



BEA WebLogic Integration™

Using Worklist User Portal

Contents

1. Introduction to Worklist User Portal	
About Worklist User Portal	1-2
2. Worklist User Portal	
Login to Worklist	2-2
Worklist User Portal Pages	2-3
Portlets in Worklist User Portal	2-4
Creating a Task Instance	2-6
Creating a Task Instance Associated with an Interactive Assignee List	2-8
Viewing Upcoming Tasks	2-11
Viewing Assigned Tasks	2-12
Claiming a Task Instance	2-12
Working on a Task Instance	2-13
Viewing the Task Instance Details	2-15
Searching for a Task	2-16
Browsing Tasks based on Task Plan	2-17
Viewing Overdue Tasks	2-18
Task List Page	2-19
Customizing the Task List	2-21
Deleting a Task Instance	2-22
3. Customizing Worklist User Portal	
Typical Workflow	3-2

Implementing the Custom Task UI Page Flow	3-2
Creating a Page Flow	3-5
Creating Form Beans	3-9
Generating Setter and Getters Methods for Properties	3-10
Defining Actions on a Page Flow	3-11
Action Methods and Form Beans	3-11
Worklist Property Editors and Actions	3-12
Creating Actions on a Page Flow	3-13
Defining JSP Pages	3-14
Creating JSP Files	3-14
Registering the Custom Task UI	3-16
Deploying the Custom Task UI	3-19

Introduction to Worklist User Portal

WebLogic Integration Worklist provides mechanisms to interact with human actors from WebLogic platform applications. Worklist is intended to complement BEA WebLogic Integration™ Java Process Definition (JPD) component. JPD is intended to handle automated business processes and does not directly concern itself with the nuances of human interaction. Worklist extends the reach of the JPD component to include human actors where required.

Worklist enables human interaction, via the capability of assigning tasks to human users, with business processes. Based on the assigned task, human users can perform actions on the tasks, which can trigger new task assignments to other users or system events. This process flow depends on the higher level business processes. For more information about Worklist, see [Using Worklist](#).

In Worklist, a task represents an activity that can be assigned to and claimed by a human user. The user oversees the progress of the task and ensures that it is completed in a correct and timely manner. A task can include one or more steps, and one or more exit points (called terminal steps). Each step represents a distinct phase in the completion of the task. Each step defines one or more actions and users move tasks through their defined steps by taking actions on the task at each step. The task moves through a set of planned steps in a manner dictated by the actions taken on the task.

The steps and properties for a given type of task are defined by a Task Plan and apply to all instances of tasks of that type. All tasks of a given type are then governed by the plan defined for that type of task. Task plans are defined by a designer prior to the creation of any task instance of the type being governed by the plan. For more information, see [Creating and Managing Worklist Task Plans](#) in *Using Worklist*.

Worklist User Portal is a web application that presents a WebLogic Portal user interface for interacting with a Worklist system instance and the tasks it is managing. It provides facilities for Worklist users (not administrators/designers) and allows them to see the tasks assigned to them or the groups they belong to. Worklist User Portal allows users to claim tasks, take actions on them, and alter their properties.

About Worklist User Portal

Worklist User Portal is an all-purpose user interface that uses the metadata for steps and actions to render an interactive interface at each step. The user interface for a given step and action will differ from the user interface for other steps and actions by virtue of the differences in the metadata for those steps and actions.

Worklist User Portal is a part of the Worklist domain and provides an interface for Worklist users to manage task instances that they are authorized to deal with. This portal provides basic out-of-the-box functionality for creating, monitoring, and updating tasks.

Worklist User Portal can be used by two types of users: an administrator kind of user and a normal user. The functions that these users can do differs based on permissions.

Functions that an administrator kind of user can do would be:

- Create tasks based on task plans
- Find tasks created by users
- View history and track status of tasks

Functions that a normal kind of user can do would be:

- Claim tasks for which the user is listed in the assignee list
- Work on the claimed tasks
- Search for tasks that are assigned or claimed by the user
- Plan and understand the work scheduled associated with the user

Worklist User Portal contains multiple portlets. Portlets are pluggable user interface components that are managed and displayed in the portal. Each portlet allows you to do different functions. For example, you can search for specific tasks in the Quick Search portlet. For more information, see [“Portlets in Worklist User Portal” on page 2-4](#).

Users work on tasks associated with them from either the Worklist User Portal or a custom Task user interface (if a custom UI is developed for a step in the task plan or the entire task plan).

Worklist User Portal allows users to see the tasks assigned to them, claim tasks, take actions on tasks, and update the worklist properties (values) associated with the actions they perform. Worklist User Portal is used by Worklist users, not administrators or designers. For more information, see [Chapter 2, “Worklist User Portal”](#).

If Worklist User Portal does not meet the needs of a given task plan, a custom task user interface can be designed and developed by a Developer and associated with the entire task plan or just a step in the task plan. This ensures that the required user interface can be designed without forcing the implementation of a completely new web user interface. For more information, see [Chapter 3, “Customizing Worklist User Portal”](#).

Introduction to Worklist User Portal

Worklist User Portal

Worklist provides a web-based user portal for interacting with tasks. This portal allows users to see the details of a task and take actions on the task to move it through its lifecycle. Each step in a task plan can define unique actions and thus present its own unique considerations to the user.

This document provides the following information:

- [Login to Worklist](#)
- [Portlets in Worklist User Portal](#)
- [Creating a Task Instance](#)
- [Creating a Task Instance Associated with an Interactive Assignee List](#)
- [Viewing Upcoming Tasks](#)
- [Viewing Assigned Tasks](#)
- [Claiming a Task Instance](#)
- [Working on a Task Instance](#)
- [Viewing the Task Instance Details](#)
- [Searching for a Task](#)
- [Browsing Tasks based on Task Plan](#)
- [Viewing Overdue Tasks](#)

- [Task List Page](#)
- [Deleting a Task Instance](#)

Login to Worklist

To login to Worklist User Portal:

1. Open a Web browser and enter the following URL:

```
http://localhost:7001/<web project name>/
```

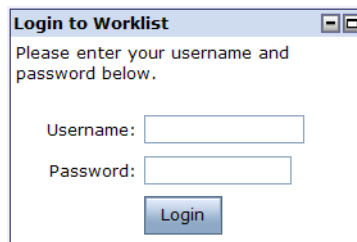
The Web project corresponds to the Web application of Worklist that acts as the user interface for the system. For more information, see “Worklist Application” in [Creating and Managing Worklist Task Plans](#) in *Using Worklist*. You can use any external browser, for example Internet Explorer, or the default browser that comes with Workshop for WebLogic Platform.

Worklist User Portal is displayed.

Note: E-mail notifications regarding tasks associated with you contain the URL to the specific task instances. You can also click on this URL to access the task in the Worklist User Portal.

2. In the Login to Worklist portlet, enter your user name and password.

Figure 2-1 Login to Worklist User Portal

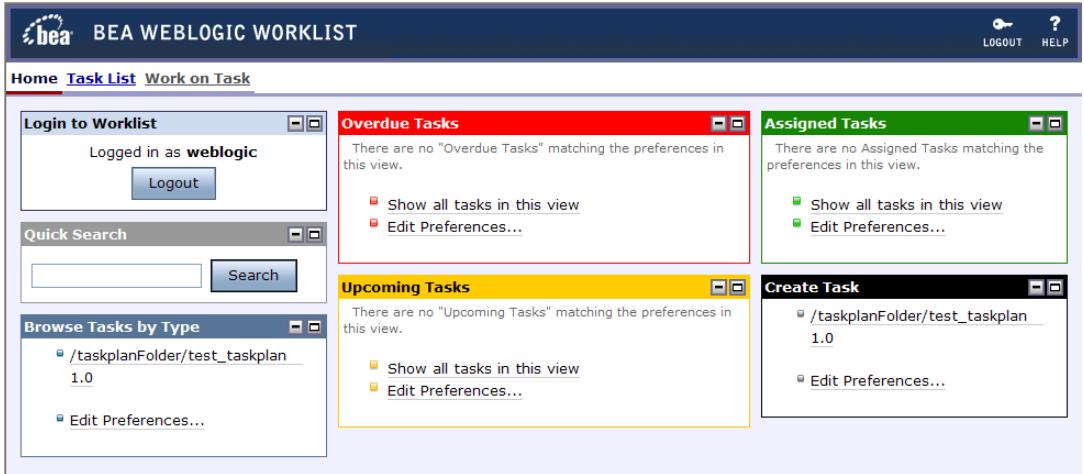


The screenshot shows a window titled "Login to Worklist". Inside the window, there is a text prompt: "Please enter your username and password below." Below this prompt are two input fields: "Username:" followed by a text box, and "Password:" followed by a text box. At the bottom of the form is a blue button labeled "Login".

