



Project WebSynergy Milestone 4 Developer's Guide

Beta



Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 820-6299

Copyright 2008 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, JDK, JRE, NetBeans, MySQL, GlassFish, OpenSolaris, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. or its subsidiaries in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2008 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux Etats-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certains composants de ce produit peuvent être dérivées du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc., ou ses filiales, aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

Contents

Preface	5
1 Project WebSynergy Technical Overview	9
Architectural Overview	9
Packaging Overview	9
Development Tools	9
2 Development Tasks	11
Developing Portlets Using NetBeans Portal Pack 2.0 in Project WebSynergy	11
Developing Portlets Using Eclipse Portal Pack in Project WebSynergy	12
Installing the Eclipse Portal Pack	12
Using the Eclipse Portal Pack	13
Creating and Deploying a Portlet	14
Developing Services in Project WebSynergy	15
Overview of Services	16
Implementing Services	16
Using Existing Project WebSynergy Services	16
Developing Workflows Using Simple API for Workflow in Project WebSynergy	16
Introduction to SAW	16
Prerequisites	17
Adding SAW plugin in NetBeans IDE	17
Creating a Web Application Using the Drag and Drop Features in SAW plugin	18
Creating a SAW Portlet Using the SAW plugin	20
Deploying the Workflow Portlet an Open Source Portlet Container	21
Deploying the Workflow Portlet on Project WebSynergy	23
Using the Workflow Portlet	25
References	27

- Developing Themes Using ViewDesigner Plugin in Project WebSynergy 27
 - Overview of ViewDesigner Plugin 27
 - Using Desktop DDT and ViewDesigner Plugin 28
- 3 API Reference33**
 - API Reference Section 33
- 4 Sample Code35**
 - Sample Code Section 35

Preface

To be documented

Who Should Use This Book

To be documented

To be documented

- To be documented
- To be documented
- To be documented

Before You Read This Book

To be documented

To be documented

- To be documented
- To be documented
- To be documented

To be documented

How This Book Is Organized

To be documented

To be documented

To be documented

Related Books

To be documented

- To be documented
- To be documented
- To be documented

Related Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- [Documentation](http://www.sun.com/documentation/) (<http://www.sun.com/documentation/>)
- [Support](http://www.sun.com/support/) (<http://www.sun.com/support/>)
- [Training](http://www.sun.com/training/) (<http://www.sun.com/training/>)

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P-1 Typographic Conventions

Typeface	Meaning	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name%</code> you have mail.

TABLE P-1 Typographic Conventions (Continued)

Typeface	Meaning	Example
AaBbCc123	What you type, contrasted with onscreen computer output	machine_name% su Password:
<i>aabbcc123</i>	Placeholder: replace with a real name or value	The command to remove a file is <i>rm filename</i> .
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . A <i>cache</i> is a copy that is stored locally. Do <i>not</i> save the file. Note: Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX® system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P-2 Shell Prompts

Shell	Prompt
C shell	machine_name%
C shell for superuser	machine_name#
Bourne shell and Korn shell	\$
Bourne shell and Korn shell for superuser	#

Project WebSynergy Technical Overview

To be documented

Architectural Overview

To be documented

Packaging Overview

To be documented

Development Tools

To be documented

Development Tasks

This chapter explains how you can use certain plugins available as part of Project WebSynergy to develop portlets. This chapter includes the following sections:

- [“Developing Portlets Using NetBeans Portal Pack 2.0 in Project WebSynergy” on page 11](#)
- [“Developing Portlets Using Eclipse Portal Pack in Project WebSynergy” on page 12](#)
- [“Developing Services in Project WebSynergy” on page 15](#)
- [“Developing Workflows Using Simple API for Workflow in Project WebSynergy” on page 16](#)
- [“Developing Themes Using ViewDesigner Plugin in Project WebSynergy” on page 27](#)

Developing Portlets Using NetBeans Portal Pack 2.0 in Project WebSynergy

The Portal Pack provides a set of plugins for NetBeans IDE to support the portlet application development life-cycle of within NetBeans IDE. Using these set of plugins, developers can develop, package, deploy, and test portlets within the NetBeans IDE. By automatic code and Deployment Descriptor generation, the Portal Pack helps developers to develop portlets quickly. Portal Pack plugins provide a tight integration with portal servers such as the WebSynergy Server, Liferay Portal Server, Sun Java System Portal Server 7.x, and OpenPortal Portlet Container on `java.net` to support deployment and undeployment of portlets on both the local and remote servers.

For more information on how to develop portlets using Portal Pack 2.0 on NetBeans IDE, see [Using Portal Pack 3.0 Plugins to Create Portlets on NetBeans 6.5 IDE](#).

Developing Portlets Using Eclipse Portal Pack in Project WebSynergy

The Eclipse Portal Pack provides a set of plugins that help you to develop JSR 168/286 Portlets and deploy them on supported portlet containers.

There are three plugins available in the Eclipse Portal Pack.

