



Sun Java System Portal Server 7 Community Sample Guide

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

Part No: 819-3031
January 2006

Copyright 2006 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, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. 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 2006 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.

Certaines 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. 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.



060125@13215



Contents

Preface	7
1 Understanding the Community Sample Portal	11
Overview of the Community Sample	11
Community Sample Portal	12
Community Sample Desktop	12
Content Filepath	13
Community Sample Portal Channels	13
Community Services	13
Community Portlets	15
2 Configuring Back-end Services	17
Configuring Back-end Services for Portlets	17
▼ To Install and Configure the Back-end Services	17
▼ To Enable Mail Service to the Organization Using Delegated Administration CLI	18
▼ To Enable Proxy Authentication in Calendar Server	18
▼ To Partition the Proxy Users in a Separate Organization	18
Gathering Information for the Portlets	19
3 Installing and Configuring Shared Events Portlet	21
Deploying the SharedEvents Portlet	21
▼ To Install and Configure SharedEvents Portlet	21

4	Installing and Configuring the Shared Tasks Portlet	25
	Installing and Configuring the SharedTasks Portlet	25
	▼ To Install and Configure the SharedTasks Portlet	25
5	Deploying and Editing the Wiki Portlet	29
	Deploying the Wiki Portlet	29
	▼ To Deploy the Portlet	29
	▼ To Enable Wiki Portlet to Display Channels and/or Portlets	30
	Wiki Preferences	31
6	Open-Source Java DB	33
	Introduction to Java DB	33
	Introduction to the Database	33
	Database Configuration	34
	Datasource Creation	34
	Starting, Stopping, and Disabling the Java DB	34
	▼ To Start and Stop Java DB	34
	▼ To Disable Java DB	35
7	Community Template	37
	Overview of the Community Template	37
	What Is A Community Template?	37
	What Are the Available Templates?	38
	How Are The Templates Stored?	38
	How Are The Templates Managed?	39
	Template Syntax and Semantics	40
	Template Descriptor File	41
	Creating and Modifying a Template	42
	▼ To Create a New Template for Single Portal Environment	42
	▼ To Customize or Modify an Existing Template for Single Portal Environment	43
	▼ To Create a Template for Multi-Portal Environment	43
8	Customizing the Community Sample User Interface	45
	Overview of the Community Sample Desktop User Interface	45
	Introduction to the Community Sample	45
	Customizable Files	46

Sample Customizations	50
▼ To Use a Custom Stylesheet	50
▼ To Change the Logo and Title Appearance in the Header	50
9 Customizing Sun Java Web User Interface Theme	51
Introduction to Sun Java Web User Interface Theme	51
Customizing the User Interface	51
▼ To Use Sun Java Web User Interface Theme in a Page	52
▼ To Modify or Create a New Sun Java Web User Interface Theme	52
10 Modifying the Properties Files	55
communityportlets.properties File	55
communitymc.properties File	56
communitygmt.properties File	57
A Attributes	59
B Tokens for Wiki Portlet	61
C Tokens for Shared Tasks Portlet	65
D Tokens for Shared Events Portlet	69
E Tokens for Surveys and Polls Portlet	73
F Tokens for File Sharing Portlet	75

Preface

The *Sun Java System Portal Server 7 Community Sample Guide* provides an overview of the Sun Java™ System Portal Server software Community Sample. It also includes instructions for deploying and customizing portlets manually after the Portal Server software installation, setting up templates, and customizing the user interface of the sample portal.

Who Should Use This Book

The audience for this book include:

- Installers and administrators installing one or more of the community services manually after the Portal Server software installation
- Administrators and other individuals customizing the properties and setting up custom templates for communities
- Administrators and other individuals customizing the community sample portal user interface

Related Books

- *Sun Java System Portal Server 7 Command-Line Reference*
- *Theme for Sun Java Web User Interface Components* Tech Note

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/>)
- Support (<http://www.sun.com/support/>)
- 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> . <i>A 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	#

Default Paths and File Names

The following table describes the default paths and file names used in this book.

TABLE P-3 Default Paths and File Names

Term	Description
<i>PortalServer-base</i>	Represents the base installation directory for Sun Java System Portal Server 7 software. The Portal Server software default base installation and product directory depends on your specific platform: Solaris™ systems: /opt/SUNWportal Linux systems: /opt/sun/portal
<i>AccessManager-base</i>	Represents the base installation directory for Sun Java™ System Access Manager software. The Access Manager software default base installation and product directory depends on your specific platform: Solaris™ systems: /opt/SUNWam
<i>DirectoryServer-base</i>	Represents the base installation directory for Sun Java™ System Directory Server software. The Directory Server software default base installation is /var/opt/mps/serverroot.
<i>ApplicationServer-base</i>	Represents the base installation directory for Sun Java™ System Application Server software. The Application Server software default base installation is /opt/SUNWappserver8.
<i>WebServer-base</i>	Represents the base installation directory for Sun Java™ System Web Server software. The Web Server software default base installation is /opt/SUNWwbsvr.
<i>PortalServer-DataDir</i>	Represents the directory where JSPs, templates and property files, and tag libraries are installed. By default, this is: <ul style="list-style-type: none">■ /var/opt/SUNWportal/ on Solaris■ /var/opt/sun/portal on Linux

Understanding the Community Sample Portal

This chapter contains the following:

- “Overview of the Community Sample” on page 11
- “Community Sample Portal” on page 12
- “Community Sample Portal Channels” on page 13

Overview of the Community Sample

The portal collaboration feature allows end users to create and join communities, and interact with other community members through a set of collaborative portlet applications (community services). Communities are, therefore, an association of members and services. These services are: file share, shared tasks & events, polls & surveys, wiki, and discussions.

Communities are created and managed by end-users. The user can be in one or more roles (visitor, owner, member). The owner of the community can remove the community or transfer ownership of the community.

The portal administrator can define community templates that defines the layout mechanism and the available services, and end-users can create communities within categories using the available template(s). See [Chapter 7](#) for more information. Users must join to take part in community collaboration. All communities are public, in that any user may join a community.

An end-user can

- Search for a community based on the community name and description. Then the user can join a community that is discovered via searching.
- Browse through community names and descriptions based on categories. Then the user can join a community that is discovered via browsing.