3. Click Login.

Assigned, upcoming, overdue tasks associated with you are displayed in the appropriate portlet in the User Portal home page. Task plans based on which you can create task instances are displayed in the Create task portlet.

Figure 2-2 Worklist User Portal Home Page



Note: After you are done, click Logout to exit from Worklist User Portal.

Worklist User Portal Pages

The Worklist User Portal consists of three pages - Home, Task List, and Work on Task. The different functions that can be done in each of these pages are as follows:

Page Name	Actions that can be done in the page
Home	<ul style="list-style-type: none">• Login and logout of the Worklist User Portal.• Understand your workload and plan your schedule based on tasks listed in the Upcoming Tasks and Overdue Task portlets.• View tasks assigned to you in the Assigned tasks portlet.• View tasks assigned to you or claimed by you in the Upcoming Tasks or Overdue Tasks portlets.• Claim tasks listed in the Assigned tasks portlet.• View tasks created or owned by you in the Upcoming Tasks or Overdue Task portlets.• Create tasks based on task plans.• Quickly search for a task based on name in the Quick Search portlet.• Browse for task instances created against the task plan. <p>Note: All the default portlets are displayed in the Home page. For more information, see “Portlets in Worklist User Portal” on page 2-4.</p>
Task List	<ul style="list-style-type: none">• View tasks assigned, claimed, or owned by you.• View upcoming and overdue tasks.• View tasks that are associated with you. You can be associated with a based on previous task affinity - you are owner, creator or claimant of the task.
Work on Task	<ul style="list-style-type: none">• Work on tasks that have been claimed by you.• View task history.• View details of the tasks. You can view and edit task details based on your permissions. <p>Note: If any custom task user interface is defined for the current step in the task plan or the entire task plan associated with the task, that interface is displayed instead of the default interface. For more information, see Chapter 3, “Customizing Worklist User Portal.”</p>

Portlets in Worklist User Portal

Worklist User Portal contains the following default portlets:

Portlet Name	Actions that can be done
Login to Worklist	Use your credentials to login to Worklist. For more information, see “Login to Worklist” on page 2-2 . After you are done, click Logout to exit from Worklist User Portal.
Create Task	Create a task instance based on any existing task plan. For more information, see “Creating a Task Instance” on page 2-6 .
Assigned Tasks	View a lists of tasks that are assigned to you. The assigned tasks are tasks assigned to a user and possibly other users or groups. You can claim the task or directly claim and start working on the task. For more information, see “Viewing Assigned Tasks” on page 2-12 , “Claiming a Task Instance” on page 2-12 , and “Working on a Task Instance” on page 2-13 . In addition, you can click on the task name to view the task details. For more information, see “Viewing the Task Instance Details” on page 2-15 .
Upcoming Tasks	View a list of upcoming tasks assigned to you. Tasks for which you are the owner are also listed in this portlet. You can click on the task to view the task details, edit the task details, claim the task, claim and start work on the task, or view history of the task. For more information, see “Viewing Upcoming Tasks” on page 2-11 .
Overdue Tasks	View a list of tasks assigned to you that have not been completed before the task completion due date. For more information, see “Viewing Overdue Tasks” on page 2-18 .
Browse Tasks by Type	Browse all tasks listed for all the task plans that you can Query. Click on any task plan to view all the task instances that are created based on the selected task plan. All the task instances are listed in the Task List page. For more information, see “Browsing Tasks based on Task Plan” on page 2-17 .
Quick Search	Search for tasks that match the specified task name criteria. The search results are displayed in the Task List page. For more information, see “Searching for a Task” on page 2-16 .

Note: Tasks that are displayed in the portlets are based on your permissions. For example, the Browse Tasks by Type portlet will show only those tasks for which the current user has Query privileges. The Create New Task portlet will show only those tasks for which the

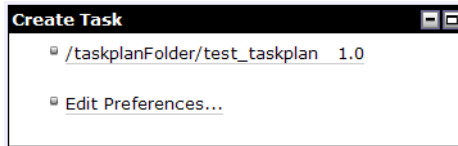
current user has Create privileges. For more information, see “Security Policies” in [User Management](#) in *Using Worklist Console*.

Creating a Task Instance

You can create a task instance based on any existing task plan in the Create Task portlet. For more information about task plans, see [Creating and Managing Worklist Task Plans](#) in *Using Worklist*.

Note: You can create a task instance only if you have task creation privileges. For more information, see “Security Policies” in [User Management](#) in *Using Worklist Console*.

Figure 2-3 Create Task Portlet



Note: Editing preferences for this portlet is similar to the Browse Tasks by Type portlet. For more information, see [“Edit Preferences of Browse Tasks by Type Portlet”](#) on page 2-17. Changes to the preferences are only valid for the current session.

To create a task instance:

1. In the Create Tasks portlet, click on the task plan based on which you want to create the task instance. The Create New Task page is displayed.

Figure 2-4 Create Task Instance

Task Name is required.

Create New Task of type /taskplanFolder/test_taskplan

* Indicates a required field.

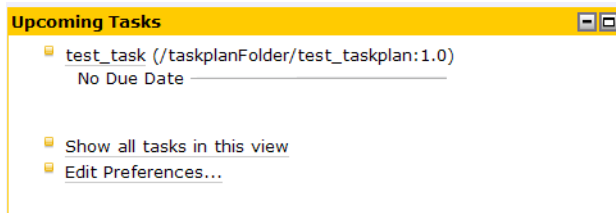
*Task Name:	<input type="text"/>	Task Name is required.	name of the task
*Task Plan Constructor:	Constructor 1 <input type="button" value="Refresh"/>		select type of constructor from list
Start Step:	Step1		start step for the selected constructor
Comment:	<input type="text"/>		enter comments here
DUE DATES			
Completion Due Date:	<input type="text"/>		Nov 13, 2006 10:47:02 PM
Start Step Completion Due Date:	<input type="text"/>		Nov 13, 2006 10:47:02 PM
Task Time Estimate:	<input type="text"/>		<days>dt <hours>h <minutes>m format

2. Enter a valid and unique name for the task.

Note: Make sure that you do not enter an empty string as task name.
3. Select the task plan constructor based on which the task instance should be created.
4. If required, enter any comments relevant to the task instance.
5. If required, enter the date by which the task should be completed. Enter the date in Month dd, yyyy h:mm:ss AM/PM format. For example, you can enter date as Nov 28, 2006 06:00:00 AM.
6. If required, enter the date by which the starting step in the task should be completed. Enter the date in Month dd, yyyy h:mm:ss AM/PM format.
7. If required, enter the estimated time required to complete the task. Enter the time in <days>dt <hours>h <minutes>m format.
8. Click Create Task.

The task instance is created and shows up in the Upcoming Tasks portlet on the home page.

Figure 2-5 New Task Instance Listed in Upcoming Tasks Portlet



Note: The task shows up on the Inbox of the current user because the current user is the owner of this task instance. By default, the user who creates the task instance becomes the owner of that task instance. This enables the owner to track the status of the task instance although the owner is not assigned to work on the task instance.

Creating a Task Instance Associated with an Interactive Assignee List

If Interactive is selected as the value of Candidate List Handling, a human actor can use Worklist User Portal or Custom Task UI, if any, defined for the task plan or step in a task plan. When candidate list handling is interactive, you need to manually assign task through the portal interface either from the Create Task portlet or the Work on Task page (if the task requests interactive assignment on the current step).

Interactive assignment of a user or group to a task involves:

- Generating a list of candidate claimants
- Evaluating the availability and workload of each candidate claimant
- Selecting a candidate that has an appropriately low workload and sufficient availability as claimant for the task.

If required, modify the initial assignee list to include users or groups to generate an appropriate list of candidates.

To create a task instance and provide assignee instructions:

1. Do [step 1](#) through [step 7](#) in the “[Creating a Task Instance](#)” on [page 2-6](#).

Figure 2-6 Creating a Task Instance and Assigning Claimant Interactively

Create New Task of type /taskplanFolder/test_taskplan

* Indicates a required field.

