Dropping Task Business Data Fields to a View" on page 78
- "Configuring Business Data (Task Inputs and Outputs)" on page 65

Working with Services in Tasks

webMethods 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 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 96.

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 82

"Modifying a Service in a Task" on page 84

"Extending a Task with Custom Java Code" on page 85

Related Topics

- "Configuring Simple Task Conditions" on page 96
- "Configuring Business Data (Task Inputs and Outputs)" on page 65
- "Working with Task Privileges" on page 86
- "Working with Task Assignments" on page 92
- "Working with Task Events" on page 102
- "Working with Task Rule Sets" on page 110
- "Modifying Task Client Options" on page 113
- "Working with Tasks in the Bindings View" on page 114
- "Changing the Task Type Name" on page 117
- "Working with Task Expressions" on page 118
- "Creating a Task Notification Event" on page 130
- "Configuring Tasks" on page 65

Adding a Service to a Task

You can add a service to a task by dragging and dropping an existing Integration Server service, or by using the web service connector wizard. For more information about invoking services, see "Working with Services in Tasks" on page 80.

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 the  Expression Binding button to select a binding attribute.

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

1. On the My webMethods 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 the  Expression Binding button to select a binding attribute.

Related Topics

- "Modifying a Service in a Task" on page 84
- "Configuring Simple Task Conditions" on page 96
- "Configuring Business Data (Task Inputs and Outputs)" on page 65
- "Working with Task Privileges" on page 86
- "Working with Task Assignments" on page 92
- "Working with Task Events" on page 102
- "Working with Task Rule Sets" on page 110
- "Modifying Task Client Options" on page 113
- "Working with Tasks in the Bindings View" on page 114
- "Changing the Task Type Name" on page 117
- "Working with Task Expressions" on page 118
- "Creating a Task Notification Event" on page 130
- "Configuring Tasks" on page 65

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 the Show Expert Properties  button 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 the 

Expression Binding button to select a binding expression.

4 Save your modifications.

Related Topics

- "Adding a Service to a Task" on page 82
- "Configuring Simple Task Conditions" on page 96
- "Configuring Business Data (Task Inputs and Outputs)" on page 65
- "Working with Task Privileges" on page 86
- "Working with Task Assignments" on page 92
- "Working with Task Events" on page 102
- "Working with Task Rule Sets" on page 110
- "Modifying Task Client Options" on page 113
- "Working with Tasks in the Bindings View" on page 114
- "Changing the Task Type Name" on page 117
- "Working with Task Expressions" on page 118
- "Creating a Task Notification Event" on page 130
- "Configuring Tasks" on page 65

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 85.
- 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 86.

Related Topics

- "Working with Services in Tasks" on page 80
- "Adding a Service to a Task" on page 82
- "Modifying a Service in a Task" on page 84

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 like `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 86, the Java code for those additions is visible here.

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

Related Topics

- "Extending a Task with Custom Java Code" on page 85
- "Adding Actions and Data to a Task in the Bindings View" on page 86

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 "Creating a New Method" in the *webMethods Composite Application Framework Help*.
- Click **Add>Data** to open the Add New Property wizard. For more information about working with this wizard, see "Adding a New Property to a Page Bean" in the *webMethods Composite Application Framework 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.

Related Topics

- "Extending a Task with Custom Java Code" on page 85
- "Adding Your Own Java Code" on page 85

Working with Task Privileges

When you create a task, webMethods 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 87
- "Viewing Task Functional Privileges" on page 88
- "Adding a Task Functional Privilege" on page 89
- "Adding a Task Functional Privilege Container" on page 88
- "Modifying and Deleting a Task Functional Privilege Component" on page 90
- "Binding a Task Functional Privilege to a User Interface Control" on page 91

Related Topics

- "Working with Security Roles" on page 59
- "Modifying Task Client Options" on page 113
- "Configuring Tasks" on page 65

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 *webMethods Task Engine User's Guide*.

Related Topics

- "Viewing Task Functional Privileges" on page 88
- "Adding a Task Functional Privilege Container" on page 88
- "Adding a Task Functional Privilege" on page 89
- "Modifying and Deleting a Task Functional Privilege Component" on page 90
- "Binding a Task Functional Privilege to a User Interface Control" on page 91
- "Working with Task Privileges" on page 86
- "Working with Security Roles" on page 59
- "Modifying Task Client Options" on page 113

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.

You can also view task privileges in My webMethods and assign them to or remove them from user roles. For more information, see the *webMethods Task Engine User's Guide.*

Related Topics

- "About Privilege Containers" on page 87
- "Adding a Task Functional Privilege Container" on page 88
- "Adding a Task Functional Privilege" on page 89
- "Modifying and Deleting a Task Functional Privilege Component" on page 90
- "Binding a Task Functional Privilege to a User Interface Control" on page 91
- "Working with Task Privileges" on page 86
- "Working with Security Roles" on page 59
- "Modifying Task Client Options" on page 113

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 87

To add a task functional privilege container

1 Navigate to the functional privileges editor as described in "Viewing Task Functional Privileges" on page 88.

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.

Related Topics

- "About Privilege Containers" on page 87
- "Viewing Task Functional Privileges" on page 88

- "Adding a Task Functional Privilege" on page 89
- "Modifying and Deleting a Task Functional Privilege Component" on page 90
- "Binding a Task Functional Privilege to a User Interface Control" on page 91
- "Working with Task Privileges" on page 86
- "Working with Security Roles" on page 59
- "Modifying Task Client Options" on page 113

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 88.

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 87.

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.

4 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 91.