- **JSR 168/286 Portlet Builder** — This plugin allows you to create a JSR 168/286 portlet project using the wizards available in the Eclipse IDE. Once you create a portlet project using the wizards, the Eclipse Portal Pack plugin creates the appropriate class files and updates the `portlet.xml` file. In addition, the plugin provides a `build.xml` file that allows you to build a WAR, which you can deploy on any portal.
- **WebSynergy** — This plugin registers the Project WebSynergy that runs on GlassFish server in Eclipse IDE, as a Server Runtime. Currently, the WebSynergy plugin allows you to start and stop the Project WebSynergy (GlassFish server) from within the IDE using the generic server framework in Eclipse.
- **OpenPortal Portlet Container** — This plugin registers the OpenPortal Portlet Container 2.0 that runs on GlassFish server in Eclipse IDE, as a Server Runtime. Currently, the OpenPortal Portlet Container plugin allows you to start and stop the Portlet Container (GlassFish server) from within the IDE using the generic server framework in Eclipse.

Installing the Eclipse Portal Pack

You need to install the Eclipse Portal Pack to use the plugins and develop and deploy portlets.

▼ To Install the Eclipse Portal Pack

- 1 **Download Eclipse Portal Pack from the Eclipse Portal Pack [download page](#).**
- 2 **Unzip the Eclipse Portal Pack.**
- 3 **Copy the `.jar` files to the `ECLIPSE_HOME\plugins` directory.**
- 4 **(Optional) — Restart Eclipse, only if it is running while you copy the `.jar` files.**

Using the Eclipse Portal Pack

This section explains how to use the Eclipse Portal Pack and the plugins in Project WebSynergy. You need to perform the following tasks before using the Eclipse Portal Pack.

1. Creating Runtime and Server for Project WebSynergy
2. Creating a Runtime and Server for OpenPortal Portlet Container 2.0

Creating a Runtime and Server for Project WebSynergy

You need to create a runtime and server for Eclipse Portal Pack in order to deploy it on Project WebSynergy.

▼ To Create a Runtime and Server for Project WebSynergy

- 1 On Eclipse IDE, click the Window menu and select Preferences.**
The Preferences window appears.
- 2 From the left pane, click Installed Runtimes.**
The Installed Server Runtime Environments appear on the right pane.
- 3 Click Add.**
The New Server Runtime window appears. In this window, you can choose the runtime environment for a newly installed server.
- 4 From the available runtime types, expand Sun Microsystems Inc and select WebSynergy. Select the Also create new local server check box.**
- 5 Click Next.**
- 6 Enter the JRE name, GlassFish Home directory, and the Domain Name information for Project WebSynergy.**
These details are useful to create a runtime for Project WebSynergy.
- 7 Click Next.**
- 8 Enter the server address, server port, and the debug port information for Project WebSynergy.**
- 9 Click Finish.**
The Project WebSynergy server is created.

Creating a Runtime and Server for OpenPortal Portlet Container 2.0

You need to create a runtime and server for Eclipse Portal Pack in order to deploy it on OpenPortal Portlet Container 2.0.

▼ To Create a Runtime and Server for OpenPortal Portlet Container 2.0

- 1 **On Eclipse IDE, click the Window menu and select Preferences.**

The Preferences window appears.

- 2 **From the left pane, click Installed Runtimes.**

The Installed Server Runtime Environments appear on the right pane.

- 3 **Click Add.**

The New Server Runtime window appears. In this window, you can choose the runtime environment for a newly installed server.

- 4 **From the available runtime types, expand Sun Microsystems Inc and select OpenPortal Portlet Container 2.0. Select the Also create new local server check box.**

- 5 **Click Next.**

- 6 **Enter the JRE name, GlassFish Home directory, and the Domain Name information for OpenPortal Portlet Container 2.0.**

These details are useful to create a runtime for OpenPortal Portlet Container 2.0.

- 7 **Click Next.**

- 8 **Enter the server address, server port, and the debug port information for OpenPortal Portlet Container 2.0.**

- 9 **Click Finish.**

The OpenPortal Portlet Container 2.0 server is created.

Creating and Deploying a Portlet

You can use the plugins in Eclipse Portal Pack to create and deploy portlets on Project WebSynergy. This section explains how to create and deploy portlets on Project WebSynergy using the Eclipse Portal Pack plugins.

▼ To Create and Deploy a Portlet

- 1 **On Eclipse IDE, click the File menu and select New. Form New, choose Dynamic Web Project.**

The New Dynamic Web Project window appears.

- 2 **In the New Dynamic Web Project window, enter a name for your project, choose WebSynergy or OpenPortal Portlet Container 2.0 as project runtime, and select Portlet 1.0 or 2.0 support as the configuration.**

Note – If Portlet 2.0 configuration is not available in the list, you need to manually add the configuration. To do this, click Modify, select Portal Pack and click Save.

- 3 **Click Finish.**

A new web project is created.