*Task Name:	<input type="text" value="InteractiveTask"/>	name of the task						
*Task Plan Constructor:	<input type="text" value="Constructor 1"/> <input type="button" value="Refresh"/>	select type of constructor from list						
Start Step:	Step1	start step for the selected constructor						
Comment:	<input type="text"/>	enter comments here						
DUE DATES								
Completion Due Date:	<input type="text"/>	Nov 24, 2006 12:46:21 AM						
Start Step Completion Due Date:	<input type="text"/>	Nov 24, 2006 12:46:21 AM						
Task Time Estimate:	<input type="text"/>	<days>dt <hours>h <minutes>m format						
ASSIGNEE INSTRUCTIONS								
Data Range:	From: <input type="text" value="Nov 24, 2006 12:38:47 AM"/> To: <input type="text" value="Dec 1, 2006 12:38:47 AM"/> <input type="button" value="Refresh"/>	e.g. Nov 24, 2006 12:46:21 AM						
Assignee List:	<table border="1"> <thead> <tr> <th></th> <th>Name</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>Joe (selected claimant)</td> </tr> <tr> <td><input type="checkbox"/></td> <td>John</td> </tr> </tbody> </table> <p><input type="button" value="Delete"/> <input type="button" value="Select Claimant"/> <input type="button" value="Clear Claimant"/> <input type="button" value="Show Score"/> <input type="button" value="Restore Default"/></p>		Name	<input type="checkbox"/>	Joe (selected claimant)	<input type="checkbox"/>	John	<p>This is the list from which the claimant for this task will be selected.</p> <p>To select a claimant click the checkbox next to the name of the user, then click the Select Claimant button</p>
	Name							
<input type="checkbox"/>	Joe (selected claimant)							
<input type="checkbox"/>	John							
Add to Assignee List:	User <input type="text"/> <input type="button" value="Add to List"/> <input type="button" value="Select Claimant"/>							
<input type="button" value="Create Task"/> <input type="button" value="Cancel"/>								

2. Specify the date range for which the user’s workload score and free time calculation should be based.

Any existing assignees for the task (based on the assignment instructions specified while creating the task plan associated with this task instance) are listed in the Assignee list. You can do any of the following actions with the users displayed in this list:

- Select the check box next to any user and click Select Claimant to claim this task on behalf of the selected user.
- Click Clear Claimant to remove the selected claimant for the task and potentially assign it to some other user.
- Select the check box next to any user and click Delete to remove that user from the assignee list.

- Click Restore Default to view the default values in all the fields in the Assignee Instructions section of this page.
- Click Show Score to view the availability, workload score and cumulative score of all the users in the assignee list.

If you do not want to view the score, click Hide Score.

Figure 2-7 Availability and Score of Users in the Assignee List

Assignee List:

	Name	Availability (minutes)	Workload Score	Cumulative Score
<input type="checkbox"/>	Joe	2400	10	2390.0
<input type="checkbox"/>	John	2400	20	2380.0

|
 |
 |
 |

3. Select Group or User from Add to Assignee List, enter valid user or group name (based on selection), and click Add to List.

When you specify a group, all users in that group are automatically added to the Assignee list.

You can also select User, enter a valid user name and click Select Claimant to automatically claim the task for the selected user.

4. Click the assignee name to view the availability details of the user as shown in [Figure 2-8](#). You can do the following tasks in this page:
 - View the number of tasks claimed by the user, the workload score for the user, the availability of the candidate (includes date and time range start time and duration). These details are applicable to the specified date range.
 - Enter a new date range, click Refresh and view the recalculated availability details of the user.
 - Click Select as Claimant to claim the task for this user after looking at the availability and workload of the user.
 - Click Done to return the Create a Task page.

Figure 2-8 Workload of selected user

The screenshot shows a 'Create Task' dialog box with the following sections:

- WORKLOAD**
 - Claimed Task Count: 1
 - Workload Score: 10
- AVAILABILITY**
 - Date Range:
 - From: Nov 24, 2006 12:38:47 AM
 - To: Dec 1, 2006 12:38:47 AM
 - Refresh button
 - e.g. Nov 24, 2006 12:49:38 AM

Start Time	End Time	Duration in Minutes
Fri Nov 24 09:00:47 IST 2006	Fri Nov 24 17:00:47 IST 2006	480
Mon Nov 27 09:00:47 IST 2006	Mon Nov 27 17:00:47 IST 2006	480
Tue Nov 28 09:00:47 IST 2006	Tue Nov 28 17:00:47 IST 2006	480
Wed Nov 29 09:00:47 IST 2006	Wed Nov 29 17:00:47 IST 2006	480
Thu Nov 30 09:00:47 IST 2006	Thu Nov 30 17:00:47 IST 2006	480

Buttons: Done, Select As Claimant

5. Click Create Task.

The task instance is created and shows up in the Upcoming Tasks portlet on the home page.

Note: The task shows up on the Inbox of the current user because the current user is the owner of this task instance. By default, the user who creates the task instance becomes the owner of that task instance. This enables the owner to track the status of the task instance although the owner is not assigned to work on the task instance.

Viewing Upcoming Tasks

You can view your scheduled tasks in the Upcoming Tasks portlet (Figure 2-5). In this portlet, you can view:

- Tasks claimed by you that have not been completed before the task completion due date and for which the completion due date is within the selected date range.
- Tasks owned by you that have not been completed before the task completion due date and for which the completion due date is within the selected date range.
- Tasks created by you that have not been completed before the task completion due date and for which the completion due date is within the selected date range.

Note: By default, the start date is current date and time and the end date of the date range is 7 calendar days from current date.

- Tasks claimed, owned, and created by you that have not been completed by the task completion due date.

By default, a maximum of 8 tasks are displayed in this portlet. To view all the upcoming tasks, click the Show all tasks in this view link. All upcoming tasks are displayed in the Task Link page. Click on the task name to view the task details in the Work on Task page. For more information, see [“Task List Page” on page 2-19](#).

Note: Editing preferences for this portlet is similar to the Overdue Tasks portlet. For more information, see [“Edit Preferences in the Overdue Tasks Portlet” on page 2-18](#). Changes to the preferences are only valid for the current session.

Viewing Assigned Tasks

You can view tasks assigned to you in the Assigned Tasks portlet ([Figure 2-9](#)). In this portlet, you can:

- Click on Claim to claim the task. For more information, see [“Claiming a Task Instance” on page 2-12](#).
- Click on Work to start working on the task. For more information, see [“Working on a Task Instance” on page 2-13](#).

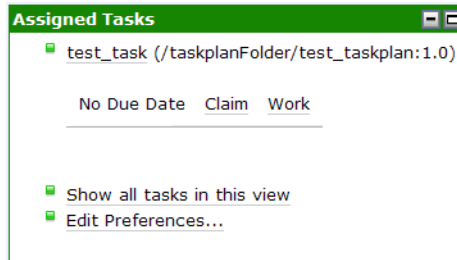
By default, a maximum of 8 tasks are displayed in this portlet. To view all the assigned tasks, click the Show all tasks in this view link. All assigned tasks are displayed in the Task Link page. Click on the task name to view the task details in the Work on Task page. For more information, see [“Task List Page” on page 2-19](#).

Note: Editing preferences for this portlet is similar to the Browse Tasks by Type portlet. For more information, see [“Edit Preferences of Browse Tasks by Type Portlet” on page 2-17](#). Changes to the preferences are only valid for the current session.

Claiming a Task Instance

If you are in the assignee list of a task, you can claim the task as follows:

1. In Worklist User Portal, locate the task that you want to claim.

Figure 2-9 Task Assigned to User Listed in Assigned Tasks Portlet

2. Click **Claim** against the required task. This action moves the task from your Assigned Tasks portlet to your Upcoming Tasks portlet as shown in [Figure 2-5](#).

When a user in a group assigned to this task claims it, this task is only displayed in the Upcoming Tasks portlet of that user. It is no longer displayed for any user in the group assigned to this task.

Note: You can click **Work** to directly claim and work with the task. For more information, see [“Working on a Task Instance” on page 2-13](#).

Working on a Task Instance

To work on a task that is assigned to you or that you have claimed:

1. Click on the task in the task name in your Upcoming Tasks or Overdue Tasks portlet. In addition, you can click on the **Work** link (associated with the task) in the Assigned Tasks portlet. The various actions that you can take in the current step of the task are displayed on the **Work on <task name> in <step name> Step** page.

Note: You can also click on the **Work on Task** link to work on a task. In this page, you can view the current step of the task, take actions on the task, see and edit details of the task, and reassign the task.