Related Topics

- "About Privilege Containers" on page 87
- "Viewing Task Functional Privileges" on page 88
- "Adding a Task Functional Privilege Container" on page 88
- "Modifying and Deleting a Task Functional Privilege Component" on page 90
- "Binding a Task Functional Privilege to a User Interface Control" on page 91
- "Working with Task Privileges" on page 86
- "Working with Security Roles" on page 59
- "Modifying Task Client Options" on page 113

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

1 Navigate to the functional privileges editor as described in "Viewing Task Functional Privileges" on page 88.

2 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**.

3 Publish the modified task application project to the run-time environment.

Related Topics

- "About Privilege Containers" on page 87
- "Viewing Task Functional Privileges" on page 88

- "Adding a Task Functional Privilege Container" on page 88
- "Adding a Task Functional Privilege" on page 89
- "Binding a Task Functional Privilege to a User Interface Control" on page 91
- "Working with Task Privileges" on page 86
- "Working with Security Roles" on page 59
- "Modifying Task Client Options" on page 113

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 89).
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 62). 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 63.

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 *webMethods Task Engine User's Guide*.

Related Topics

- "About Privilege Containers" on page 87
- "Viewing Task Functional Privileges" on page 88
- "Adding a Task Functional Privilege Container" on page 88
- "Adding a Task Functional Privilege" on page 89
- "Modifying and Deleting a Task Functional Privilege Component" on page 90
- "Working with Task Privileges" on page 86
- "Working with Security Roles" on page 59
- "Modifying Task Client Options" on page 113

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 or roles you want the task assigned to, and any conditions that you want to apply to the assignment.

You can create two or more task assignments, and place them in different rule 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 rule sets, see "Working with Task Rule Sets" on page 110.

Related Topics

- "Creating a Task Assignment" on page 92
- "Configuring Assignment General Information" on page 93
- "Configuring Assignees for a Task Assignment" on page 94
- "Configuring Task Assignment Options and Conditions" on page 95
- "Configuring Simple Task Conditions" on page 96
- "Configuring Advanced Task Expressions" on page 120
- "Configuring Distribution Management for a Task Assignment" on page 99
- "Working with Task Assignments" on page 92
- "Configuring Tasks" on page 65

Creating a Task Assignment

After you create a task you can create one or more task assignments.

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.
4. Save the task.

> **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
- Assignees

- Option and condition settings
- Distribution management settings

Related Topics

- "Configuring Assignment General Information" on page 93
- "Configuring Assignees for a Task Assignment" on page 94
- "Configuring Task Assignment Options and Conditions" on page 95
- "Configuring Simple Task Conditions" on page 96
- "Configuring Advanced Task Expressions" on page 120
- "Configuring Distribution Management for a Task Assignment" on page 99
- "Working with Task Assignments" on page 92

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 Description** — Type a description of the assignment. This text is also displayed in My webMethods.
   - **Rule Set** — Type a new rule set name, or select an existing rule set name from the drop-down list. For more information on rule sets, see "Working with Task Rule Sets" on page 110.

Related Topics

- "Creating a Task Assignment" on page 92
- "Configuring Assignees for a Task Assignment" on page 94
- "Configuring Task Assignment Options and Conditions" on page 95
- "Configuring Simple Task Conditions" on page 96
- "Configuring Advanced Task Expressions" on page 120

- "Configuring Distribution Management for a Task Assignment" on page 99
- "Working with Task Assignments" on page 92

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, it will be assigned to all assignees listed here. You can create two or more task assignments, and place them in different rule sets; this enables you to select different assignee lists for different task implementations in the run-time environment. For more information about rule sets, see "Working with Task Rule Sets" on page 110.

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 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 from the task business data or a task info field.
   - **Security Role** — Click or type the name of an available security role.

     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.

   - **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 click a service (these services must be previously defined in (added to) the Bindings view).

6 Click **OK**.

7 Repeat steps 4-6 to add additional assignees.

8 Save the task.

Related Topics

- "Creating a Task Assignment" on page 92
- "Configuring Assignment General Information" on page 93
- "Configuring Task Assignment Options and Conditions" on page 95
- "Configuring Simple Task Conditions" on page 96
- "Configuring Advanced Task Expressions" on page 120
- "Configuring Distribution Management for a Task Assignment" on page 99
- "Working with Task Assignments" on page 92

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 rule 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 rule sets, see "Working with Task Rule Sets" on page 110.

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 99.

To configure options and conditions for a task

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 102.

- **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 96
  - "Configuring Advanced Task Expressions" on page 120

5 Save the task to apply your changes.

Related Topics

- "Creating a Task Assignment" on page 92
- "Configuring Assignment General Information" on page 93
- "Configuring Assignees for a Task Assignment" on page 94
- "Configuring Distribution Management for a Task Assignment" on page 99
- "Working with Task Assignments" on page 92

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 95 and "Configuring a Task Event Type" on page 105.

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 120.

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.

Related Topics

- "Modifying and Deleting Simple Task Conditions" on page 98
- "Rearranging the Order of Simple Task Conditions" on page 98
- "Configuring Advanced Task Expressions" on page 120
- "Creating a Task Assignment" on page 92
- "Configuring Assignment General Information" on page 93
- "Configuring Assignees for a Task Assignment" on page 94
- "Configuring Task Assignment Options and Conditions" on page 95
- "Configuring Advanced Task Expressions" on page 120
- "Configuring Distribution Management for a Task Assignment" on page 99
- "Working with Task Assignments" on page 92

Modifying and Deleting 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 96.