- 4 **To add a portlet to the web project, click the File menu. From the options, select New and click Other.**

The New window to create a Java portlet appears.

- 5 **From the list of Wizards, expand the Portlet wizard and click Java Portlet.**

- 6 **Click Next.**

The Create a Portlet Class window appears.

- 7 **Enter the details of the portlet that you want to generate.**

- 8 **Click Finish.**

A portlet class is created and an entry about the portlet is created in the `portlet.xml` file.

- 9 **To deploy the portlet, right click on the name of the portlet, select Run As, and click Run On Server. Accept the default values.**

This action deploys the portlet web application and starts the target server, which is either WebSynergy or OpenPortal Portlet Container.

Developing Services in Project WebSynergy

To be documented

Overview of Services

To be documented

Implementing Services

To be documented

Using Existing Project WebSynergy Services

To be documented

Developing Workflows Using Simple API for Workflow in Project WebSynergy

This section explains about the Simple API for WorkFlow (SAW) feature and how you can develop workflow portlets using NetBeans™ 6.0 IDE or higher versions. The topics covered in this section are:

- Introduction to SAW
- Prerequisites
- Adding SAW plugin in NetBeans IDE
- Creating a Web Application Using the Drag and Drop Features in SAW plugin
- Creating a SAW Portlet Using the SAW plugin
- Deploying the Workflow Portlet an Open Source Portlet Container
- Deploying the Workflow Portlet on Project WebSynergy
- Using the Workflow Portlet
- References

Introduction to SAW

SAW is a generic workflow API to perform human workflow interaction with various workflow engines. SAW provides a framework to plug-in any business process specific implementation that implements SAW interfaces. SAW is shipped with a default implementation for Sun Java™ Composite Application Platform Suite. You can have similar SAW implementations for other business processes such as JBoss Business Process Management (jBPM) and Open Enterprise Service Bus (OpenESB).

Prerequisites

Before you build an application using SAW APIs, you must install and configure the following components:

- Java Development Kit (JDK™) 1.5 or later versions from <http://java.sun.com/javase/downloads/index.jsp>.
- Sun Java Composite Application Platform Suite. Currently, SAW implementation is available only for Sun Java Composite Application Platform Suite. Therefore, a Sun Java Composite Application Platform Suite installation is required. If you need any other OpenSource workflow engine implementation, visit <https://saw.dev.java.net/>, for regular updates.
- NetBeans 6.0 from <http://download.netbeans.org/netbeans/6.0/final/>.
- Portal Pack 2.0 Beta from <http://portalpack.netbeans.org/>
- OpenPortal Portlet Container 2.0 Beta from <https://portlet-container.dev.java.net/>
- SAW plug-in for NetBeans from <http://portalpack.netbeans.org/>

Before proceeding with the implementation of SAW in your application, you need to configure a business process using Sun Java Composite Application Platform Suite. For more information how to create a business process in Sun Java Composite Application Platform Suite, read the article <http://developers.sun.com/portalservers/reference/techart/workflow.html>

Adding SAW plugin in NetBeans IDE

This section provides you the procedure to add SAW plug-in in NetBeans IDE.

▼ To Add SAW plugin in NetBeans IDE

- 1 **Download and install NetBeans 6.0 Beta 1, which is available here**
<http://www.netbeans.org/community/releases/60/>
- 2 **Download and install SAW plug-in for NetBeans 6.0 IDE, which is available here**
<http://portalpack.netbeans.org/>. For more information on SAW project, visit <https://saw.dev.java.net/>.
- 3 **From the NetBeans 6.0 IDE, click Tools and select Plug-ins.**
The Plugins window appears.
- 4 **Click the Downloaded tab and select the installed SAW plug-in.**
- 5 **Click Add Plugins to add the selected SAW plug-in in NetBeans.**

6 Click Install.

The SAW plug-in is installed in NetBeans IDE.

Tip – You can click the Installed tab, to verify whether the SAW plug-in is installed and added to the list.

Creating a Web Application Using the Drag and Drop Features in SAW plugin

This section provides a procedure to explain how to create a simple web application in NetBeans using the drag and drop features available in SAW plug-in.

▼ To Create a Web Application Using the Drag and Drop Features in SAW plugin

- 1 Click New Project in NetBeans IDE.
- 2 Select Web from Categories and Web application from Projects and click Next.
- 3 Enter a project name.
- 4 From the Server drop down list, select GlassFish v2 and click Next.
- 5 From the Frameworks list box, select the SAW plug-in.

You can see that the Java Composite Application Platform Suite is selected by default, for the SAW implementation

6 Click Finish.

Tip – Check the SAW project web site <https://saw.dev.java.net/> regularly, for updates on these implementations.

- 7 **Expand the Source Packages folder to see the `ImplementationType.properties` and `WorkflowConfig.properties` property files, under the default package folder.**

The `ImplementationType.properties` file sets the `ImplementationType` as Java Composite Application Platform Suite. This can be changed in future saw releases to point to JBPM, OpenESB or other workflow Engines

8 Set the `WorkflowConfig.properties` file appropriately, to point the Java Composite Application Platform Suite Integration server.