If any custom task user interface is defined for the current step in the task or the task plan associated with the task, that interface is displayed instead of the default interface. For more information, see [Chapter 3, “Customizing Worklist User Portal.”](#)

Figure 2-10 Working on a Task

The screenshot shows a web application window titled "Task Work". The main heading is "Work on 'test_task' in 'Step1' Step". Below this is a section for "TASK GENERAL INFORMATION" with a "VIEW" link. The task details are as follows:

Task Name:	test_task
Current Step:	Step1
Comment:	
Priority:	1
Owner:	weblogic
Claimant:	John
Current Assignee(s):	Groups [group1]

Below the task details are sections for "PROPERTIES" and "ACTIONS". The "ACTIONS" section contains three radio button options: "Action 1", "Action 2", and "Action 3". At the bottom of the window are four buttons: "Next >", "Edit / View Details", "History", and "Cancel".

2. If required, you can click Edit / View Details to view the task details or click History to view the all the events related to the current task. After you are done, click the Back arrow in the browser to return to this page.
3. Select the action and click Next. You can view the details of the selected action in the task.

Figure 2-11 Details of Action Selected in Task

The screenshot shows a window titled "Task Work" with a sub-header "'Action 3' test_task". The main content area is divided into two sections: "TASK GENERAL INFORMATION" and "KEY ACTION PROPERTIES". Under "TASK GENERAL INFORMATION", there are three rows: "Task Name: test_task", "Description:", and "Comment:". The "KEY ACTION PROPERTIES" section contains the text "No required properties." At the bottom of the window, there are three buttons: "Submit", "View More Info", and "Cancel".

4. Specify the required properties, if any, for the action.

If any user properties defined for the task plan are required for this action, they are displayed in the Key Action Properties section of this page.

5. If required, click View More Info to view the task details.
6. Click Submit.

The task transitions to the next logical entity as defined in the task plan.

Viewing the Task Instance Details

Based on your task affinity (owner, creator, claimant, or assignee for the task) or state of the task, you can view the task details from the Assigned Tasks, Upcoming Tasks, or Overdue Tasks portlet. You can also click the Task List link in the home page and click Upcoming Tasks, Associated Tasks, Claimed Tasks, Owned Tasks, Assigned Tasks, or Overdue Tasks link to view a list of tasks related to you.

To view the task details from a relevant portlet in the User Portal home page:

1. Click the task name in the portlet. The View more details of <task name> page appears.

You can view task details like task name, current step name, task ID, task plan, date of task creation and last update, Administrative and Working state of the task.

Figure 2-12 Viewing Task Details

The screenshot shows a window titled "Task Work" with a header "View More Details of test_task". The content is organized into several sections:

- TASK GENERAL INFORMATION:**
 - Task Name: test_task
 - Current Step: Step1
 - Comment:
- PROPERTIES:** (with an [EDIT](#) link)
- TASK DETAIL INFORMATION:**
 - Task ID: 5481434252111622650-748a7f14.10ef08c6a69.-7fd1
 - Task Plan: /taskplanFolder/test_taskplan:1.0:test_EAR
 - Last Update: Thu Nov 23 18:30:05 IST 2006 by weblogic
 - Creation: Thu Nov 16 19:33:29 IST 2006 by weblogic
 - Administrative State: Active
 - Working State: Assigned
 - Priority: 1 (Priority for this task)
 - Owner: weblogic
 - Claimant:
 - Current Assignee(s): Groups [group1]
- DUE DATES:**
 - Current Step Completion Due Date: Nov 23, 2006 10:13:08 PM
 - Task Completion Due Date: Nov 23, 2006 10:13:08 PM

At the bottom, there are four buttons: Claim, Claim & Work, History, and Cancel.

2. You can edit any editable properties of the task.
3. You can click on Work to start work on the task or click History to view the event history of the task.
4. Click Submit.

Searching for a Task

In the Quick Search portlet, you can search for any tasks that match for the specified task criteria.

To search for a task:

1. In the Quick Search portlet, enter the search criteria.

You can use the * (asterisk) character to match one or more characters.

2. Click Search.

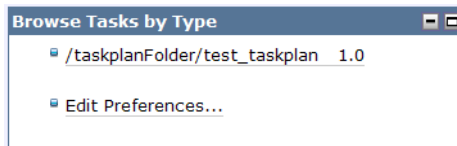
All tasks for which you have query privileges and that match the search criteria are displayed in the Task List page. For more information, see [“Task List Page” on page 2-19](#).

Browsing Tasks based on Task Plan

You can browse for tasks based on specific task plan in the Browse Tasks by Type portlet. This portlet shows all task plans for which you have query privileges. For each task plan, the complete hierarchical path and version is displayed. Click on the task plan name to view the tasks instances of this task plan in the Task List page. For more information, see [“Task List Page” on page 2-19](#).

Note: You have query privileges if you are assigned to the roles listed in the Query policy for the task plan. For more information, see “Security Policies” in [User Management in Using Worklist Console](#).

Figure 2-13 Browse Tasks associated with Various Task Plans

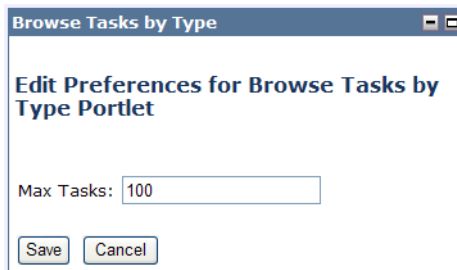


Edit Preferences of Browse Tasks by Type Portlet

To edit preferences of the Browse Tasks by Type portlet:

1. Click Edit Preferences.

Figure 2-14 Edit Preferences of Browse Tasks by Type Portlet



2. Enter the maximum number of tasks that you want to view this portlet. The default value is 100.
3. Click Save.

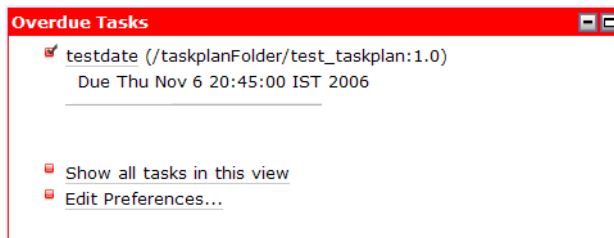
The preferences for this portlet are saved and you are returned to the Home page. These changes to the preferences are only valid for the current session.

Note: You can edit preferences of the Assigned Tasks and Create Tasks portlet in a similar manner.

Viewing Overdue Tasks

You can view tasks that are overdue in the Overdue Tasks portlet. This portlet shows any tasks claimed, owned, or created that have not been completed before the completion due date specified for the task. That is, tasks that have not been completed and for which the completion due date is earlier than the current date are displayed in this portlet. Click on the task name to view the task details in the Work on Task page.

Figure 2-15 Overdue Tasks Portlet



By default, a maximum of 8 tasks are displayed in this portlet. To view all the overdue tasks, click the Show all tasks in this view link. All overdue tasks are displayed in the Task Link page. Click on the task name to view the task details in the Work on Task page. For more information, see [“Task List Page” on page 2-19](#).

Edit Preferences in the Overdue Tasks Portlet

You can edit preferences to set the display of tasks in the Overdue Tasks portlet.

1. Click Edit Preferences.

Figure 2-16 Edit Preferences for Overdue Tasks Portlet

2. Enter the maximum number of tasks that you want to view this portlet. The default value is 8.
3. Enter the date and time range that is taken into account to display tasks in this portlet. Only tasks that are overdue within the start and end dates specified by you are listed in this portlet.
4. Select `True` instead of `False` to view tasks that are not associated with a due date in this portlet.
5. Click `Save`.

The preferences for this portlet are saved and you are returned to the Home page. These changes to the preferences are only valid for the current session.

Note: You can edit preferences of the Upcoming Tasks portlet in a similar manner.

Task List Page

The task list page has a set of views shown in a portlet at the left-hand side of the page. These views define a range of tasks that will be displayed when the view is selected. Tasks meeting the criteria for the view are shown in the task list portlet at the right-hand side of the page. Clicking different views in the view list causes the contents of the task list at the right to change. For example, click the Assigned Tasks link to view all the tasks that are assigned to you. The various views that are available in this page are:

- Upcoming Tasks
- Associated Tasks (tasks in any of the below views)

- Claimed Tasks (tasks claimed by you, sorted by creation date)
- Owned (tasks owned by you, sorted by creation date)
- Assigned Tasks (tasks assigned to you, sorted by creation date)
- Overdue Tasks (tasks that are not completed before the due date)

Figure 2-17 Task List Page



You can click on any task to view the task details. For more information, see [“Viewing the Task Instance Details” on page 2-15](#).