5 Save the task to apply your changes.

To delete a simple task assignment condition

1 In the **Conditions** box, select the simple condition you want to delete.

2 Click **Delete**. The condition is removed from the task.

3 Save the task to apply your changes.

Related Topics

- "Rearranging the Order of Simple Task Conditions" on page 98
- "Configuring Simple Task Conditions" on page 96

Rearranging the Order of Simple Task Conditions

You can rearrange the order of the simple conditions that govern task assignments and events by clicking a condition in the **Conditions** list, and clicking the **Up** or **Down** button. Although this functionality is available for all join options (All, Any, or One), it has significance only for the Any join option.

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. Therefore, to speed up performance, for example, you can 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 service at the bottom of the condition list.

After reordering the conditions, be sure to save the task to apply your changes.

Related Topics

- "Modifying and Deleting Simple Task Conditions" on page 98
- "Configuring Simple Task Conditions" on page 96

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:

   Note: If you select round-robin or queue length distribution, the assignee list must contain *only* My webMethods Server roles, security roles, or services that return roles. 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

   - **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** — With this setting, tasks are assigned sequentially among the available users who are members of the roles in the assignee list.
     - 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 100. For more information about user availability by calendar, see "Working with User and Business Calendars" on page 146 and "About User Calendars and Task Distribution Management" on page 101.
   - **Queue length** — Use this entry to assign the task to the role-based assignee with the shortest queue length (that is, with the fewest assigned tasks).
     - Click the **Use User's Default Inbox** option to determine the user's queue length directly from the contents of each user's inbox
     - 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 "Saved Searches View" in the *webMethods Process Development Help*.

- 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 101. For more information about user availability by calendar, see "Working with User and Business Calendars" on page 146 and "About User Calendars and Task Distribution Management" on page 101.

5 Save the task to apply your changes.

Related Topics

- "Creating a Task Assignment" on page 92
- "Configuring Assignment General Information" on page 93
- "Configuring Assignees for a Task Assignment" on page 94
- "Configuring Task Assignment Options and Conditions" on page 95
- "Configuring Simple Task Conditions" on page 96
- "Configuring Advanced Task Expressions" on page 120
- "Working with Task Assignments" on page 92

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 99.

Note: For round-robin, the assignee list must contain *only* My webMethods Server roles, security roles, or services that return roles. Individual user names cannot be in the assignee list for the round-robin distribution management methods

Round-robin distribution considers both roles and users 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 contains another (nested) role, the nested role 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 both one or more security roles that are linked to My webMethods Server roles, those security roles are handled exactly like other roles.

Related Topics

- "About Queue Length Task Distribution" on page 101
- "About User Calendars and Task Distribution Management" on page 101
- "Configuring Distribution Management for a Task Assignment" on page 99

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-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 99.

Note: For queue length, the assignee list must contain *only* My webMethods Server roles, security roles, or services that return roles. 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).

Related Topics

- "About Round-Robin Task Distribution" on page 100
- "About User Calendars and Task Distribution Management" on page 101
- "Configuring Distribution Management for a Task Assignment" on page 99

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, 9 A.M.	Friday
Friday, 9:01 A.M.	Monday (the Friday business day has already started, so the next business day is now Monday)

All tasks queued between 9:01 A.M. Friday and 9:00 A.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 *webMethods Task Engine User's Guide*.

Related Topics

- "About Round-Robin Task Distribution" on page 100
- "About Queue Length Task Distribution" on page 101
- "Configuring Distribution Management for a Task Assignment" on page 99

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 rule sets; this enables you to select different events and conditional behavior for different task implementations in the run-time environment. For more information about rule sets, see "Working with Task Rule Sets" on page 110.

Related Topics

- "Creating a Task Event" on page 103
- "Configuring Event General Information" on page 104
- "Configuring a Task Event Type" on page 105
- "Configuring a Task Event Action" on page 106
- "Configuring Simple Task Actions" on page 107
- "Configuring a Date/Time Event" on page 108
- "Adding a Service to a Task" on page 82
- "Configuring Tasks" on page 65

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. Save the task.

You can now configure the following event attributes:

- General information about the event
- Event type
- Condition settings and options (when applicable)
- Event action (event actions are not required for filtering event types)

Related Topics

- “Configuring Event General Information” on page 104
- “Configuring a Task Event Type” on page 105
- “Configuring a Task Event Action” on page 106
- “Configuring Simple Task Actions” on page 107
- “Configuring a Date/Time Event” on page 108
- “Working with Task Events” on page 102

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 Description** — Type a description of the event. This text is also displayed in My webMethods.
- **Rule Set** — Type a new rule set name, or click an existing rule set name from the drop-down list. For more information on rule sets, see “Working with Task Rule Sets” on page 110.

Related Topics

- “Creating a Task Event” on page 103
- “Configuring a Task Event Type” on page 105
- “Configuring a Task Event Action” on page 106
- “Configuring Simple Task Actions” on page 107
- “Configuring a Date/Time Event” on page 108
- “Adding a Service to a Task” on page 82
- “Working with Task Events” on page 102

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 rule sets; this enables you to select different event behavior for different task implementations in the run-time environment. For more information about rule sets, see "Working with Task Rule Sets" on page 110.

For information about how to create a task event, see "Creating a Task Event" on page 103.

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 120). 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**, **Completed**, **Delegated**, **Description Updated**, **Name Updated**, **On Error**, **Priority Updated**, **Queued**, **Reassigned**, **Resumed**, **Suspended**, 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 108.
   - **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. For more information about filter expressions, see "About Filter Expressions" on page 125 and "About Condition Expressions" on page 123. No actions are associated with this event type.
- **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 96.

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 106.

Related Topics

- "Creating a Task Event" on page 103
- "Configuring Event General Information" on page 104
- "Configuring a Task Event Action" on page 106
- "Configuring Simple Task Actions" on page 107
- "Configuring a Date/Time Event" on page 108
- "Working with Task Events" on page 102

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 103.