Each portal will have its own set of communities. Communities within a portal will only be visible to users in that portal. The community users are stored in a relational database, one database instance per portal (see [Chapter 6](#) for more information).

Community Sample Portal

This section provides an overview of the Community Sample portal:

- [“Community Sample Desktop” on page 12](#)
- [“Content Filepath” on page 13](#)

Community Sample Desktop

After installing the Portal Server software, the welcome page at `http://host:port/portal/welcome/index.jsp` includes links to the three sample portals. Select the Community Sample to view the Community Sample portal anonymous desktop.

The Community Sample desktop anonymous page, by default, includes:

- A customizable header with logo, title, help button, and date/time info and a footer with links to Sun resources
- A breadcrumb with link to get back to the Welcome page

The Communities Home page can be viewed by logging in as `test/test`. To participate in the Community Sample, users must be in the `CommunitySample` organization. The Communities home page allows users to browse and/or search communities, view list of their communities, preview and visit interesting communities, and create communities using the community portlets. For more information on the community portlets, see [“Community Portlets” on page 15](#). This page includes:

For each community that the user is a member of, they will have a community page that allows the user to interact with the services. This is the page that allows the user to leave the community and remove the community if the user is the owner. The community visitor page allows users to preview a community and join the community.

For more information on customizing the user interface of the sample portal, see [Chapter 8](#).

Content Filepath

After installation and configuration of the Community Sample PAR file, content for Community Sample portal is deployed in the following directories.

PortalServer-DataDir/portals/portal-ID/communitytemplates/

This directory contains the templates for the Community Sample and the associated properties files for the templates.

PortalServer-DataDir/portals/portal-ID/desktop/community_sample/

Provides dynamic content including JavaServer™ Pages (JSP™), tag library for Sun Java™ Web User Interface Components' theme (*portletSetupTags.tld*), properties file, and containers.

PortalServer-DataDir/portals/portal-ID/desktop/classes/CommunitySamplePortal.properties

Provides properties for the Community Sample. If you modify properties in this file, restart the server for the new values to take effect.

PortalServer-DataDir/portals/portal-ID/web-src/community

Provides static content including JavaScript, images, style sheet (*css/desktop.css*), and channel content.

Community Sample Portal Channels

This section includes description of the community:

- “Community Services” on page 13
- “Community Portlets” on page 15

Community Services

A community service is an application that runs in a community context to provide per-community collaborative functions to end-users. Community services are portlets or providers. The Portal Server includes the following community services:

File sharing The File sharing (JSF-based) application offer a hierarchical view of a file repository similar to File System browsing applications provided by the OS. Through this hierarchical view, the user can browse the repository, download files, upload files, create folders, delete and rename files and folders, copy and move files and folders. All these user interactions are done through a portlet.

This service can be customized by modifying the *defaulttheme.jar* file (see [Chapter 9](#)) and default deployment values can be modified in the *tokens.properties* file (see [Appendix F](#)).

Survey	<p>The Survey (JSF-based) application allows users to create surveys for the community, answer the surveys, then review and search survey results in the portlet. There are two types of users for the Survey portlet application: end user and community administrator. All members of a community can create a survey.</p> <p>This service can be customized by modifying the <code>defaulttheme.jar</code> file (see Chapter 9) and default deployment values can be modified in the <code>tokens.properties</code> file (see Appendix E).</p>
Polls	<p>The Poll (JSF-based) application allows users to create polls for the community, answer the polls, then review and search poll results in the portlet. There are two types of users for the Poll portlet application: end user and community administrator. All members of a community can create a poll.</p> <p>This service can be customized by modifying the <code>defaulttheme.jar</code> file (see Chapter 9) and default deployment values can be modified in the <code>tokens.properties</code> file (see Appendix E).</p>
Shared Events	<p>The Shared Events application allows users to create, update, delete, search Calendar events for the community this portlet serves. The portlet allows users to create, retrieve, update, and delete Calendar events. Only users within the community have access to the shared calendar associated to the community.</p> <p>This service can be customized by modifying the <code>defaulttheme.jar</code> file (see Chapter 9) and default deployment values can be modified in the <code>tokens.properties</code> file (see Appendix D).</p>
Shared Tasks	<p>The Shared Tasks (JSF-based) application allows users to create, update, delete, search Calendar tasks for the community this portlet serves. The portlet allow users to perform create, retrieve, update, and delete operations on Calendar tasks. Only users within the community have access to the shared calendar associated to the community.</p> <p>This service can be customized by modifying the <code>defaulttheme.jar</code> file (see Chapter 9) and default deployment values can be modified in the <code>tokens.properties</code> file (see Appendix C).</p>
Search	<p>The Search service allows community users to search for content within a community. This channel is provider-based.</p>

Discussions	This application allows discussions in the context of the community. It also supports HTML discussions with links. This channel is provider-based.
Subscriptions	This application allows community user's to subscribe to discussions out of the community discussion channel, and to save searches from the community search channel. It differs from the developer sample, in that it does not allow category subscriptions. This channel is provider-based.
Wiki	<p>This application supports wiki-based communities and portlets. An entire community can be set up as a wiki (with portlets within it) and single portlets within a community can also be wikis. The wiki portlet supports both text and HTML modes. An HTML wiki uses the Xinha HTML editor and includes a portlet picker for easy selection of portlets to include within a wiki page. The Wiki can be configured to use either files or database at the back-end. The Developer Sample desktop also uses a wiki portlet in its wiki tab.</p> <p>The Wiki portlet uses JSPWiki (http://jspwiki.org) and the HTML editor uses Xinha (http://xinha.python-hosting.com).</p> <p>For more information on:</p> <ul style="list-style-type: none"> ■ Deploying this portlet, see Chapter 5. ■ Modifying this portlet tokens, see Appendix B.

Most of these services use Sun Java Web User Interface Components. These services allow for customizations via modifications to stylesheet. See [Chapter 9](#) for more information.

Community Portlets

The community portlets bundled with the Portal Server implement the community infrastructure.

Community portlet	This application allows users to create communities
Role Management portlet	This application lists all registered members of the community and allows the community owner to add and delete users from the community. Community owners can also transfer ownership of the community to a registered member via this portlet. Only community owners are given access to the Role management portlet and can delete a community. Registered users can join any community and leave a community at any time.