The task list may be long. If the list is longer than the maximum number of tasks configured to be listed in this page, the page will show only that maximum number. Click on the arrow buttons to navigate to the next page to see a continuation of the list.

Every column in the task list have a sort header. Clicking on the sort header for a column sorts the tasks in the list by that column. Clicking repeatedly on a header toggles between sorting in ascending order and sorting in descending order. The current sort order will be shown by an up or down arrow icon on the header.

By default, tasks in this view are sorted by:

- Ascending Due Date
- Descending Priority
- Ascending Creation Date

Customizing the Task List

You can customize the column headers or search for required tasks by clicking on Customize Table Header. The option allows you to specify the columns to be showed in the list table, to provide the sorting order, and to set the page size. The sorting order could be complex with option to provide multiple columns for sorting and the order (ascending/descending) for each of them.

Figure 2-18 Customizing the task list

Customize Table Header ***

CONSTRUCT A CUSTOM QUERY

Task Name: Use * in start or end as wild card.

Task IDs: Enter comma separated values

Task State: Unassigned Assigned Claimed
 Active Suspended Completed Aborted Error

Comment (regex):

Priority from: to:

Assignee Contains:

Claimant Contains:

Owner Contains:

Completion Due Date (After): Nov 21, 2006 3:29:44 AM

Completion Due Date (Before): Nov 21, 2006 3:29:44 AM

Include tasks with no completion due date:

View

Column Display

Available	Selected
Admin State	Task Plan ID
Assignees	Create Date
Claimant	Summary State
Comment	Current Step Name
Completion Due Date	Completion Due Date
Constructor Name	Owner

Preserve Search Criteria:

Maximum Results Returned:

Number of rows displayed per page:

Apply Reset Clear Close

To create a custom query and search for tasks:

1. In the Task List page, click Customize Table Header.
 In the Construct a Custom Query form, specify the search criteria by specifying any of the following values.
2. Enter the task name. You can use the * character at the start or the end of the name.
3. Enter the task ID. Separate multiple values using comma.
4. Select the check box against the Administration and Working state of the task.

5. Enter any comment. You can use regular expression.
6. Enter the priority range.
7. Enter the Assignee, Owner, or Claimant names.
8. Specify the completion due date range.
9. If required, select the Include tasks with no completion due date check box.
10. Click Apply.

All tasks that match the specified criteria are displayed in the Task List page.

To customize the table header:

1. In the Task List page, click Customize Table Header. The Construct a Custom Query form appears.
2. In the View form, select the columns that you want to view in the Task List page.
3. Select the Preserve Search Criteria check box to retain these details for the current session.
4. Select the maximum number of search results that should be displayed on the page.
5. Select the number of rows that should be displayed in the page.
6. Click Apply.

The view in the Task List page is updated based on the current settings.

Deleting a Task Instance

You can delete a task instance from Worklist Console. For information, see “Deleting Tasks” in [Worklist Administration](#) in *Using Worklist Console*.

Customizing Worklist User Portal

Worklist provides Worklist User Portal, a web-based user portal for interacting with tasks. This portal allows users to see the details of a task, and take actions on a task to move it through its life cycle. Each step in a task plan can define unique actions and thus present its own unique considerations to the user.

Worklist User Portal is an all-purpose user interface that uses the metadata for steps and actions to render a usable interface at each step. The user interface for a given step and action will differ from the user interface for other steps and actions by virtue of the differences in the metadata for those steps and actions. For example, when rendering the user interface at a given step, the Worklist User Portal shows the description and actions the task plan designer defined for that step. Worklist User Portal has only this information to work on when rendering the user interface, and cannot account for any more subtle needs a user may have with regard to this step (For example, summary information that would be needed to enter a value for an action's required properties, or calculations that may be applied to individual task properties to derive the result value for a property)

In some cases, like the previous examples, a given step or the actions that can be taken on that step may require a customized user interface. Worklist accounts for this need for extensibility by allowing a web developer to design a custom web user interface to be used for a given step, or an entire task plan. This custom web user interface is called a Custom Task User Interface. Custom Task UI allows a web developer to design custom pages only where needed, and still use the facilities of the default Worklist User Portal everywhere else. This allows a gradual investment in the customization of the user portal, instead of requiring a wholesale replacement of the Worklist User Portal interface.

A custom interface can be designed for just a step in a task plan or for the entire task plan.

This document provides the following information:

- [Typical Workflow](#)
- [Implementing the Custom Task UI Page Flow](#)
- [Creating a Page Flow](#)
- [Creating Form Beans](#)
- [Defining Actions on a Page Flow](#)
- [Defining JSP Pages](#)
- [Registering the Custom Task UI](#)
- [Deploying the Custom Task UI](#)

Typical Workflow

Typical workflow in creating a custom task user interface:

1. Mock-up the web pages you want to show the user at the steps for which the custom task UI will be registered. Define the flow from page to page, and the links and buttons that connect the pages together. Also define the links or buttons that represent step actions that can be taken on this step.
2. Create a Beehive NetUI Page Flow Controller in the web application that hosts the Worklist user portal.
3. Define a data structure (called a Form Bean) for each page you mocked up in step 1.
4. Define actions on the Page Flow Controller from step 2 that represent the transitions between pages, and the step actions that can be taken.
5. Define the JSP pages for the mocked-up web pages you defined in step 1. Each page will be *bound* to the Form Bean you defined for the page in step 3, and refer to the actions you defined in step 4.

For a sample scenario, see [Adding a Customized User Interface](#) in *Tutorial: Building a Worklist Application*.

Implementing the Custom Task UI Page Flow

The custom task UI page flow can be implemented in one of two ways:

- By extending the custom controller with `TaskUIPageFlowController`.

Note: This is the preferred method for implementing the custom task UI page flow and this is the method that is assumed in the rest of this section. For information about the various public helper methods available on the controller, see the Javadoc of the controller available at the following location:

<http://edocs.bea.com/wli/docs92/worklist.javadoc/index.html>

- By extending custom controller with the basic `PageFlowController`. For requirements that must be met when you use this controller, see “[Task UI Page Flow Requirements](#)” on [page 3-3](#).

Task UI Page Flow Requirements

If you do not extend the custom controller using `TaskUIPageFlowController`, you must manually meet the following requirements of a task UI `PageFlowController`. These requirements allow a calling task work page flow to call into the task UI page flow and have that page flow return to it via actions the caller is guaranteed to have implemented.

- May optionally extend `com.bea.wli.worklist.TaskUIPageFlowController` to provide common utility method implementations.

- Must be marked as a *nested* page flow by setting the

```
@Jpf.Controller( nested=true )
```

annotation on the page flow controller class

- Must implement a begin action taking no parameters. If the page flow controller extends `TaskUIPageFlowController`, this method must call the `super.beginActionHelper()` method to correctly initialize its state. Otherwise, this method should obtain an instance of `TaskUIContextInfo` to provide context information about the task work controller. This can be obtained from the HTTP session as follows:

```
import com.bea.wli.worklist.WorklistUIContextHelper;
WorklistUIContextHelper helper =
    new WorklistUIContextHelper(getSession());
TaskUIContextInfo contextInfo = helper.getTaskUIContextInfo();
```

- Must provide a user interface component that allows the user to take the `done` action for the custom task UI. This will end the user's session in the custom task UI. What it means to leave a session depends on what the host application that contains the Worklist UI has defined for the lifetime of the session. For example, in Worklist User Portal, leaving a

session means to leave the “Work on Task” page and return to the “Home” page of the portal. The UI component for this action (for example, button or link) should use the `done` action context and label text provided in the `TaskUIContextInfo`.