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 107
- "Configuring Advanced Task Expressions" on page 120

5 Save the task.

Related Topics

- "Creating a Task Event" on page 103
- "Configuring Event General Information" on page 104
- "Configuring a Task Event Type" on page 105
- "Configuring Simple Task Actions" on page 107
- "Configuring a Date/Time Event" on page 108
- "Working with Task Events" on page 102

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.

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 120.

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. The Task Action Selection dialog box appears.
   - **Set Task Priority** — Sets the priority of the task to the specified value.
   - **Set Task Expiration** — Sets the expiration date for the task based on the defined date and time values.
   - **Set Task Name** — Sets the task name to the specified value.
   - **Set Task Description** — Sets the task description to the specified value.
   - **Set Task Custom ID** — Sets the task custom ID to the specified value.
   - **Set Business Data Field** — Sets the specified task business data field to the specified date and time values.
   - **Set Business Data Date Field** — Sets the specified task business data date field to the specified value.
   - **Suspend Task** — Places the task in Suspended status.
   - **Resume Task** — Resumes a suspended task.

- **Complete Task** — Places the task in Completed status.
- **Expires Task** — Places the task in Expired status.
- **Cancel Task** — Places the task in Canceled status.
- **Re-assign Task** — Re-assigns the task to a specified entry in the task assignee list.
- **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 82.
- **Send Notification** — Executes the specified task notification.
- **Invoke Service** — Executes the specified service. For specific information about this action, see "Adding a Service to a Task" on page 82.

5 Click **OK** to add the action.

6 Repeat steps 4-5 if you want to add another action.

7 Save the task.

Related Topics

- "Creating a Task Event" on page 103
- "Configuring Event General Information" on page 104
- "Configuring a Task Event Type" on page 105
- "Configuring a Task Event Action" on page 106
- "Configuring a Date/Time Event" on page 108
- "Working with Task Events" on page 102

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 105.

To define a date/time event type

1 Select a Date/Time event type as described in "Configuring a Task Event Type" on page 105. 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 Custom Field Type as Business Data" on page 70 and "Adding Task Business Data Fields to a View" on page 77.

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 146.

- 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 82.

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 106.

Related Topics

- "Creating a Task Event" on page 103
- "Configuring Event General Information" on page 104
- "Configuring a Task Event Type" on page 105
- "Configuring a Task Event Action" on page 106
- "Configuring Simple Task Actions" on page 107
- "Working with Task Events" on page 102

Working with Task Rule Sets

You can define one or more rule sets that are associated with a task, although there is not a requirement for a task to have a rule set. Rule 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 rule set for each set of assignees and events.

When the task is integrated into a process, you can specify which rule set applies to the task when it runs within that particular process. This enables you to create "basic" tasks and then add rule sets that adapt them for specific usages.

- Each assignment and event you create can be added to a rule set; you can define as many rule sets as necessary, containing any combination of assignments and/or events.
- An assignment or event can belong to only one rule set. If you want the same assignment or event behavior in more than one rule set, you must create a new assignment or event that copies the behavior of the original and associate it with another rule set.

- When an assignment or event belongs to no rule 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 rules set (as opposed to duplicating it in all rule sets).
- If no rule 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 rule set when adding a task to a process, see "Configuring Task Steps" in the *webMethods Process Development Help*.

Related Topics

- "Viewing Task Rule Sets" on page 111
- "Creating a Task Rule Set" on page 112
- "Associating an Assignment or Event with a Rule Set" on page 112
- "Removing an Assignment or Event from a Rule Set" on page 113
- "Configuring Tasks" on page 65

Viewing Task Rule Sets

You can view task rule 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 **Rule Set**. This column indicates which rule set the assignment or event is associated with. Click the column heading to sort the assignments or events by rule set.
- **Assignments and Events tabs** — Open the task editor and click the **Assignments** or **Events** tab. The General area contains a box labeled **Rule Set**. This box indicates which rule set the assignment or event is associated with.

Related Topics

- "Creating a Task Rule Set" on page 112
- "Associating an Assignment or Event with a Rule Set" on page 112
- "Removing an Assignment or Event from a Rule Set" on page 113
- "Working with Task Rule Sets" on page 110

Creating a Task Rule Set

You can create a rule set and associate task assignments and events with the rule set.

To create a task rule 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 **Rule Set** box, type the name of the rule set you want to create.

3 Save the task.

The assignment or event you are working with is now associated with rule set you created, and is available for use with other assignments and events.

Related Topics

- "Viewing Task Rule Sets" on page 111
- "Associating an Assignment or Event with a Rule Set" on page 112
- "Removing an Assignment or Event from a Rule Set" on page 113
- "Working with Task Rule Sets" on page 110

Associating an Assignment or Event with a Rule Set

To associate a task assignment or event with a rule 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 **Rule Set** box, select the name of the rule set.

3 Save the task.

The assignment or event you are working with is now associated with rule set you selected.

Related Topics

- "Viewing Task Rule Sets" on page 111
- "Creating a Task Rule Set" on page 112
- "Removing an Assignment or Event from a Rule Set" on page 113
- "Working with Task Rule Sets" on page 110

Removing an Assignment or Event from a Rule Set

To remove a task assignment or event from a rule 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 **Rule Set** box, delete the name of the rule set, or select the empty entry in the drop-down list.

3 Save the task.

The assignment or event you are working with is now removed from rule set you selected.

Note: If you remove all assignments and events from a rule set, the rule set is removed from the task. A task cannot support an empty rule set.