Configuring Back-end Services

Certain service portlets in the Community Sample require certain back-end services to be setup and configured prior to installing and configuring the portlet. These back-end services are Sun Java™ System Calendar Server and Sun Delegated Administration software. You can use an existing deployment or, choose to perform a fresh install and configuration of these services. This chapter provides instructions for configuring the back-end services for the Community Sample portlets.

This chapter contains the following:

- [“Configuring Back-end Services for Portlets” on page 17](#)
- [“Gathering Information for the Portlets” on page 19](#)

Configuring Back-end Services for Portlets

The Calendar Server, Delegated Administration software, and Sun Java System Portal Server software must use the same Sun Java System Directory Server. Mail service should be enabled for the organization.

- [“To Install and Configure the Back-end Services” on page 17](#)
- [“To Enable Mail Service to the Organization Using Delegated Administration CLI” on page 18](#)
- [“To Enable Proxy Authentication in Calendar Server” on page 18](#)
- [“To Partition the Proxy Users in a Separate Organization” on page 18](#)

▼ To Install and Configure the Back-end Services

- Step** ● See detailed documentation for:

- Installing Calendar Server at Configuring Calendar Server at Part II, “Postinstallation Configuration,” in *Sun Java System Calendar Server 6 2005Q4 Administration Guide* in *Sun Java System Calendar Server 6 2005Q4 Administration Guide*.
- Installing Delegated Administration software at Configuring Delegated Administration software at *Sun Java System Communications Services 6 2005Q4 Delegated Administrator Guide*.

▼ To Enable Mail Service to the Organization Using Delegated Administration CLI

- Step** ● See Chapter 5, “Command Line Utilities,” in *Sun Java System Communications Services 6 2005Q4 Delegated Administrator Guide*.

For example, type `/opt/SUNWcomm/bin/commadmin domain modify -D admin -n domain -w passwd -S mail -H preferred-mailhost -d domain` to add mail LDAP attribute to the user.

▼ To Enable Proxy Authentication in Calendar Server

- Steps**
1. Edit `CS-config-directory/ics.conf` file (where *CS-config-file* is by default `/etc/opt/SUNWics5/config/` directory) and change the value of `service.http.allowadminproxy` property to `yes`.
 2. Restart the Calendar Server.

▼ To Partition the Proxy Users in a Separate Organization

A proxy user is created for each community to manage that community’s shared calendar. Follow the instructions in this section to avoid populating the manually created Portal Server users’ organization.

- Steps**
1. Create a new parallel organization with a unique domain name to the organization containing the community users via Delegated Administrator. For example, create an organization called `proxyusers.com`.
 2. Set the `DA_DEFAULT_DOMAIN` to the proxy user’s newly created organization domain name. For example, set the `DA_DEFAULT_DOMAIN` to `proxyusers.com`. If the portlets have already been deployed, update the relevant SSO adapters via the Portal

Server administration console and set the `DA_DEFAULT_DOMAIN` to the proxy users organization domain name.

Gathering Information for the Portlets

Before deploying the portlets, gather the following information:

<code>UGRoot Suffix</code>	This is the server/Group Root suffix of the Directory Server. This information is provided while running the <code>comnds_setup.pl</code> script.
<code>CSUGRoot Suffix</code>	This is the <code>UGRoot Suffix</code> under which the calendar users are created by default. It is under this suffix, the calendar admin user (<code>calmaster</code>) is created. It is important to choose this suffix appropriately such that the calendar admin user is able to proxy-authenticate other calendar users in other organizations. It is recommended that this suffix be the same as <code>UGRoot Suffix</code> . For the portlets, this specifies the value of Base Distinguished Node configuration input while configuring Calendar Server.
<code>DA Default Domain</code>	This is the DNS domain name of the default organization used by Delegated Administration software (for provisioning users by default). For the portlets, this specifies the value of Default Domain configuration input while configuring Delegated Administration software.
<code>DA Default Organization DN</code>	The Organization distinguished node will be the LDAP subtree under which all the users and groups that belong to the Delegated Administration software Default Domain are located. Note that the distinguished node must be located under the <code>UGRoot Suffix</code> . For the portlets, this specifies the value of Organization Distinguished Node for the Default domain configuration input while configuring Delegated Administration software.

Installing and Configuring Shared Events Portlet

This chapter includes instructions for installing and configuring the shared Events portlet manually after the Sun Java System Portal Server software installation.

Deploying the SharedEvents Portlet

▼ To Install and Configure SharedEvents Portlet

Before You Begin Configure the back-end services and gather the required information as outlined in [Chapter 2](#).

Steps 1. **Create the portlet WAR file.**

To create the WAR file:

Note – After Portal Server software has been installed, the Shared Events application will be under the *PortalServer-base/portletapps/sharedevents* directory. If Portal Server has been configured, the shared events application will be in *PortalServer-DataDir/portals/portal-ID/portletapps/sharedevents* directory.

a. **Edit the `tokens.properties` file to specify customizable parameters for the shared events portlet.**

See [Appendix D](#) for more information.

b. **Run `ant customize`.**

2. Load SSO Adapter Template.

Change directories to

PortalServer-DataDir/portals/portal-ID/portletapps/sharedevents/src/conf
directory and type **ant -f config.xml** to load the SSO Adapter configuration.

3. Deploy the Portlets.

a. Create a file containing the password of the amadmin user. For example, type,
echo mypassword >/tmp/ampasswd.

b. Type *PortalServer-base/bin/psadmin deploy-portlet -u amadmin -f /tmp/ampassword -p portal-ID -i myInstance -g --userinfofile FULL_PATH_TO_userInfoMapping.properties FULL_PATH_TO_sharedevents.war_FILE.*

c. Restart the container.

4. Add the Display Profile data to the templates.

The Shared Events portlet must be added to the Community templates to show up in new communities (created after the application has been installed). The template files are located at

PortalServer-DataDir/portals/portal-ID/communitytemplates/ by default.

Add the DP to the *member.xml* file in the template's directory.