The web application is listed in the Projects tab and the `index.jsp` appears in the code window. You can see the appropriate Workflow tags listed in the Palette.

Note – If the Workflow tags palette is not loaded in the NetBeans IDE, you can manually load it by clicking Tools and selecting Workflow from the Palette option. Moreover, if you right click on the web application and select Properties, the Libraries Category in the Project Properties window displays the SAW libraries that are added by the SAW plug-in in the NetBeans IDE. The current version of SAW libraries is 2.0. Check the SAW project web site <https://saw.dev.java.net/> regularly for updates on the plug-in for new versions of SAW.

9 Open the Java file in your web application.

You can notice that the relevant Workflow APIs are listed in the palette. You can drag and drop these Workflow APIs in the Java file, depending upon the usage for your logic. In the default package, you can see various properties required by SAW to execute. These properties are the location of Sun Java Composite Application Platform Suite, SAW implementation and so on.

Note – You cannot drag and drop duplicate methods. There are two ways to use the SAW APIs. One way is to use through JSP tag libraries and the other way is through the Java APIs.

10 Edit the `WorkflowConfig.Properties` file to define your runtime execution environment.

The web application is ready to deploy. The sample `WorkflowConfig.Properties` file is shown below:

```
# the business process to use.
sawworkflowimplclass = The SAW Workflow Implementation class to use. For example,
com.sun.saw.impls.jcaps.JCAPSWorkflow
# Properties that are needed by the JCAPS Implementation of SAW.
com.sun.saw.impls.jcaps.JCAPSWorkflow.appserverhost = machine name or IP where
Sun Java Composite Application Platform Suite business process is running.
For example, abc.india.sun.com
com.sun.saw.impls.jcaps.JCAPSWorkflow.iiopserverport =
Port on which the IIOP lookup for EJB happens. For example, 18002
com.sun.saw.impls.jcaps.JCAPSWorkflow.appserverusername =
Administrator user name. For example, Administrator
com.sun.saw.impls.jcaps.JCAPSWorkflow.appserverpassword =
Administrator password of JCAPS Integration server. For example, <password>
com.sun.saw.impls.jcaps.JCAPSWorkflow.contextfactory =
The context factory. For example, com.sun.jndi.cosnaming.CNCtxFactory
com.sun.saw.impls.jcaps.JCAPSWorkflow.serviceJndi =
The JNDI look up name. For example, WorkflowService
```



Caution – Do not edit the `ImplementationType.properties` file.

- 11 Right click on the created web application and select **Libraries**.
- 12 Click **Add Jar** and select the `WorkflowServiceClient.jar` from **Java Composite Application Platform Suite** and click **OK**.
- 13 Right click on the web application and select **Clean and Build**.
- 14 Right click on the web application and select **Run**.
A web browser is launched and it displays the user interface defined in your web application.

Creating a SAW Portlet Using the SAW plugin

This section explains you how to create a simple workflow portlet in NetBeans IDE using the SAW plug-in.

▼ To Create a SAW Portlet Using the SAW plugin

- 1 Click **New Project** in NetBeans IDE.
- 2 Select **Web** from **Categories** and **Web application** from **Projects** and click **Next**.
- 3 Enter a project name.
- 4 From the **Server** drop down list, select **Open Portal Portlet Container 2.0 Beta** and click **Next**.
- 5 From the **Frameworks** list box, select **Portlet Support** and select the **Create Portlet** checkbox.
- 6 Select the **SAW plug-in** and click **Finish**.
- 7 **Expand the Source Packages folder to see the `ImplementationType.properties` and `WorkflowConfig.properties` property files, under the default package folder.**
The `ImplementationType.properties` file sets the `ImplementationType` as **Java Composite Application Platform Suite**. This can be changed in future saw releases to point to **JBPM**, **OpenESB** or other workflow Engines
- 8 **Set the `WorkflowConfig.properties` file appropriately, to point the Java Composite Application Platform Suite Integration server.**

9 Edit the WorkflowConfig.Properties file to define your runtime execution environment.

The sample WorkflowConfig.Properties file is shown below:

```
# the business process to use.
sawworkflowimplclass = The SAW Workflow Implementation class to use.
For example, com.sun.saw.impls.jcaps.JCAPSWorkflow
# Properties that are needed by the JCAPS Implementation of SAW.
com.sun.saw.impls.jcaps.JCAPSWorkflow.appserverhost = machine name or IP
where Sun Java Composite Application Platform Suite business process is running.
For example, abc.india.sun.com
com.sun.saw.impls.jcaps.JCAPSWorkflow.iiopserverport =
Port on which the IIOP lookup for EJB happens. For example, 18002
com.sun.saw.impls.jcaps.JCAPSWorkflow.appserverusername =
Administrator user name. For example, Administrator
com.sun.saw.impls.jcaps.JCAPSWorkflow.appserverpassword =
Administrator password of JCAPS Integration server. For example, <password>
com.sun.saw.impls.jcaps.JCAPSWorkflow.contextfactory = The context factory.
For example, com.sun.jndi.cosnaming.CNCTXFactory
com.sun.saw.impls.jcaps.JCAPSWorkflow.serviceJndi = The JNDI look up name.
For example, WorkflowService
```

10 Change the portlet Java and the JSP files in your web application accordingly, to write the portlet.

You can notice that the relevant Workflow APIs are listed in the palette that you can drag and drop in the Java file, depending upon the usage for your logic.