Related Topics

- "Viewing Task Rule Sets" on page 111
- "Creating a Task Rule Set" on page 112
- "Associating an Assignment or Event with a Rule Set" on page 112
- "Working with Task Rule Sets" on page 110

Modifying Task Client Options

When you create a new task, you can specify the following task client options:

- **Auto Accept** (default = true)—When the value for this property is set to true, the task is automatically accepted by the user when the user updates the task.
- **Need Accept to Update** (default = true)—When the value for this property is set to true, the user must accept a task before any actions or modifications can be done.
- **Enable User Routing** (default = false)—When the data binding for this property is set to true, the user can forward an instance of the task in the user's inbox to another user. This ability appears as a task functional privilege that can be assigned to a user role.

Note: You can also set these values in the Bindings view at a later time. For more information, see "Modifying Task Binding Properties" on page 116.

To modify a task client option

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 Details portlet.
4. Click the **Update** button in the User Interfaces area. The Task UI Update dialog box 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 Properties dialog box appears.
7. Select or clear the check boxes for the three available task client options as desired.
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.

Related Topics

- "Adding a Task to a Process" on page 29
- "Creating Tasks" on page 25
- "Configuring Tasks" on page 65

Working with Tasks in the Bindings View

When you open a portlet view in the portlet view editor, you can examine a wide range of task properties with the Bindings view. The Bindings view is populated automatically when you open the view in the portlet view editor. When you click an item in the Binding view hierarchy, the properties for that item are displayed in the Properties view.

For task views, you can modify task specific property elements, as described in "Modifying Task Binding Properties" on page 116. For more information about the available binding properties for a task, see "About Available Task Binding Properties" on page 115.

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 85.

Related Topics

- "Working with Task Privileges" on page 86
- "Working with Security Roles" on page 59
- "Configuring Tasks" on page 65

About Available Task Binding Properties

When you open a portlet view in the portlet 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. The portlet view task bindings listed below are among those most commonly modified.

The following properties are found by opening the Task View portlet default view and browsing to **Managed Beans > [*TaskName*]View/default > [*Task Name*]**:

- **Auto Accept** (default = true)—When the data binding value for this property is set to true, the task is automatically accepted by the user when the user updates the task.
- **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.
- **adHocRouting** (default = false)—When the data binding for this property is set to true, the user can forward an instance of the task in the user's inbox to another user. This ability appears as a task functional privilege that can be assigned to a user role.
- **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.

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** (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.

You can also modify the first three of the above bindings on the **Overview** tab of the task editor as described in "Modifying Task Client Options" on page 113.

Related Topics

- "Working with Tasks in the Bindings View" on page 114
- "Working with Task Privileges" on page 86
- "Working with Security Roles" on page 59
- "Configuring Tasks" on page 65

Modifying Task Binding Properties

To modify task 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 the Show Expert Properties  button in the Bindings view toolbar.

3 Locate the property element you want to work with; for example:

- Expand the **Managed Beans > [*TaskName*]View/default >[*Task Name*]** entries to view the following property elements available for the Task Details portlet default view:
  - **adhocRouting**
  - **Auto Accept**
  - **Need Accept to Update**
  - **Maximum Users can Accept**
- Expand the **Managed Beans > [*TaskName*]inboxResults/default > Task Search Provider/Search Query** entries to view the following property element available for the Task Inbox Search Results portlet default view:
  - **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 the  Expression Binding button 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.

Related Topics

- "About Available Task Binding Properties" on page 115
- "Working with Tasks in the Bindings View" on page 114
- "Working with Task Privileges" on page 86
- "Working with Security Roles" on page 59
- "Modifying Task Client Options" on page 113

- "Working with Tasks in the Bindings View" on page 114
- "Accessing the New Portlet View Wizard" on page 53

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 webMethods Designer, and anywhere the Task Type name is displayed in My webMethods. For more information, see "About Task Names" on page 23.

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.

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 "Changing the Task Type Inbox Tab Name" on page 118.

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 programatically 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.

Related Topics

- "About Task Names" on page 23
- "Changing the Task Type Inbox Tab Name" on page 118
- "Configuring Tasks" on page 65

Changing the Task Type Inbox Tab Name

Although you can change a task type name in Designer as described in "Changing the Task Type Name" on page 117, the task type inbox tab in My webMethods will still display the original task type name when you publish the task type to My webMethods Server.

To change the task type inbox tab name displayed in My webMethods

1. Open the task in the task editor by double-clicking the task in the Solutions view.
2. Click the **taskDefinition.xml** tab.
3. In the taskDefinition.xml file, locate the `CONTEXT alias=webm.apps.workflow` XML element and change its `folder name=` attribute to the new name of the task.
4. Save the task and publish it to My webMethods Server.
5. You must now log on to My webMethods Server as sysadmin and delete the previous task type inbox tab for this task, as described in the *webMethods Task Engine User's Guide* section, "Deleting a Task Type Inbox Tab".

Related Topics

- "Changing the Task Type Name" on page 117
- "About Task Names" on page 23
- "Configuring Tasks" on page 65

Working with 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 119. For specific information, see the following topics:

- "Configuring Advanced Task Expressions" on page 120
- "Building a Sample Task Expression" on page 121
- "Viewing Sample Task Expression Syntax" on page 123
- "About Condition Expressions" on page 123
- "About Filter Expressions" on page 125

Task Expression Overview

The task expression language used within webMethods 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 22.

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 120.

For further information about creating task expressions, see the following topics:

- "About Date/Time Values in Expressions" on page 127
- "About Expression Managed Beans" on page 128

Related Topics

- "Creating a Task Notification Event" on page 130
- "Configuring Simple Task Conditions" on page 96
- "Working with Task Notifications" on page 134
- "Configuring Tasks" on page 65

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. 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 "Working with Task Expressions" on page 118 and "Building a Sample Task Expression" on page 121.