For Shared Events, do the following:

a. Add your new channel to the available ones.

```
<Available>
....
<Reference value="%COMMUNITY_CONTAINER%/Sharedevents"/>
</Available>
```

b. Add your new channel to the selected ones.

```
<Selected>
....
<Reference value="%COMMUNITY_CONTAINER%/Sharedevents"/>
</Selected>
```

c. Add the channel definition.

```
<Channels>
....
<Channel name="Sharedevents" provider="__Portlet__sharedevents.sharedevents">
  <Properties>
    <String name="title" value="Community Calendar Events"/>
    <String name="description" value="Community Events Portlet"/>
    <String name="width" value="thick"/>
    <String name="__Portlet__ps.communityId" value="|DUMMY"/>
    <String name="__Portlet__community.calendar.config" value="|DUMMY"/>
    <Collection name="__Portlet__PreferenceProperties">
      <Collection name="default">
```

```

        <String name="ps.communityId" value="|DUMMY"/>
        <String name="community.calendar.config" value="|DUMMY"/>
    </Collection>
    <Collection name="isReadOnly">
        <Boolean name="ps.communityId" value="false"/>
        <Boolean name="community.calendar.config" value="false"/>
    </Collection>
</Collection>
<Collection name="userInfoDescriptions">
    <Collection name="timezone">
        <String name="en-US" value="Preferred User Time Zone"/>
    </Collection>
</Collection>
</Properties>
</Channel>
</Channels>

```

5. **Restart the web container.**
6. **Login and verify that the service channel is displayed on your desktop.**
 - a. **Go to `http://HOST:PORT/portal/dt` and select **Community Sample**.**
 - b. **Login as `test/test` and create a new community. Then log out.**
 - c. **Re-login to the **Community Sample** portal as `test/test`.**
The Shared Events portlet will be displayed.

Installing and Configuring the Shared Tasks Portlet

This chapter includes instructions for installing and configuring the shared Tasks portlet on the Community Sample portal.

Installing and Configuring the SharedTasks Portlet

▼ To Install and Configure the SharedTasks Portlet

Before You Begin Configure the back-end services and gather the required information as described in [Chapter 2](#).

Steps 1. **Create the portlet WAR file.**

To create the WAR file:

- a. **Edit the `tokens.properties` file to specify customizable parameters for the portlet you are deploying.**

See [Appendix C](#) for more information.

- b. **Type `/usr/sfw/bin/ant customize`.**

After Portal Server software has been installed, the Shared Tasks application will be under the `PortalServer-base/portletapps/sharedtasks` directory. If Portal Server has been configured, the shared tasks application will be in `PortalServer-DataDir/portletapps/sharedtasks` directory.

2. Load SSO Adapter Template.

Change directories to

PortalServer-DataDir/portletapps/sharedtasks/build/conf directory and type **ant -f config.xml** to load the SSO Adapter configuration.

3. Deploy the Portlets.

a. Create a file containing the password of the amadmin user. For example, type, **echo mypassword >/tmp/ampasswd**.

b. Type **/opt/SUNWportal/bin/psadmin deploy-portlet -u amadmin -f /tmp/ampassword -p myPortal -i myInstance -g --userinfofile FULL_PATH_TO_userInfoMapping.properties FULL_PATH_TO_sharedtasks.war_FILE**.

c. Restart the container.

4. Add the Display Profile data to the templates.

The Shared Tasks portlet must be added to the Community templates to show up in new communities (created after the application has been installed). The template files are located at

PortalServer-DataDir/portals/portal-ID/communitytemplates/ by default.

Add the DP to the member.xml file in the template's directory.

For Shared Tasks, do the following:

a. Add your new channel to the available ones.

```
<Available>
....
<Reference value="%COMMUNITY_CONTAINER%/Sharedtasks"/>
</Available>
```

b. Add your new channel to the selected ones.

```
<Selected>
....
<Reference value="%COMMUNITY_CONTAINER%/Sharedtasks"/>
</Selected>
```

c. Add the channel definition.

```
<Channels>
....
<Channel name="Sharedtasks" provider="__Portlet__sharedtasks.sharedtasks">
  <Properties>
    <String name="title" value="Community Calendar Tasks"/>
    <String name="description" value="Community Tasks Portlet"/>
    <String name="width" value="thick"/>
    <String name="__Portlet__ps.communityId" value="|DUMMY"/>
    <String name="__Portlet__community.calendar.config" value="|DUMMY"/>
    <Collection name="__Portlet__PreferenceProperties">
      <Collection name="default">
```

```
        <String name="ps.communityId" value="|DUMMY"/>
        <String name="community.calendar.config" value="|DUMMY"/>
    </Collection>
    <Collection name="isReadOnly">
        <Boolean name="ps.communityId" value="false"/>
        <Boolean name="community.calendar.config" value="false"/>
    </Collection>
</Collection>
<Collection name="userInfoDescriptions">
    <Collection name="timezone">
        <String name="en-US" value="Preferred User Time Zone"/>
    </Collection>
</Collection>
</Properties>
</Channel>
</Channels>
```

5. **Restart the web container.**
6. **Login and verify that the service channel is displayed on your desktop.**
 - a. **Go to `http://HOST:PORT/portal/dt` and select **Community Sample**.**
 - b. **Log in as `test/test` and create a new community. Then log out.**
 - c. **Re-log in to the **Community Sample** portal as `test/test`.**
The Shared Tasks portlet will be displayed.

Deploying and Editing the Wiki Portlet

This chapter contains the following sections:

- “Deploying the Wiki Portlet” on page 29
- “Wiki Preferences” on page 31

Deploying the Wiki Portlet

This section includes instructions for deploying and editing the wiki portlet after the Portal Server installation. After Portal Server is installed, portlets (available for deployment) are in the *PortalServer-base/portletapps/portletname* directory. If Portal Server has been configured, the portlet application will be in *PortalServer-DataDir/portals/portal-ID/portletapps/portletapp* directory. If Portal Server has not been configured, you must manually copy the files to the *PortalServer-DataDir/portals/portal-ID/portletapps/portletapp* directory before deploying the portlet.

▼ To Deploy the Portlet

- Steps**
1. **Log in to the Portal Server host and change directories to where the portlet application is located.**

The wiki portlet is in the *PortalServer-DataDir/portals/portal-ID/portletapps/wiki* directory.