11 Right click on the created portlet application and select Libraries.**12 Click Add Jar and select the WorkflowServiceClient.jar from Java Composite Application Platform Suite and click OK.****13 Right click on the web application and select Clean and Build.****14 Right click on the web application and select Run.**

The portlet gets rendered on the portlet container.

Deploying the Workflow Portlet an Open Source Portlet Container

You need to deploy the Workflow portlet in order to work with the Workflow API. This section provides a procedure to explain how to deploy the workflow portlet on Open Source Portlet Container.

▼ To Deploy the Workflow Portlet on Open Source Portlet Container

- 1 Download the workflow portlet from the SAW project web site <https://saw.dev.java.net/>.
- 2 Extract the workflowPortlet.war into a local directory using the `jar -xvf workflowPortlet.war` command.
- 3 Modify the WorkflowConfig.properties file in the \workflowPortlet\WEB-INF\classes directory by providing suitable values to the following properties:
 - sawworkflowimplclass = com.sun.saw.impls.jcaps.JCAPSWorkflow. This means that you are using SAW Workflow implementation class, JCAPSWorkflow.
 - com.sun.saw.impls.jcaps.JCAPSWorkflow.appserverhost = *machine name* or *IP address*, where Sun Java Composite Application Platform Suite business process is running. For example, *abc.india.sun.com*.
 - com.sun.saw.impls.jcaps.JCAPSWorkflow.iiopserverport = *port* on which the IIOP lookup for EJB happens. For example, *18002*.
 - com.sun.saw.impls.jcaps.JCAPSWorkflow.appserverusername = *Administrator user name*. For example, *Administrator*.
 - com.sun.saw.impls.jcaps.JCAPSWorkflow.appserverpassword = *Administrator password*. For example, *xxx*.
 - com.sun.saw.impls.jcaps.JCAPSWorkflow.contextfactory = *The context factory*. For example, *com.sun.jndi.cosnaming.CNCtxFactory*.
 - com.sun.saw.impls.jcaps.JCAPSWorkflow.serviceJndi = *The JNDI look up name*. For example, *WorkflowService*.
- 4 Modify the workflowportlet.properties file in \workflowPortlet\WEB-INF\classes directory by providing appropriate values to the properties. The sample workflowportlet.properties file to deploy on Portlet Container is shown below.
authenticationRepository = accessManager. Specify the authentication repository that is used.

Note – accessManager is the default value that is provided for authenticationRepository in the workflowportlet.properties file. Only if you want to deploy the .war file using the Open Source Portlet Container, then change the authenticationRepository value to appServer.

- 5 Ensure that you have the client stubs (WorkflowServiceClient.jar) generated out of the workflow service, deployed on Sun Java Composite Application Platform Suite. Copy this to \workflowPortlet\WEB-INF\lib directory by using the `cp WorkflowServiceClient.jar workflowServiceClient-1.0.jar` command.
- 6 Recreate the .war file using the `jar -cvf` command.

7 Deploy the Workflow portlet using the admin tab of the Portlet Server Administration Console.

To deploy the workflow portlet on Open Source Portlet Container:

- Create a user under admin-realm of GlassFish V2 or Application Server 9.1, on which the Open Source Portlet Container is running. For example, create a user, CPina in the admin-realm of GlassFish V2.
- Login to the GlassFish V2 admin.
- Navigate to Configuration -> Security -> Realms.
- Click admin-realm and Click Manage Users.
- Add CPina, group list : asadmin and password as CPina.
- Access the portlet by typing `http://machine:port/portal/dt` in a web browser.
- Log in as user, CPina. The tasks that are assigned to CPina will be displayed on the workflow portlet.

If it is a web application, then the `saw-api-0.6.jar` should be in the appropriate classpath. The `saw.tld`, `saw-impl-jcaps-0.6.jar`, and `WorkflowServiceClient.jar` should be bundled with the application.

8 Modify the `jreHome\lib\logging.properties` file. To do this, update the `java.util.logging.FileHandler.formatter=java.util.logging.SimpleFormatter` property. The logging levels can be configured here. For example, INFO, ERROR and so on.

Note – You need not modify the `logging.properties` file in the case of OpenPortal. For stand alone Java applications, you need to modify the `logging.properties` file to specify the type of format and the level of log.