4 Save the task to apply your changes.

Related Topics

- "Creating a Task Assignment" on page 92
- "Configuring Assignment General Information" on page 93
- "Configuring Assignees for a Task Assignment" on page 94

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 123.
- For more information about action expressions, see "About Action Expressions" on page 124.

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 120.
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 the Show Expert Properties  button 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**.

Related Topics

- "Working with Task Expressions" on page 118
- "Configuring Advanced Task Expressions" on page 120
- "Viewing Sample Task Expression Syntax" on page 123
- "About Condition Expressions" on page 123
- "About Action Expressions" on page 124
- "About Filter Expressions" on page 125
- "About Expression Operators" on page 126
- "About Date/Time Values in Expressions" on page 127
- "About Expression Managed Beans" on page 128

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 webMethods 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.

Related Topics

- "Working with Task Expressions" on page 118
- "Configuring Advanced Task Expressions" on page 120
- "Building a Sample Task Expression" on page 121
- "About Condition Expressions" on page 123
- "About Action Expressions" on page 124
- "About Filter Expressions" on page 125
- "About Expression Operators" on page 126
- "About Date/Time Values in Expressions" on page 127
- "About Expression Managed Beans" on page 128

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 120 and "Building a Sample Task Expression" on page 121.

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.

Related Topics

- "Working with Task Expressions" on page 118
- "Configuring Advanced Task Expressions" on page 120
- "Building a Sample Task Expression" on page 121
- "Viewing Sample Task Expression Syntax" on page 123
- "About Action Expressions" on page 124
- "About Filter Expressions" on page 125
- "About Expression Operators" on page 126
- "About Date/Time Values in Expressions" on page 127
- "About Expression Managed Beans" on page 128

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}
```

Related Topics

- "Working with Task Expressions" on page 118
- "Configuring Advanced Task Expressions" on page 120
- "Building a Sample Task Expression" on page 121
- "Viewing Sample Task Expression Syntax" on page 123
- "About Condition Expressions" on page 123
- "About Filter Expressions" on page 125
- "About Expression Operators" on page 126
- "About Date/Time Values in Expressions" on page 127
- "About Expression Managed Beans" on page 128

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.

Related Topics

- "Working with Task Expressions" on page 118
- "Configuring Advanced Task Expressions" on page 120
- "Building a Sample Task Expression" on page 121
- "Viewing Sample Task Expression Syntax" on page 123
- "About Condition Expressions" on page 123
- "About Action Expressions" on page 124
- "About Expression Operators" on page 126
- "About Date/Time Values in Expressions" on page 127
- "About Expression Managed Beans" on page 128

About Expression Operators

You can view the available expression operators by clicking the **Add Operator** button on the Edit Expression dialog box. The following operators are available:

Operator	Definition
+	Add
-	Subtract
*	Multiply
/	Divide
%	Modulus <<meaning? 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

Operator	Definition
`&&`	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

Related Topics

- "Working with Task Expressions" on page 118
- "Configuring Advanced Task Expressions" on page 120
- "Building a Sample Task Expression" on page 121
- "Viewing Sample Task Expression Syntax" on page 123
- "About Condition Expressions" on page 123
- "About Action Expressions" on page 124
- "About Filter Expressions" on page 125
- "About Date/Time Values in Expressions" on page 127
- "About Expression Managed Beans" on page 128

About Date/Time Values in Expressions

You can specify date and time values in an expression; for example:

`#{System.currentDateTime}` 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.

Related Topics

- "Working with Task Expressions" on page 118
- "Configuring Advanced Task Expressions" on page 120
- "Building a Sample Task Expression" on page 121
- "Viewing Sample Task Expression Syntax" on page 123
- "About Condition Expressions" on page 123
- "About Action Expressions" on page 124
- "About Filter Expressions" on page 125
- "About Expression Operators" on page 126
- "About Expression Managed Beans" on page 128

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}`.

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
- `#{currentUser}`—an instance of PortalUserModel or PortalRoleModel which represents the principal to which task is about to be assigned. This can be used to obtain user/role 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} = "Order Number: " +
#{currentTask.taskData.order.orderNumber};

#{currentTask.taskData.orderValue} = (#{currentTask.taskData.orderValue} * 100)
+ 10;

#{currentTask.applyChangesNoAccept};
```

Related Topics

- "Working with Task Expressions" on page 118
- "Configuring Advanced Task Expressions" on page 120
- "Building a Sample Task Expression" on page 121
- "Viewing Sample Task Expression Syntax" on page 123
- "About Condition Expressions" on page 123
- "About Action Expressions" on page 124
- "About Filter Expressions" on page 125
- "About Expression Operators" on page 126
- "About Date/Time Values in Expressions" on page 127

Creating a Task Notification Event

You can create a task notification that My webMethods users can subscribe to, as described in "Creating a Task Notification" on page 136 (the task must be assigned to the user before the user can subscribe to the notification).

After you create a task notification, you must create a task notification event to trigger the notification. Task notification can be specified as a single result in an event action, or it can be combined with other actions.

To create a task notification event

Note: At least one task notification must be created before you can create a task notification event. For more information, see "Creating a Task Notification" on page 136.

1. Create a task event as described in "Creating a Task Event" on page 103. 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 96.
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 136).
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.

For more information about expressions, see "Working with Task Expressions" on page 118.

Related Topics

- "Creating Tasks" on page 25
- "Working with Task Portlets" on page 33
- "Working with Task Notifications" on page 134
- "Configuring Tasks" on page 65

Working with Task Pages

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 Start
- Task Inbox Bar
- Task Inbox Results

By default, the following pages are available:

- Details
- Start
- Inbox

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.