2. **Create the portlet WAR file.**

To create the WAR file:

- a. Edit the `tokens.properties` file to specify customizable parameters for the portlet.
See [Appendix B](#) for more information.
 - b. Run `ant customize`.
Ensure that Java™ DB is running before running `ant`.
3. Deploy the Portlets.
- a. Create a file containing the password of the `amadmin` user. For example, type, `echo mypassword >/tmp/ampasswd`.
 - b. Type `/opt/SUNWportal/bin/psadmin deploy-portlet -u amadmin -f /tmp/ampassword -p myPortal -v -i myInstance -g --userinfofile FULL_PATH_TO_userInfoMapping.properties FULL_PATH_TO_wiki.war_FILE`.
 - c. Restart the container.
4. Log in and verify that the wiki portlet is deployed.
- a. Go to `http://HOST:PORT/portal/dt` and select `sample` where you deployed the wiki portlet.
 - b. Log in to the sample and view the portlet.

▼ To Enable Wiki Portlet to Display Channels and/or Portlets

- Step**
- The wiki portlet must be the first selected element of a `DynamicAggregationContainer`. The display profile of the Developer Sample desktop and of the sample wiki community include such a container. The wiki portlet can display and aggregate any channel or portlet that is in the available list of the `DynamicAggregationContainer`. If a channel is not in the available list, it will not appear on the page with no error or other indication. This allows the wiki portlet to include different channels for different users based on their display profiles.

Wiki Preferences

The wiki portlet includes the following preferences that can be modified via the Portal Server administration console:

<code>wikiTemplate</code>	Specifies the name of the JSP wiki template used by this wiki portlet. The Portal Server software includes two of these: a full page template called <code>wikicomcommunity</code> and a rearranged version used by smaller wiki portlets called <code>wikiportlet</code> .
<code>wikiPage</code>	Specifies the name of the default wiki page for this wiki portlet
<code>editorType</code>	Specifies the type of editor used by this wiki portlet. Values can be <code>text</code> or <code>HTML</code> . A text wiki portlet uses a simple text editor, whereas an HTML wiki uses a WYSIWYG HTML editor.
<code>editorOptions</code>	Specifies the HTML editor plug-ins used in this wiki. This is specified only when <code>editorType</code> is <code>HTML</code> . Valid values are: <code>PortletPicker</code> , <code>Stylist</code> , <code>CharacterMap</code> , <code>ContextMenu</code> , <code>FullScreen</code> , <code>ListType</code> , <code>SuperClean</code> , <code>TableOperations</code> .
<code>availablePortlets</code>	Specifies the list of channel names (either portlets, or provider channels) which this wiki portlet allows the user to select from when placing a portlet on the page. Note that only channels in the corresponding <code>DynamicAggregationContainer</code> available list will actually be displayed. Typically, these two lists would contain the same channels.

Open-Source Java DB

This chapter contains the following:

- [“Introduction to Java DB” on page 33](#)
- [“Starting, Stopping, and Disabling the Java DB” on page 34](#)

Introduction to Java DB

The sections contains the following:

- [“Introduction to the Database” on page 33](#)
- [“Database Configuration” on page 34](#)
- [“Datasource Creation” on page 34](#)

Introduction to the Database

The Sun Java System Portal Server software uses the open-source database written in the Java™ programming language (Java™ DB) to store configuration and membership for the collaboration feature. The Portal Server software installs and configures the database. The Java DB runs as a standalone Java application. The Portal Server software does not manage the Java DB process; it must be manually started and stopped using the Java DB NetworkServerControl application (see [“Starting, Stopping, and Disabling the Java DB” on page 34](#) for more information). The database credentials are `portal/portal`.

Database Configuration

The Portal Server software creates a Java DB server instance at install time. A Java DB server instance is defined by a unique value for the Java DB system home directory. By default, the value is *PortalServer-DataDir/derby*. Whenever the Java DB server process is started, it must be passed the Java DB system home value by specifying the system property `derby.system.home`. The Java DB system home directory contains the database configuration, logs, and data files.

Datasource Creation

The Portal Server software components that use the Java DB access it via J2EE JDBC datasources. When a new portal instance is created, the portal software creates one JDBC datasource for each component that accesses the database. In other words, there is one datasource per component, per portal server instance.

Portal accesses the database using a JDBC datasource configured in the web container. The datasource configuration can be modified using the web container console, or command line interface. The database URL for the Java DB community database is of the form `jdbc:derby://host:port/component_portal-ID`. When connecting to the Java DB using third-party tools, use the driver `org.apache.derby.jdbc.ClientDriver`. This driver is in the JAR file `/usr/share/lib/Derby/derbyclient.jar`.

Starting, Stopping, and Disabling the Java DB

This section contains the following:

- “To Start and Stop Java DB” on page 34
- “To Disable Java DB” on page 35

▼ To Start and Stop Java DB

For more information, see <http://db.apache.org/derby/javadoc/publishedapi/org/apache/derby/drda/NetworkServerControl.html> and <http://db.apache.org/derby/docs/10.0/manuals/admin/hubprnt14.html#HDRCBDJHHFD>.

- Step** ● **The database can be stopped and started using the Java DB `NetworkServerControl` class.**

For example, type:

- `java -Dderby.system.home=PortalServer-DataDir/derby org.apache.derby.drda.NetworkServerControl start` to start the database.
- `java -Dderby.system.home=PortalServer-DataDir/derby org.apache.derby.drda.NetworkServerControl shutdown` to stop the database.

Note – To run this command:

- The `derby.jar`, `derbytools.jar`, and `derbynet.jar` files must be in your classpath. By default, these JAR files are installed into `/usr/share/lib/Derby` directory.
 - The system property `Dderby.system.home` must be set to `PortalServer-DataDir/derby`.
-

▼ To Disable Java DB

- Steps**
1. **Log in to the Portal Server host as root and go to `PortalServer-DataDir/portals/portal-ID/config/` directory.**
 2. **Edit the `communitymc.properties` file and remove the `jdo` entry from the `manager.contributors` list to disable Java DB.**
See “[communitymc.properties File](#)” on page 56 list for more information on this file and its contents. Note that if this change is applied, the community sample will not function properly.

Community Template

This chapter contains the following sections:

- [“Overview of the Community Template” on page 37](#)
- [“Template Syntax and Semantics” on page 40](#)
- [“Template Descriptor File” on page 41](#)
- [“Creating and Modifying a Template” on page 42](#)

Overview of the Community Template

This section contains the following:

- [“What Is A Community Template?” on page 37](#)
- [“What Are the Available Templates?” on page 38](#)
- [“How Are The Templates Stored?” on page 38](#)
- [“How Are The Templates Managed?” on page 39](#)

What Is A Community Template?