- Must return to the calling page flow under the following circumstances, and using the indicated `exit/return` action on the calling page flow.
 - After taking any action on the current step of the task, the taskUI page flow must return from an internal `return` action using the `taskUIStepDoneAction` on the caller. An example `Forward` annotation is given below:

```
@Jpf.Forward(name = "success",
returnAction = "taskUIStepDoneAction")
```

- When the taskUI user decides to leave the UI session (by clicking the required `done` action UI component), the taskUI page flow must return from an internal `return` action using the `taskUISessionDoneAction` on the caller. What it means to leave a session depends on what the host application that contains the Worklist UI has defined for the lifetime of the session. For example, leaving a session might mean to logout. An example `Forward` annotation is given below:

```
@Jpf.Forward(name = "success",
returnAction = "taskUISessionDoneAction")
```

- If the user selects a task view (from the list of task views provided in the `TaskUIContextInfo` instance passed into the `begin` action), the taskUI must return from an internal `return` action using the `taskUIShowTaskListAction` on the caller. An example `Forward` annotation is given below:

```
@Jpf.Forward(name = "success",
returnAction = " taskUIShowTaskListAction")
```

where the returned `Forward` instance must have the selected task list view name (as a `String`) set on it. An example internal action and `Forward` construction statement is given below:

```
@Jpf.Action(forwards = {
    @Jpf.Forward(name = "success",
        returnAction="taskUIShowTaskListAction")
})
protected Forward showTaskListAction()
{
    // JSP should pass the view name from the link/button
    // etc. the user clicks on the page
```

```
String view = getRequest().getParameter("view");  
Forward success = new Forward("success", view);  
return success;  
}
```

Creating a Page Flow

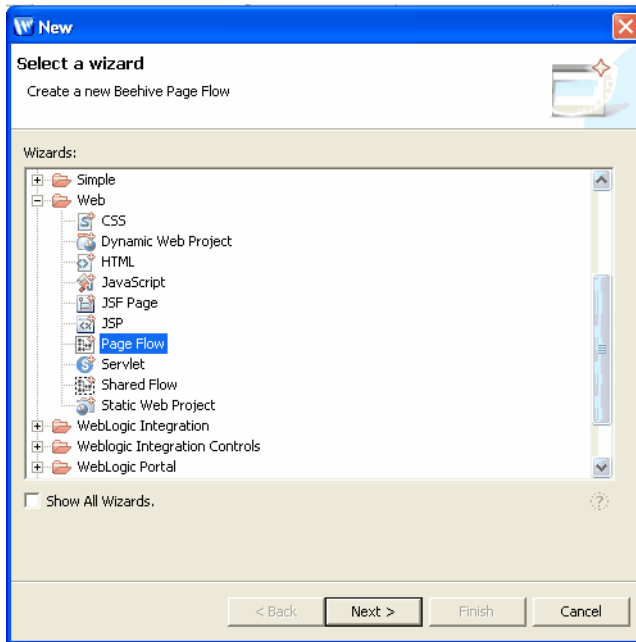
After determining the look and feel of the customized user interface pages, you need to define the logic and use of these pages by creating a Beehive Page Flow. Page Flows allow the design of an user interface on which actions can be defined. Page Flows control the actions and navigation of the web pages in the custom task interface.

To create a page flow:

1. In the Package Explorer pane, right-click the `<web project name>\src` folder, and select New → Other. The Select a Wizard dialog appears.

The Web project corresponds to the Web application of Worklist that acts as the user interface for the system. For more information, see “Worklist Application” in [Creating and Managing Worklist Task Plans](#) in *Using Worklist*.

Figure 3-1 Creating a Page Flow




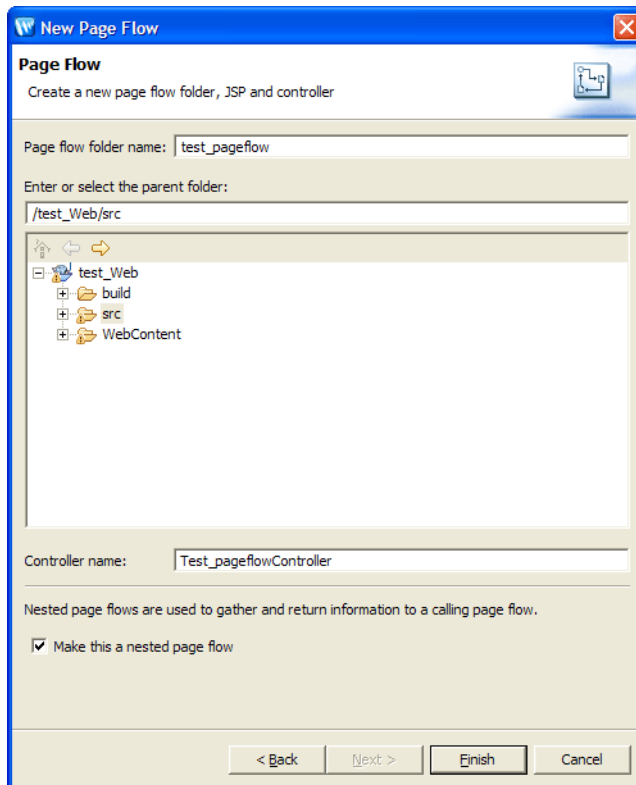
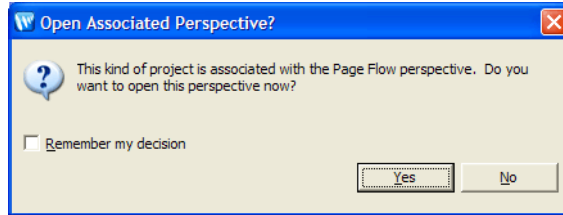
2. Click  next to Web, select Page Flow, and click Next. The New Page Flow dialog appears.

Figure 3-2 Specify Page Flow Details

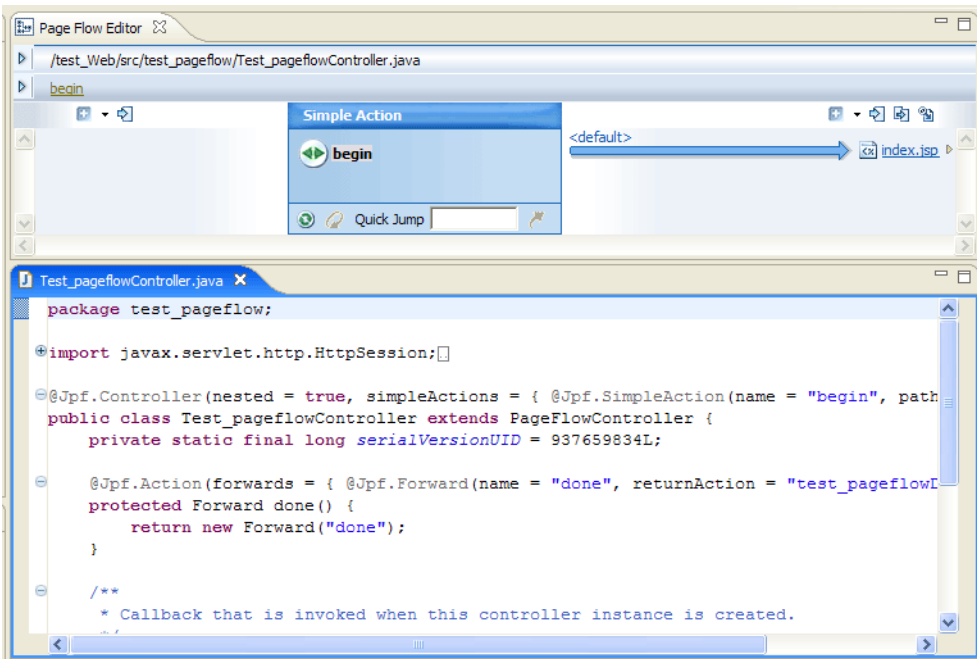
3. Enter the name of the Page Flow folder.
4. Specify the parent folder of the Page Flow.
5. Enter the name of the Page Flow controller.
6. Select the **Make this a nested page flow** check box and click Finish.
7. The Open Associated Perspective dialog appears. Click Yes.

Figure 3-3 Open Associated Perspective



The Page Flow Editor appears.

Figure 3-4 Page Flow Editor



8. Replace the following code in the <page flow controller>.java file

```
@Jpf.Controller(nested = true, simpleActions = { @Jpf.SimpleAction(name = "begin", path = "index.jsp") })
```

```
public class <page flow controller> extends PageFlowController
```

with

```
@Jpf.Controller(nested = true)

public class <page flow controller> extends
com.bea.wli.worklist.TaskUIPageFlowController {
```

Note: This change will result in some compilation errors saying ‘Action "begin" was not found.’ This error will be resolved after you define the begin action. For information about the begin action, see [“Defining Actions on a Page Flow” on page 3-11](#). To see a list of errors, warnings and information messages in the Problems View, select Window→Show View→Problems.