You can run any available page in the webMethods Designer preview server, as described in "Running a Task Portlet in the Designer Preview Server" on page 42.

You can also:

- Rename a page as described in "Renaming a Task Page" on page 132.
- Hide or display a page as described in "Hiding and Displaying a Task Page" on page 133.

To access task pages

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.

Related Topics

- "Renaming a Task Page" on page 132
- "Hiding and Displaying a Task Page" on page 133

- "Working with Task Portlets" on page 33
- "Default Task Portlet Overview" on page 34
- "Configuring Tasks" on page 65

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 133.
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 "Deleting Task Type Pages" in the *webMethods Task Engine User's Guide*.

Related Topics

- "Hiding and Displaying a Task Page" on page 133
- "Working with Task Pages" on page 131
- "Configuring Tasks" on page 65

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**. The Page Properties dialog box appears.
6. 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.

Related Topics

- "Renaming a Task Page" on page 132
- "Working with Task Pages" on page 131
- "Configuring Tasks" on page 65

Working with Task Notifications

You can create a task 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 escalated.
- A large loan request or merchandise order is received.
- A task has not been acted upon within a certain time period.

For general information about task notifications, see "Task Notification Overview" on page 134. For specific information, see the following topics:

- "Accessing the Task Notification Wizard" on page 135
- "Creating a Task Notification" on page 136
- "Modifying and Deleting a Task Notification" on page 136
- "Task Overview" on page 20
- "Creating Tasks" on page 25
- "Working with Task Portlets" on page 33
- "Working with Task Notifications" on page 134

Task Notification Overview

webMethods Designer enables you to create one or more *task notifications* in a task application. Run-time users can then subscribe to available notifications for the tasks assigned to them, enabling them to receive e-mail notification of specific task-related events. In addition, a My webMethods administrator can subscribe a user to a task notification.

For 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 escalated.
- 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 to all subscribed users. Such events are referred to as task notification events.

When a task notification event is triggered, notification is provided by e-mail to all subscribed users. The notification is sent to the user's e-mail account defined in the My webMethods user profile. For more information about task notification events, see "Creating a Task Notification Event" on page 130.

Related Topics

- "Accessing the Task Notification Wizard" on page 135
- "Creating a Task Notification" on page 136
- "Modifying and Deleting a Task Notification" on page 136
- "Task Overview" on page 20
- "Creating Tasks" on page 25
- "Working with Task Portlets" on page 33
- "Working with Task Notifications" on page 134

Accessing the Task Notification Wizard

You can access the task notification wizard in the following ways:

- On the My webMethods 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**.

Related Topics

- "Working with Task Notifications" on page 134
- "Creating a Task Notification" on page 136
- "Modifying and Deleting a Task Notification" on page 136
- "Task Overview" on page 20
- "Creating Tasks" on page 25
- "Working with Task Portlets" on page 33
- "Working with Task Notifications" on page 134

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 135.

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; click on a control to examine and modify its properties in the Properties view.

5 You must create a task notification event as described in "Creating a Task Notification Event" on page 130.

Related Topics

- "Modifying and Deleting a Task Notification" on page 136
- "Working with Task Notifications" on page 134

Modifying and Deleting a Task Notification

To modify a task notification

1 In the Solutions view, expand **[*Solution*] > [*TaskApplicationProjectName*] > [*TaskName*] > 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.

If you need to modify the event type that triggers the notification, see "Configuring a Task Event Type" on page 105 for more information.

Related Topics

- "Creating a Task Notification" on page 136
- "Working with Task Notifications" on page 134

Publishing and Deleting a Task

After you create a task, you must publish the task application that contains it to the My webMethods Server run time so that it can run on the Task Engine. You may also decide the task is no longer needed and that it should be deleted.

For more information, see the following topics:

- "Publishing a Task" on page 138
- "Deleting a Task" on page 140

Publishing a Task

After you create a task in webMethods Designer, it remains available within the design-time environment of Designer but is not present in the run-time environment of My webMethods Server. To make the task available in My webMethods Server, you must publish the task application project containing the task to the run-time environment in My webMethods Server. 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 like any other portlet project. For more information about publishing portlet projects in general, see "Publishing a Portlet Application to the Server" in the *webMethods Composite Application Framework Help*.

Related Topics

- "Automatically Publishing Tasks in a Process" on page 138
- "About Optimized Task Publishing" on page 139
- "Forcing a Full Publication of a Task Application Project" on page 140
- "Task Application Overview" on page 17

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.

To enable automatic deployment of tasks

Note: To view the preference in the following procedure, you must first enable the Show/hide advanced properties  button in the main toolbar.

1 In the main menu, click **Window >Preferences**.

2 In the preference list, expand **Software AG** and click **Process Designer>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 139.

Related Topics

- "Publishing a Task" on page 138
- "About Optimized Task Publishing" on page 139
- "Forcing a Full Publication of a Task Application Project" on page 140
- "Task Application Overview" on page 17

About Optimized Task Publishing

When a task application project is first published to a run-time environment, the publishing process can take a considerable amount of time, especially if the task application is a complex one with many tasks, custom inboxes, 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, webMethods 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.

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 a Full Publication of a Task Application Project" on page 140.

Related Topics

- "Publishing a Task" on page 138
- "Forcing a Full Publication of a Task Application Project" on page 140
- "Task Application Overview" on page 17

Forcing a Full Publication of a Task Application Project

webMethods Designer optimizes the task application publication process by publishing *only those portions of the project that have changed* since the last publication.

To force a full publication of the entire task application

1 Install the task application from within My webMethods Server using the Install Administration functionality (for more information, see the topic "Installing Portlets or Other Deployable Server Components" in the *My webMethods Server Administrator's Guide*).