A community template is comprised of a set of services (channels) and the visual layout. However, the layout is not always dictated by the community template as in the case with wiki community template where the layout is dictated by the wiki itself. Community templates define (in the role display profile document) the type of services available for the community, the default settings for each service, and the container(s) that bind the services.

Physically, a community template is a properties file, and image, plus one or more display profile documents. There may be up to three display profile documents, one per community role (such as OWNER, VISITOR or MEMBER). Each role template

defines services and the layout associated with the particular role. The content of the role template is represented in a display profile document. In essence, a community template contains the logic for handling different roles (one display profile document per role) and depending on the role, you get a different set of services and a different layout.

Communities are created from a community template. The system may have any number of community templates. In the Community Sample, end users choose a community template when they create a community.

What Are the Available Templates?

The Portal Server software includes:

Two Column	A two column table layout with thin channels on the left and wide channels on the right.
Left Navigation	A menu navigated content layout with menu on the left and selected content on the right.
Wiki	A WYSIWYG layout to allow users to create wiki-like content and page layout including support for wiki plugins, attachments, editing, syntax, and the such

Custom templates can be added to the system. See [“Creating and Modifying a Template” on page 42](#) for more information.

How Are The Templates Stored?

The community templates are stored on filesystem. Community templates are stored in *PortalServer-DataDir/portals/portal-ID/communitytemplates* directory (referred to as *communityTemplateBaseDir*). Note that this means that each Portal (in a multi-portal deployment environment) will/must have its own set of community templates. The resource bundle in *communityTemplateBaseDir* defines the meta-data associated with each template. In addition, each template has its own directory where the role templates are stored.

EXAMPLE 7-1 Sample *communityTemplateBaseDir*

```
communityTemplateBaseDir  -+-- template1 -+-- owner.xml
                          |
                          +-- member.xml
                          +-- visitor.xml
                          |
                          -+-- template2 -+-- owner.xml
                          |
                          +-- member.xml
                          +-- visitor.xml
                          |
                          -+-- template3 -+-- owner.xml
                          |
                          +-- member.xml
                          +-- visitor.xml
                          |
                          +-- template1.properties
                          |
                          +-- template1_en.properties
                          |
                          +-- template1_fr.properties
                          |
                          +-- template2.properties
                          |
                          +-- template3.properties
                          |
                          +-- template3_en_US.properties
                          |
                          +--      ...
```

How Are The Templates Managed?

The portal administrator can add a new community template, update an existing community template, archive and restore community templates on the system, and export community templates from one portal instance to others and/or keep them in sync.

Template Syntax and Semantics

Each template is made up of one or more role templates (`member.xml`, `owner.xml`, `visitor.xml`) in XML format. The template directory includes the XML files for the roles that it will serve; for example, `member.xml` for the community member, `owner.xml` for the community owner, and `visitor.xml` for the community visitor.

Each role template is a display profile document for community users in that role. The file must be based on the display profile DTD.

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<DisplayProfile version="1.0" priority="%COMMUNITY_DP_PRIORITY%">
  <Properties/>
  <Channels>
    <Container name="%COMMUNITY_CONTAINER%" provider="JSPTTableContainerProvider">
      <Properties>
        <String name="title" value="%COMMUNITY_NAME%"/>
        <String name="description" value="%COMMUNITY_DESCRIPTION%"/>
        <Boolean name="compileToRealPath" value="true"/>
      </Properties>
      <Available>...</Available>
      <Selected>...</Selected>
      <Channels>...</Channels>
    </channels>
  <Providers/>
</DisplayProfile>
```

The tokens (surrounded by %), described below, in the display profile are dynamically replaced by actual values by the template engine when a community is created.

`%TOKEN_COMMUNITY_NAME%`

Specifies the (user-friendly) name given to the community. For example, `tourists`.

`%TOKEN_COMMUNITY_ID%`

Specifies the unique string identifying the community. This name is strictly an internal representation and does not get exposed in the user interface. For example, `jdo__tourists`.

`%TOKEN_COMMUNITY_DESCRIPTION%`

Includes a description of the community.

`%TOKEN_COMMUNITY_CONTAINER%`

Specifies the top-level container for the community. For example, `jdo__touristsContainer`.

- `%TOKEN_COMMUNITY_DP_PRIORITY%`
Specifies the display profile merging priority given to the resulting community display profile. Each role is given a different value. By default, 1000 for the visitor role, 1005 for the member role, and 1010 for the owner role.
- `%TOKEN_COMMUNITY_SEARCH_URL%`
Specifies the Search server URL for the community.
- `%TOKEN_COMMUNITY_CONTENTS_SEARCH_DB%`
Specifies the search database for the community content.
- `%TOKEN_COMMUNITY_DISCUSSIONS_SEARCH_DB%`
Specifies the discussions database.

Template Descriptor File

Each template includes a resource bundle properties file which defines the meta-data associated with that template. The resource bundle is referred to as the descriptor file that can be localized. Each template descriptor file (must) define the following properties:

- `id` Specifies an unique ID of the template. The ID must match the template directory name. For example, `Baseball` for a template directory named `Baseball` with role templates (or XML files) for all three supported roles.
- `name` Specifies an user-friendly name used in the user interface (portal desktop) to identify the template. For example, `Baseball Template`.
- `description` Contains a verbose description of the template including the services it offers. For example, `Baseball-themed template containing the following services: Player Statistics, Game Discussions, TV Schedule, and Online Chat`.
- `tokens` Includes the list of tokens used in the template role files. This merely serves an informative purpose and is not required. For example, `%COMUNITY_ID% %COMMUNITY_DESCRIPTION% %COMMUNITY_CONTAINER%`.
- `previewImageURI` Specifies either the absolute or relative URI to the portal context. For example, `http://images.domain.com/images/baseball.jpg`. The relative URI must be relative to the portal web-app context path.

EXAMPLE 7-2 Sample Descriptor File

```
id=Baseball
name=Baseball Template
description=Baseball-themed template containing the following services: Player Statistics, Game Di
tokens=%COMMUNITY_ID% %COMMUNITY_DESCRIPTION% %COMMUNITY_CONTAINER%
previewImageURI=http://images.domain.com/images/baseball.jpg
```

Creating and Modifying a Template

To create a new or modify an existing template, following the instructions in this section. You can create a template in one of the following three ways:

- Export the template, add content, and import the content using the `psadmin` utility.
- Create content and import the content to overwrite existing template.
- Add new files to existing templates.

▼ To Create a New Template for Single Portal Environment

Steps 1. Go to the `communityTemplateBaseDir` and create a:

- New directory for the new template
- Copy an existing template to the new template directory

For example, type:

```
cd PortalServer-DataDir/portals/portal-ID/communitytemplates
mkdir NewTemplate
cp 2column/* NewTemplate/
```

2. **Modify the role based display profile documents in the new template directory as needed.**

For more information on the role based display profile documents, see [“Template Syntax and Semantics”](#) on page 40.

3. **Create and edit the properties file to include the properties described in Template Descriptor File and save the file.**

For example, to create a new properties files for the new template, type:

```
cp 2colimn.properties NewTemplate.properties
```

Or,

touch *NewTemplate.properties*

Note – In order to see the newly added template, log out of any current portal session and re-login to see the change.

▼ To Customize or Modify an Existing Template for Single Portal Environment

- Steps**
1. Go to the *communityTemplateBaseDir/template* directory and open the file you wish to modify.
 2. Log out of any current portal session and re-login to see the change.

▼ To Create a Template for Multi-Portal Environment

In a multi-portal environment (when there are more than one portal on the system), use PAR mechanism (as opposed to directly editing files in *communityTemplateBaseDir*) so that the change of community templates can be applied across multiple portals. This will allow all the portals to have the same set of community templates. If you do not wish to have synchronized environment across portals, use the instructions outlined in [“To Create a New Template for Single Portal Environment”](#) on page 42.

- Steps**
1. Either use `psadmin export --type desktop` to export desktop data (which includes community templates) and then export it so the content can be edited or, create a new PAR structure from scratch with only the community templates and no other desktop data.
Follow instructions in [“To Create a New Template for Single Portal Environment”](#) on page 42 to edit content.

■ **Create a new PAR file which contains:**