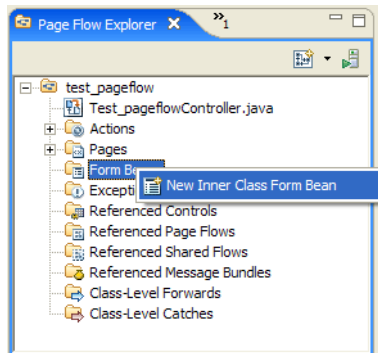
Creating Form Beans

For each web page that you need in the custom task UI, you need to create a Form Bean. These form beans can be represented as inner java classes of the Page Flow Controller, or they can be standalone java classes. Form Beans should implement `java.io.Serializable` as described in the JDK Javadoc.

To create form beans:

1. In the Page Flow Explorer, right-click Form Beans and select New Inner Class Form Bean.

Figure 3-5 New Inner Class Form Bean



2. A new form bean with the default name NewFormBean is created.
3. Right-click NewFormBean and select Rename to give a more meaningful name to the Form Bean.
4. Repeat [step 1](#) to [step 3](#) to create Form Beans for all the web pages required for your custom task UI.

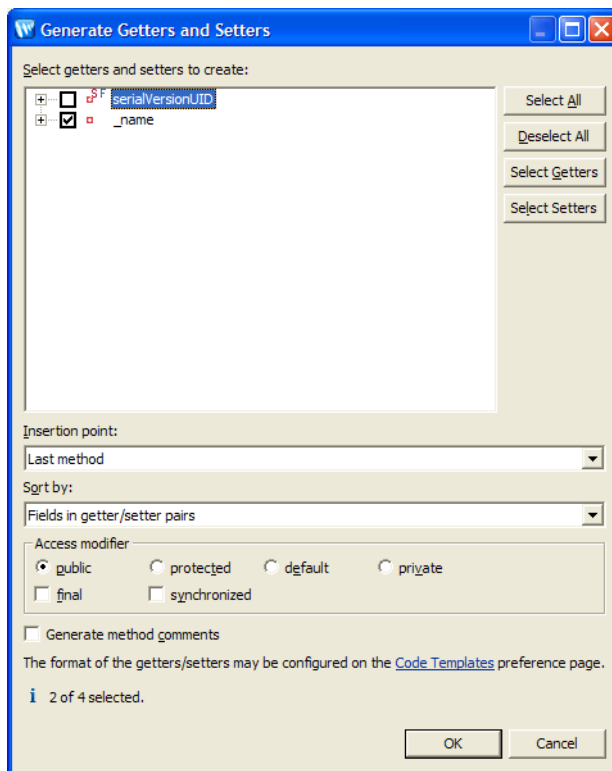
Generating Setter and Getters Methods for Properties

Form Beans are the basis for the data handling done in the web pages for the custom task UI. The Form Bean is a JavaBean object with a number of properties. You need to generate the Setter and Getter methods for each of these properties.

To generate the setter and getter methods:

1. Double-click on the Form Bean in the Page Flow Explorer View.
2. Declare the variables in the Form Bean.
3. Select the variables and right-click. Select Source → Generate Getters and Setters. The Generate Getters and Setters dialog appears.

Figure 3-6 Generate Getter and Setter Methods



4. Select the variables and click OK.

5. Repeat the above steps for properties in all the Form Beans required for your custom task UI. The Getter and Setter methods for all properties are now generated.

Defining Actions on a Page Flow

You define actions on the page flow to move between the web pages in your custom task interface and to take appropriate actions on the task. Page Flow actions are methods on the Page Flow Controller that allow navigation within the interface and optionally, calculate results and pass Form Beans to the required web pages.

Form Beans are passed from an action to a web page to populate display fields on the page. These field values are collected and placed into properties on the Form Bean when the web page is submitted back to the server for processing.

Action methods can accept a Form Bean populated as the result of clicking a Submit button on a web page by defining the Form Bean as a parameter to the action method. Action methods can also pass a Form Bean on to a target web page to which the action is forwarding. This is done by passing a Forward Object that has a Form Bean Object set on it.

Define page flow actions for all actions that are designed and required for your custom task interface. In addition, you need to define the `begin` action for initializing the Page Flow Controller. The `begin` action initializes this controller and calls the super class helper (`beginActionhelper`) to initialize task context, standard form beans for a task and action, and property editing support.

Note: `begin()` has to call the super class helper only if the custom page controller extends `TaskUIPageFlowController`.

Action Methods and Form Beans

Action methods can accept Form Beans and forward to pages using Form Beans. When submitting a web page and in the process of calling the action associated with a Submit, the NetUI framework creates a new form bean instance by default (using the no-arg public constructor for the Form Bean class). This new bean instance is then populated via Java reflection with data from data binding tags in the submitted web page form.

This process has some limitations. For example, if your Form Bean contains transient, hidden information that is not represented in the web page JSP tags, the Form Bean that actually gets passed to the action method (the bean that is created by the NetUI framework) will not have this information.

To avoid the overhead and possible behavioral problems of creating new beans each time an action method is called, you can specify a `useFormBean` field on the `@Jpf.Action` annotation for an `action` method. This allows to hold a single copy of the Form Bean in the Page Flow Controller's state and the action method then just fetches the object from that state instead of creating a new form bean object.

Worklist Property Editors and Actions

Worklist provides some built-in support for editing properties in your custom task UI. It includes a JSP tag, default editors, and some helper methods in the base `TaskUIPageFlowController`. These facilities allow you to easily edit the following types of properties using out-of-box UI:

- Multi-line/multi-page text
- JavaBean/XMLBean objects

In addition, the Property Editor facility allows human users using the custom task UI to easily edit properties using an inline editor (simple form field) as well as a standalone editor for the complex types mentioned above. This facility makes robust editing of properties a fairly simple matter.

Create the following actions in your custom page flow controller to handle task's user property editor:

- `okPropAction` – The stand-alone editor returns to this action when you click OK to apply the edit. It returns on this action passing an `EditorValueHolder` holding the value that was created or edited in the editor. This action should specify the `_editorValue` field of your Page Flow in a `useFormBean` attribute in the `@Jpf.Action` annotation. This allows the returned or completed editor value form bean to be passed in the `_editorValue` field and avoids creating a copy of this potentially large form bean. For an example of this action and its annotations, see [Tutorial: Building a Worklist Application](#)
- `cancelPropAction` – This is the action the stand-alone editor calls when the user clicks Cancel in the editor.
- `<edit property action>` – This action handles an initiate stand-alone editor call that comes from the JSP and the `worklist propertyEditor` tag. Your action method should call the `TaskUIPageFlowController.editPropActionHelper` to calculate the stand-alone editor's URI, and then forwards to that URI. This editor is a nested page flow, and returns to this controller (the caller) via the `okPropAction` and `cancelPropAction` return actions.

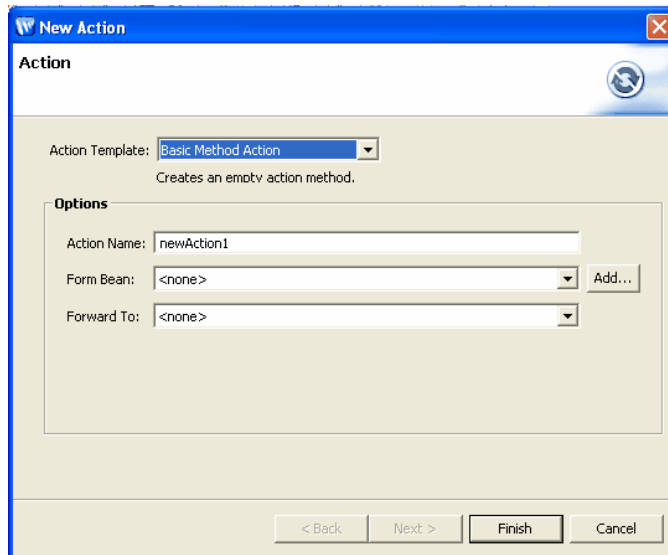
Note: The actual name of the action is specified in the Worklist `propertyEditor` tag in the `actionName` attribute.

Creating Actions on a Page Flow

To create actions on a page flow:

1. In Page Flow Explorer View, right-click Actions and select New Action. The New Action dialog appears.

Figure 3-7 Create New Action



2. Select any existing action templates. For example, create Basic Method Action to create an empty action method.
3. Enter a name for the action.
4. Specify the Form Bean associated with this action. If required, click Add to select a Form Bean.
5. If required, specify the Form Bean to which this action should be forwarded.
6. Click Finish.
7. Repeat the above steps for all the actions required for the Page Flow.