Deploying the Workflow Portlet on Project WebSynergy

You need to deploy the Workflow portlet in order to work with the Workflow API. This section provides a procedure to explain how to deploy the workflow portlet on Open Portal 7.2.

▼ To Deploy the Workflow Portlet on Project WebSynergy

- 1 **Extract the `workflowPortlet.war` into a local directory using the `jar -xvf workflowPortlet.war` command.**

Note – The workflow portlet is available in the Project WebSynergy install under the /opt/sun/portal/portlet/ directory for Linux and /opt/SUNWportal/portlet/ directory for Solaris. You can also download the workflow portlet from the Portlet Repository <http://wiki.java.net/bin/view/OpenPortal/PortletsInTheRepository>.

- 2 **Ensure that you have the client stubs (WorkflowServiceClient.jar) generated out of the workflow service, deployed on Sun Java Composite Application Platform Suite. Copy this to \workflowPortlet\WEB-INF\lib directory by using the cp WorkflowServiceClient.jar workflowServiceClient-1.0.jar command.**
- 3 **Recreate the .war file using the jar -cvf command.**
- 4 **Deploy the Workflow portlet using the admin tab of the Portal Server Administration Console.**
To deploy the workflow portlet on Open Portal:
 - Create a channel with this portlet provider and add it to one of the tab
 - Click the created new channel created and change the following portlet preferences:
 - sawworkflowimplclass = com.sun.saw.impls.jcaps.JCAPSWorkflow. This means that you are using SAW Workflow implementation class, JCAPSWorkflow.
 - com.sun.saw.impls.jcaps.JCAPSWorkflow.appserverhost = machine name or IP address, where Sun Java Composite Application Platform Suite business process is running. For example, abc.india.sun.com.
 - com.sun.saw.impls.jcaps.JCAPSWorkflow.iiopserverport = port on which the IIOP lookup for EJB happens. For example, 18002.
 - com.sun.saw.impls.jcaps.JCAPSWorkflow.appserverusername = Administrator user name. For example, Administrator.
 - com.sun.saw.impls.jcaps.JCAPSWorkflow.appserverpassword = Administrator password. For example, xxx.
 - com.sun.saw.impls.jcaps.JCAPSWorkflow.contextfactory = The context factory. For example, com.sun.jndi.cosnaming.CNCtxFactory.
 - com.sun.saw.impls.jcaps.JCAPSWorkflow.serviceJndi = The JNDI look up name. For example, WorkflowService.

Note – accessManager is the default value that is provided for authenticationRepository in the workflowportlet.properties file. You need not change this property, if the portlet is deployed on Open Portal.

- 5 **Create a user with userId as CPina in Access Manager with the status as active. Assume that the Business process assigns the newly created tasks to this user.**

- 6 Access the portlet by typing `http://machine:port/portal/dt` in a web browser.**
- 7 Log in as user CPina. The tasks that are assigned to CPina will be displayed on the workflow portlet.**

Only in the case of Open Portal, the `saw-api-0.6.jar` should be available in the server classpath. If it is a web application, then the `saw-api-0.6.jar` should be in the appropriate classpath. The `saw.tld`, `saw-impl-jcaps-0.6.jar`, and `WorkflowServiceClient.jar` should be bundled with the application.

Using the Workflow Portlet

1. Login to the Portal Server 7.2 desktop (`http://portalmachine:port/portal`) with username and password as CPina.
2. As soon as the user logs in, all the tasks assigned to the user are displayed. The page contains three sections.
 - First section — Search Criteria section, where the user can mention the criteria based on which the tasks have to be filtered and displayed.
 - Second section — Task List section that displays all the tasks that match the search criteria of a user.
 - Third section — Displays the buttons, which are the operations that can be performed on any task.
3. The search criteria section has the following attributes using which the user can filter tasks:
 - Start Date and End date - Filter tasks based on a valid start date and end date.
 - TaskIds - If the user knows the taskids, then the user can enter them in the text area that is separated by commas.
 - UserIds - Filter tasks based on the current owner of the task.
 - Task Status - Filter tasks based on the status of the task. For example, pending or completed.
 - GroupId - Filter tasks based on the group id. For example, Vice President Marketing and so on.
 - Flex String Type - Filter tasks based on a flex string type. For example, `flexString1`, `flexInt1`, `flexDouble1` and so on.
 - Flex String Value - The value of the flex string type that is selected by the user.
4. When the user clicks the Search button, tasks are filtered based on the search criteria that the user has entered and displayed in the "Task List" section.
5. The "Clear" button clears any of the search criteria that the user has entered or selected.
6. Assume that a new task is allocated to the user "CPina". Then "CPina" can do the following operations on the task that is assigned:

- *EXECUTE* - Select the task and click the Execute button. A new page for executing a task is displayed. This page shows the input for the task in a text box and flex string in another text box. The user can update both the text boxes and click the Save button. These values get updated for the task. On executing, the task gets checked out on the name of the logged in user (CPina), who is the current owner of this task. Currently, the task can be executed only once. This is because, on execution, the task is checked out, and a task can be checked out only once by a given user. If the user tries to execute the second time, then "Task Exception" error message will be displayed.
 - *COMPLETE* - Select the task and click the Complete button. Now, the status of the task is marked as completed. The user has to be the current owner for performing this operation on this task.
 - *CHECKIN* - Select the task and click the CheckIn button. This operation is opposite of check out. The currently logged in user is not the current owner now. The user has to be the current owner for performing this operation on this task.
 - *HISTORY* - Select the task and click the History button. On clicking this, the task history is shown in a new JSP. All operations that were performed on the task is displayed. Click the Back button on the task history JSP to come back to the task list page.
 - *ESCALATE* - Select the task and click the Escalate button. On clicking this, the task gets assigned to the manager of the currently logged in user. The user has to be the current owner for performing this operation on this task.
 - *REASSIGN* - Select users from the Reassign selected task to drop down, and click the Reassign button, the task gets assigned to these users. The user has to be the current owner for performing this operation on this task.
 - *DELETE* - Select the task and click the Delete button. On this operation, the task is deleted. The user has to be the current owner for performing this operation on this task.
7. Pagination support is provided for the Task List. On the first page the Previous link is not displayed. On the last page, the Next link is not displayed. On all the other pages, both the Previous and Next link is displayed.
 8. Any Error messages are displayed in the top of the portlet in red font.
 9. Any confirmation messages are displayed in the top of the page in blue font.
 10. The attributes that can be modified using the portlet preferences are the noOf records and the executeJSP. You need to click the edit mode to change the values for the portlet preferences.
 11. All the messages that are displayed are internationalized.

References

For more information on SAW implementation, refer the following resources:

- For regular updates on SAW project, visit the SAW project web site <https://saw.dev.java.net/>.
- For more information on how to incorporate workflow in Sun Portal Server with Sun Java Composite Application Platform Suite, read the article <http://developers.sun.com/portalserver/reference/techart/workflow.html>.
- For more information on how to work with SAW in NetBeans 6.0 IDE, check this [screencast](#).
- Download JavaDocs for SAW API from here <https://saw.dev.java.net/docs/SAWDownloadPage.html>.
- For regular updates on Portal Pack project, visit the Portal Pack project web site <http://portalpack.netbeans.org/>.
- For regular updates on OpenPortal Portlet Container project, visit the Portlet Container project web site <https://portlet-container.dev.java.net/>.
- For more information on NetBeans and the most recent downloads available, visit the NetBeans web site <http://www.netbeans.org/>.

Developing Themes Using ViewDesigner Plugin in Project WebSynergy

The ViewDesigner plugin enables a web designer to design and customize a theme of a portal page in Project WebSynergy. You can download the plugin from the ViewDesigner project website. The topics covered in this section are:

- Overview of ViewDesigner Plugin
- Using Desktop DDT and ViewDesigner Plugin

Overview of ViewDesigner Plugin

The layout and design of any web or portal page(s) is done by professional web designers. Web designers are comfortable with tools such as Dreamweaver, Microsoft FrontPage and so on. Most portals at present, provide a web based user interface to create, design, and theme a portal. So, it would be better and an advantage, if portals provide a method by which a web designer is able to use professional web design tools to create an elegantly designed portal.

ViewDesigner makes it possible for a web designer to create a well designed user interface for a portal in the Project WebSynergy. ViewDesigner aims to provide plugins for popular web

design tools, which enables a web designer to easily customize, design, and theme a portal page. For more information on ViewDesigner, see <https://designview.dev.java.net/>

At present, Project WebSynergy provides a plugin for DreamWeaver that works in conjunction with the Desktop Design Tool (DDT) feature.

Using Desktop DDT and ViewDesigner Plugin

This section explains how a web designer can use the DDT and ViewDesigner plugin in Project WebSynergy.

1. Install the extension in Dreamweaver (.mxd file) by navigating to MacroMedia Extension Manager -> Install Plugin -> Select the mxd file -> Install -> Accept License Agreement.
2. Download the websynergy theme war file from Sun Java System Portal Server 7.2 console's DDT by selecting the layout and theme that you wish your portal page to have. To do this:
 - a. Login to Portal Server 7.2 Console.
 - b. Click Portals tab.
 - c. Select a portal from the list.
 - d. Click Manage Themes URL.
 - e. Select a Theme to download.
 - f. Click Download.
3. Import the downloaded websynergy theme war file into Dreamweaver using the ViewDesigner plugin. To do this:
 - a. Open Dreamweaver.
 - b. Click File.
 - c. Select Import and click Import Skin option.

The ViewDesigner plugin opens the portal sample page (index.html) in Dreamweaver.

4. Enter values for the following:
 - Path to websynergy theme war file — Location where you have downloaded the websynergy theme war file.
 - Folder to extract websynergy theme war file — The directory where you want to extract the websynergy theme war file.
 - Path to Java.exe — Location of the java.exe.