Related Topics

- "Deleting a Task Application Project" on page 19
- "Publishing a Task" on page 138
- "About Optimized Task Publishing" on page 139
- "Task Application Overview" on page 17

Deleting a Task

Tasks can be deleted in webMethods 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 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**.

You may also want to delete the task in My webMethods if the task has been published to My webMethods Server.

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 Engine Administration**.

3 Select the task you want to delete and click **Delete**.

Related Topics

- "Task Overview" on page 20
- "webMethods Task Development" on page 11

Working with Tasks in the Metadata Library

webMethods Designer automatically publishes task metadata to the metadata library. After publication, you can drag the task from the metadata library to the design canvas of a process, as described in "Adding an Existing Task to a Process" on page 31.

You can also view the metadata structure of the task. To do so, locate the existing task you want to add in the metadata Library view. Expand the various metadata elements to view their contents.

In some cases, you may not find the task metadata in the library; for example, after you import a project, or if you reset a local metadata store, thereby removing all existing metadata. In this case, you must manually publish the metadata to the library, as described in "Manually Publishing Task Metadata" on page 141.

Related Topics

- "Task Overview" on page 20

Manually Publishing Task Metadata

Tasks can be dragged from the metadata library and dropped on the process design canvas in the process editor. However, the task must first exist in the metadata library. Normally, this metadata is created automatically; if the task metadata is not present in the library, use the following procedure to extract it manually.

To manually publish task metadata to the metadata library

1. Locate the task you want to work with in the Solutions view.
2. Right-click the task and click **Publish**. The Publish Assets dialog box appears.
3. Click **OK**.

After publishing the task, you can drag the task from the metadata library to the design canvas of a process, as described in "Adding an Existing Task to a Process" on page 31.

Related Topics

- "Working with Tasks in the Metadata Library" on page 141
- "Task Overview" on page 20
- "Creating Tasks" on page 25

Working with Task Document Handling

Just like any portlet, you can add document handling capabilities to the tasks you create. You can provide users the ability to attach, view, update, and remove documents within a task. You can provide document handling on any task view by adding an Attachment Panel control to the view.

For more information about the Attachment Panel control, see "Attachments Panel" in the *webMethods Composite Application Framework Help*.

Note: A new attachment panel is added to the portlet view each time you complete this procedure.

To add document handling capability to 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 and expand the User Interfaces area if it is not already visible.
3. Click the default view of the Task Details portlet.
4. Click the **Update** button in the User Interfaces area. The Task UI Update dialog box 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 Task Client Properties dialog box 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** check box.
8. Click **Finish**. An attachments panel is added to the Task Data subgroup on the Task Detail view.
9. The changes to the task are saved automatically.

Related Topics

- "Task Overview" on page 20
- "webMethods Task Development" on page 11

Working with Custom Task Inboxes

When a My webMethods user logs on, all of the tasks assigned to that user appear in the user's My Inbox panel. Aside from the standard page editing capability provided with the toolbar  Display Options button, the My Inbox panel cannot be modified or customized.

However, 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.

When you create a task in webMethods Designer, you can choose to include several default user interface portlet views, as described in "Default Task Portlet Overview" on page 34. 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.

Related Topics

- "About Creating a Custom Task Inbox" on page 143
- "Example: Modifying the Default Search Fields and Results" on page 144
- "Task Overview" on page 20
- "Configuring Tasks" on page 65

About Creating a Custom Task Inbox

When you create a task in webMethods Designer, you can choose to include several default user interface portlet views. You can customize these views by removing existing interface components or by adding specific business date from within the task:

- 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.
- 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.

In addition, you can create a completely new portlet, as described in "Accessing the New Portlet Wizard" on page 46, and add to it a new portlet view, as described in "Accessing the New Portlet View Wizard" on page 53. 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.

Related Topics

- "Example: Modifying the Default Search Fields and Results" on page 144
- "Working with Task Document Handling" on page 142
- "Working with Custom Task Inboxes" on page 143
- "Task Overview" on page 20

Example: Modifying the Default Search Fields and Results

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 79.
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 79, 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.

Related Topics

- "About Creating a Custom Task Inbox" on page 143
- "Working with Custom Task Inboxes" on page 143
- "Working with Task Document Handling" on page 142
- "Task Overview" on page 20

Working with Tasks in 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.

Related Topics

- "Task Overview" on page 20
- "webMethods Task Development" on page 11

Finding Tasks in webMethods Designer

After you create a task, there are several methods for locating it within webMethods Designer. You can:

- Find the task in the Solutions view by expanding a solution folder and then expanding the Tasks folder.
- Find the task by using the webMethods search feature:
  - **a** With Designer running, press CTRL+H to open the Search dialog box.
  - **b** Click the **webMethods** tab (not **webMethods Advanced**).
  - **c** In the **Asset Type** box, select Task.
  - **d** In the **Name or description contains** box, type the task search term.
  - **e** Click **Search** and view the result in the Search view.

    You can drag the task from the Search view onto the process editor canvas.

- Find a task from within a process that contains a task:
  - **a** Open the process in the process editor.
  - **b** In the Outline view, expand the entries and click a task.
  - **c** In the Properties view, click **Advanced**.
  - **d** On the Task page in the Properties view, click the  button to the right of the **Task Information** area. The Tasks dialog box appears, displaying a list of all your tasks.

Related Topics

- "Task Overview" on page 20
- "webMethods Task Development" on page 11

Working with User and Business Calendars

webMethods Designer and My webMethods Server 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 *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 101.

Related Topics

- "Task Overview" on page 20
- "Configuring Tasks" on page 65
- "webMethods Task Development" on page 11

Index

Q

R

S

T

U

V

W