All the actions for the Page Flow are created. You should see code that is similar to the following:

```
@Jpf.Action()
public Forward
newAction1(test_pageflow.Test_pageflowController.NewFormBean form) {
    Forward forward = new Forward("success");
    return forward;
}
```

Defining JSP Pages

Based on the interface mock ups that you designed, define JSP pages for the custom task UI using Beehive NetUI data binding JSP tags to render web forms that can read data from and write data into form beans. The use of these tags greatly simplifies the process of writing a data-driven JSP page.

Creating JSP Files

You can create JSP files in the Page Flow perspective as follows:

1. In the Page Flow Explorer, right-click Pages and select New JSP Page.

Note: In the Page Flow Controller, if there is a reference to a unavailable JSP, grayed out JSP pages are automatically listed under the Pages node in the Page Flow Explorer View. You can right-click on these JSPs and select Create to create the JSP file.

2. Open the JSP file to view the following code:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<%@taglib uri="http://beehive.apache.org/netui/tags-html-1.0"
prefix="netui"%>
<%@taglib uri="http://beehive.apache.org/netui/tags-databinding-1.0"
prefix="netui-data"%>
<%@taglib uri="http://beehive.apache.org/netui/tags-template-1.0"
prefix="netui-template"%>
<netui:html>
    <head>
```

```
        <netui:base/>
    </head>
    <netui:body>
        <p>Beehive NetUI JavaServer Page -
        ${pageContext.request.requestURI}</p>

    </netui:body>
</netui:html>
```

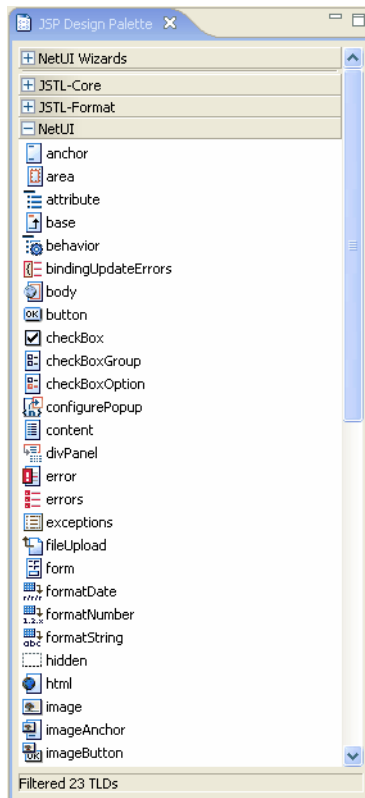
3. To create the JSP page, delete the following code from the JSP:

```
<p>Beehive NetUI JavaServer Page - ${pageContext.request.requestURI}</p>
```

and insert the text form in between the `<netui:body>` and the `</netui:body>` tag.

4. To design the interface that should appear on the web page, drag and drop required elements from the NetUI folder in JSP Design Palette View into the JSP.

Figure 3-8 JSP Design Palette



5. Use the NetUI form element to design the form for all the required actions.
6. Use the NetUI button element to enable navigation between forms and to take actions on the task.
7. Repeat the above steps for all the JSP pages required for your custom task UI.

The JSP pages are created.

Registering the Custom Task UI

After designing the custom task UI for a step in a task plan or for an entire task plan, you need to register the custom task UI. To register the custom task UI, you need to add mapping entries to `<Web project name>/WebContent/WEB-INF/task-ui-registry.xml` registry file.

Note: The Web project corresponds to the Web application of Worklist that acts as the user interface for the system. For more information, see “Worklist Application” in [Creating and Managing Worklist Task Plans](#) in *Using Worklist*.

This XML registry file is associated with the schema for the Worklist Task UI Registry, and this schema is registered with Workshop.

Open the file to edit it. In the Source tab, the initial contents of this XML file should look something like this:

```
<task-ui-registry xmlns="http://www.bea.com/wli/worklist/taskuiregistry">
</task-ui-registry>
```

In the Design tab, you can right-click any node in the tree view to act on it. You can delete nodes, and add children to nodes.

To register the custom task UI, you need the following information:

- **Task Plan ID** – The ID of the task plan for which the custom task UI applies, in external format (example, `/<path>/<task plan name>:<version>`)
- **Step Name** – Required only when registering the custom task UI for a specific step in the task plan.
- **Custom Task UI URI** – The web URI of the Page Flow Controller that will control the UI to be applied to this step or task plan.
- The URI has a `.jspx` extension even though the Page Flow Controller file is really `.java` file. This is needed to allow servlet filters in the web application to fire correctly.

Note: Step name is not required if you are registering the custom task UI designed for the entire task plan and not just a step in the task plan.

To edit `task-ui-registry.xml` first switch to the Package Explorer view and then do the following in the Design tab of the editor:

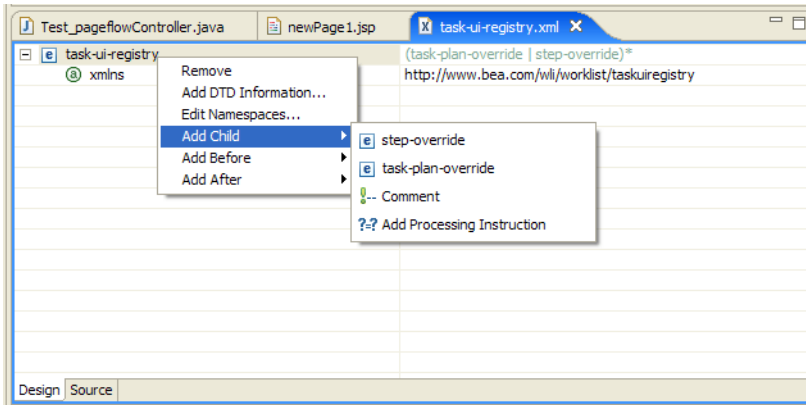
1. In the Package Explorer, navigate to the `<Web project name>/WebContent/WEB-INF/` folder and select `task-ui-registry.xml`.

Note: The Web project corresponds to the Web application of Worklist that acts as the user interface for the system. For more information, see “Worklist Application” in [Creating and Managing Worklist Task Plans](#) in *Using Worklist*.

2. Double-click the file to open it in the XML Editor.

- In the editor, right-click `task-ui-registry.xml`, and select `Add Child`. Select `step-override` or `task-plan override` based on whether you created the custom task UI for a step or the entire task plan. This adds a new override element under the root element.

Figure 3-9 Adding Child Override Element

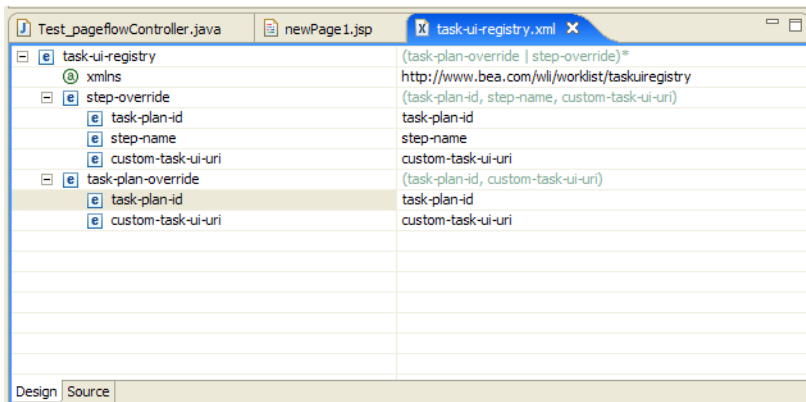


- Expand the newly added `step-override` or `task-plan-override` element and specify the task plan ID, step name, and custom task UI URI.

Note: Step name is not required if you are registering the custom task UI designed for the entire task plan and not just a step in the task plan.

- For each of these elements, click on the right-hand value column, and replace the default values with the actual values.

Figure 3-10 Map entries in registry file



Deploying the Custom Task UI

To deploy the custom task user interface:

1. On the Package Explorer View, select and right-click <web project name>.

Note: The Web project corresponds to the Web application of Worklist that acts as the user interface for the system. For more information, see “Worklist Application” in [Creating and Managing Worklist Task Plans](#) in *Using Worklist*.

2. Click Run As→Run On Server. The Define a New Server dialog appears.
3. Accept the default settings and click Next.
4. Browse and select the required domain .
5. Click Finish.

The custom Task UI is deployed on the server.

Customizing Worklist User Portal