If you enter the values for the Java.exe path for the first time, it is persisted by the plugin. This path will be pre-populated next time you navigate to the Import or Export skin dialog box.

5. Click OK. If you enter values for all the fields correctly, the plugin opens the portal sample page (index.html / index.htm).

Note – These are the error messages that are displayed when you do not enter the values for the fields correctly.

- Please select the folder to extract the websynergy theme war file. — If you leave the "Folder to extract websynergy theme war file" blank and click OK.
 - The file path entered for Skin Extract Folder is incorrect. — If you enter incorrect value for the "Folder to extract websynergy theme war file" and click OK.
 - Please select the websynergy theme war file. — If you leave the "Path to websynergy theme war file" blank and click OK.
 - The file path entered for websynergy theme war file is incorrect. — If you enter incorrect path for the "Path to websynergy theme war file" and click OK.
 - Please select only a .websynergy theme war file. — If you enter a non-websynergy theme war file path for the "Path to websynergy theme war file" and click OK.
 - Please select the Java Executable Path. — If you leave the "Path to Java.exe" blank and click OK.
 - The file path entered for Java.exe is incorrect. — If you enter incorrect path for the "Path to Java.exe" and click OK.
 - Please choose only a java.exe file — If you enter the path of non-java.exe file for the "Path to Java.exe" and click OK.
 - The Sun Java Systems Portal Server Skin Extension is not installed. — If you do not install the extension click OK.
-

6. Click the Help button to get any help related to filling up the Import Skin dialog box.
7. Click the Cancel button to close the Import Skin dialog box.
8. Make changes to the downloaded portal skin by modifying the css files included in the downloaded websynergy theme war file. The changes that can be made to the theme are:
 - Update any existing attributes of the css files. For example, font color, size and so on.
 - Add new attributes to the existing css styles.
 - Point to any other existing image or create a new image and point the css style to that.
 - When you create a new image, ensure that it is created in the images folder of the websynergy theme war file.
 - You **cannot** create new styles in an existing css file or create new css files.
 - In Portal Server 7.2 release, only the theming can be modified. The layouts **cannot** be modified.
9. Export the changes made to the websynergy theme war file using the plugin by clicking the Export Skin option in the Dreamweaver —> File —> Export.

10. Enter values for the following:

- a. Path to the Unzipped websynergy theme war file — The location where the websynergy theme war file has been extracted and theme modified.
- b. Save New websynergy theme war file to — The name and location of the new websynergy theme war file (that contains the modified theme).
- c. Path to Java.exe -> Location of the java.exe.

11. Click OK.

If export is successful, you will get the confirmation message as “Export Complete”. Use the Portal Server console to apply a new theme.

These are the error messages that are displayed when you do not enter the field values correctly.

- Please select the folder that contains the unzipped Sun Java System Portal Server websynergy theme war file. — If you leave the "Path to the Unzipped websynergy theme war file" blank and click OK.
- The file path entered for Unzipped websynergy theme war file is incorrect. — If you enter incorrect path for the "Path to the Unzipped websynergy theme war file" and click OK.
- Please select the Sun Java Systems Portal Server websynergy theme war file. — If you leave the "Save New websynergy theme war file to" blank and click OK.
- Please choose only a .websynergy theme war file. — If you enter non-websynergy theme war file path for the "Save New websynergy theme war file to" and click OK.
- Please choose only a .websynergy theme war file. — If you enter non-websynergy theme war file path for the "Save New websynergy theme war file to" and click OK.
- Please select the Java Executable Path. — If you leave the "Path to Java.exe" blank and click OK.
- The file path entered for Java.exe is incorrect. — If you enter incorrect path for the "Path to Java.exe" and click OK.
- Please choose only a java.exe file — If you enter path of non-java.exe file for the "Path to Java.exe" and click OK.
- The Sun Java System Portal Server Skin Extension is not installed. — If you do not install the extension properly and click OK.

12. Click the Cancel button to close the Export Skin dialog box.

13. Use the The DDT in Portal Server console to apply the changes to your live portal and view the look and feel change that has taken effect immediately. To do this:

- a. Login to Portal Server console.
- b. Click Portal tab.

- c. Click a Portal.
- d. Select an Organization from the Select DN list box.
- e. Click Manage Themes URL.
- f. Click the Upload button.
- g. Enter a Theme name and description. Click Save to upload the newly modified websynergy theme war file.
The new theme appears on the list of available themes.
- h. Click on theme name URL to preview the portal page. You can view all the changes that were applied.
- i. Select the DN as EnterpriseSample and choose the option next to the theme.
- j. Click the "Set as Default" button.
- k. Navigate to the portal page and click the EnterpriseSample URL. The changes that were seen can be applied here.

API Reference

To be documented

API Reference Section

To be documented

◆ ◆ ◆ 4 CHAPTER 4

Sample Code

To be documented

Sample Code Section

To be documented