```
-- META-INF -- MANIFEST.MF
|
+-- pbfiles -+-- communityTemplateBaseDir -+-- template1 -+-- owner.xml
|                                                    |
|                                                    +-- member.xml
|                                                    |
|                                                    +-- visitor.xml
|
|                                                    +-- template1.properties
|                                                    |
|                                                    +-- template1_en.properties
|                                                    |
|                                                    +-- template1_fr.properties
|                                                    |
|                                                    +-- ...
|
+-- static -- community -- images -- template1.gif
```

2. **Edit or add content as needed.**
3. **Create a new PAR file.**
4. **Use `psadmin import` subcommand to import the PAR content across all portals.**

If you exported all desktop data, note that `psadmin export` subcommand will export all desktop data; if you create a new PAR structure from scratch with only the community templates, the command will only export community templates.

Tip – For more information, see the “`psadmin export`” in *Sun Java System Portal Server 7 Command-Line Reference*.

Customizing the Community Sample User Interface

This chapter contains the following:

- [“Overview of the Community Sample Desktop User Interface” on page 45](#)
- [“Sample Customizations” on page 50](#)

Overview of the Community Sample Desktop User Interface

This section contains the following:

- [“Introduction to the Community Sample” on page 45](#)
- [“Customizable Files” on page 46](#)

Introduction to the Community Sample

The Community Sample includes a display profile loaded at install time that contains the definitions of containers, channels, and providers used by the sample. The sample relies on the display profile’s aggregation methods (see SJS Tech Ref Guide for more information) and the community templates for displaying communities and their contained portlets.

The display profile for the sample is a set of containers which handle basic aggregation and display operations. After installation, directories and files effecting the user interface of the sample portal can be found in the *PortalServer-DataDir/portals/portal-ID/desktop/community_sample* directory.

The Community Sample relies on a combination of two stylesheets (see [“Stylesheets for the Community Sample” on page 49](#)) which define the fonts, colors, and images used for its user interface. There are intentionally no conflicts in the selection of these two stylesheets.

Customizable Files

The Community Sample uses the following to render the sample portal user interface:

- [“JSP and HTML Files” on page 46](#)
- [“Container Provider, Channel Provider, and Channel Templates” on page 47](#)
- [“Properties Files” on page 49](#)
- [“Stylesheets for the Community Sample” on page 49](#)

JSP and HTML Files

`header.jsp`

This file in

PortalServer-DataDir/portals/portal-ID/desktop/community_sam directory contains the header display for all sample containers. It controls user interface for the sample via inclusion of the stylesheets.

`footer.html`

This file in

PortalServer-DataDir/portals/portal-ID/desktop/community_sam directory is included by containers (CommunityParentContainer) and closes HTML tags (namely, `body` and `html` tags) as well as displays some footer information.

`datetime.jsp`

This file in

PortalServer-DataDir/portals/portal-ID/desktop/community_sam directory is included by the `header.jsp` file and provides data/time info based on the user's time zone setting. For anonymous user, the default server time zone is used. There is currently no user interface style in this file.

`breadcrumb.jsp`

This file in

PortalServer-DataDir/portals/portal-ID/desktop/community_sam directory is included by the `header.jsp` and provides a rudimentary breadcrumb used in the sample. The user interface in this file is controlled via `desktop.css` file.

`singlePreferenceMenubar.jsp`

This file in

PortalServer-DataDir/portals/portal-ID/desktop/community_sam directory provides menubar used by `JSPDynamicSingleContainer`.

<code>singlePreferenceHeader.jsp</code>	This file in <i>PortalServer-DataDir/portals/portal-ID/desktop/community</i> directory provides header used by <code>JSPDynamicSingleContainer</code> .
<code>portletEdit.jsp</code>	This file in <i>PortalServer-DataDir/portals/portal-ID/desktop/community</i> directory wraps portlet and Provider content in an edit page.

JavaScript and Tag Libraries

<code>openURLInParent.js</code>	This file in <i>PortalServer-DataDir/portals/portal-ID/desktop/community_s</i> directory provides JavaScript for popup and detachment of channels in their own browser window.
<code>tld/portletSetupTags.tld</code>	This file in <i>PortalServer-DataDir/portals/portal-ID/desktop/community_s</i> provides single tag to include Sun Java™ Web User Interface Components theme support for stylesheet and JavaScript functionality.

Container Provider, Channel Provider, and Channel Templates

<code>PagePreferencesContainer</code>	This directory (under <i>PortalServer-DataDir/portals/portal-ID/desktop/classes/</i> directory) contains the files that allows a user to change the content and layout of portlets contained in an available container or community template container for which the <code>isEditable</code> property in the display profile is <code>true</code> . If <code>true</code> , the <code>header.jsp</code> file includes a link to Page Preferences in the breadcrumb area at right.
<code>AccountPreferencesContainer</code>	This directory (under <i>PortalServer-DataDir/portals/portal-ID/desktop/classes/</i> directory) contains the files used for allowing a user to change account preferences such as name, password, timezone, and so on.

CommunityParentContainer	This directory (under <i>PortalServer-DataDir/portals/portal-ID/desktop/classes/</i> directory) is an instance of JSPSingleContainerProvider and includes the default channel defined in the portal server console for the sample.
CommunityAnonymousContainer	This container includes a set of channels representing the anonymous user's sample display and it is an instance of the JSPTableContainerProvider.
CommunityHomeContainer	This container includes the set of channels representing the logged in user's sample display and it is an instance of the JSPTableContainerProvider.
JSPEditContainer	This directory (under <i>PortalServer-DataDir/portals/portal-ID/desktop/classes/</i> directory) contains a version of the JSPEditContainer's default files with different styles matching the overall style of the Community Sample.
Login	This channel is included in the CommunityAnonymousContainer's display profile definition and provides the user with the ability to login to the sample or create a new account.
error	This directory includes error templates. Style for these file's user interface is defined in the two stylesheets mentioned above.
JSPTableContainerProvider	This directory (under <i>PortalServer-DataDir/portals/portal-ID/desktop/classes/</i> directory) contains a version of the JSPTableContainer's default files with different styles matching the overall style of the Community Sample.
JSPMenuContainerProvider	This directory (under <i>PortalServer-DataDir/portals/portal-ID/desktop/classes/</i> directory) provides for left-side navigation layout.
JSPDynamicSingleContainer	This directory is under <i>PortalServer-DataDir/portals/portal-ID/desktop/community_sam</i> directory.

DiscussionProvider	Provides for Community look and feel changes for this provider.
SearchProvider	Provides for Community look and feel changes for this provider.

Properties Files

message.properties

This file in

PortalServer-DataDir/portals/portal-ID/desktop/community_sample/ directory provides properties used by the template files in the *error* directory. The properties in this file are swapped by the error templates in the *PortalServer-DataDir/portals/portal-ID/desktop/community_sample/error* directory at runtime.

CommunitySamplePortal.properties

This file in *PortalServer-DataDir/portals/portal-ID/desktop/classes/* directory provides properties for the Community Sample portal. These properties are used by the Community Sample portal JSPs. This file includes properties for the Community Sample portal desktop:

- Header, Masthead, Breadcrumb, and footer
- Page Preferences link and Account Settings link
- Login channel
- channelMenubar and popupChannelMenubar
- communityParentContainer/single.jsp file
- JSPEditContainer/edit.jsp file
- JSPMenuContainer/enumain.jsp file

If you modify this file, you must restart the server for the changes to take effect.

Stylesheets for the Community Sample

`defaulttheme.jar` This JAR file includes the stylesheet that controls many defaults properties for the Community Sample including: font, colors, and the like not found in the `desktop.css` file.

For more information on the Sun Java Web User Interface Components stylesheet, see *Theme for Sun Java Web User Interface Components* Tech Note.

`desktop.css` Augmentation of style is included for other components via an additional stylesheet located with the rest of the sample's web-src in the web container's web source directory under *PortalServer-DataDir/portals/portal-ID/web-src/community/css* directory. This stylesheet is not browser specific and modifications to this file only effects the sample portal's

overall appearance and that of the containers used.

Sample Customizations

This section includes the following:

- [“To Use a Custom Stylesheet” on page 50](#)
- [“To Change the Logo and Title Appearance in the Header” on page 50](#)

▼ To Use a Custom Stylesheet

- Steps**
1. **Log in to the Portal Server and change directories to**
PortalServer-DataDir/portals/portal-ID/web-src/community/css.
 2. **Rename the `desktop.css` file to `desktop.css.orig`.**
For example, type `mv desktop.css desktop.css.orig`.
 3. **Copy your custom stylesheet (CSS file) to this directory or modify the `desktop.css` file.**

▼ To Change the Logo and Title Appearance in the Header

- Steps**
1. **Log in to the Portal Server host and edit the `desktop.css` file in the**
PortalServer-DataDir/portals/portal-ID/web-src/community/css directory.
 2. **Modify the:**
`#header #logo` To modify the logo.
`#header #title` To modify the title in the header.

Customizing Sun Java Web User Interface Theme

This chapter contains the following:

- [“Introduction to Sun Java Web User Interface Theme”](#) on page 51
- [“Customizing the User Interface”](#) on page 51

Introduction to Sun Java Web User Interface Theme

The `defaulttheme.jar` file in `PortalServer-base/lib` directory is an instance of the Sun Java Web User Interface Components theme JAR file and during installation, the configurator appends `PortalServer-base/lib/defaulttheme.jar` onto the web container’s shared server classpath. The `defaulttheme.jar` file controls many default properties for the Community Sample including fonts and colors.

Customizing the User Interface

This section contains the following:

- [“To Use Sun Java Web User Interface Theme in a Page”](#) on page 52
- [“To Modify or Create a New Sun Java Web User Interface Theme”](#) on page 52

▼ To Use Sun Java Web User Interface Theme in a Page

- Step** ● In the JSP files that generate the <head> section of the HTML (or the `header.jsp` file), you can designate a Sun Java Web User Interface Components theme to be used for the entire page. For example:

```
<pui:setupTheme themeName="defaulttheme" .../>
```

To use Sun Java Web User Interface Components theme in a page, the `portletSetupTags.tld` tag library descriptor file must be included; otherwise, the `<pui:setupTheme ...>` will fail. This file must be located where the JSP files can access it. For example, in the community sample, it is located at `PortalServer-DataDir/portals/portal-ID/desktop/community_sample/tld` directory.

▼ To Modify or Create a New Sun Java Web User Interface Theme

- Steps**
1. **Make a copy of `defaulttheme.jar` file.** For example, copy this file to `PortalServer-DataDir/portals/portal-ID/`.
 2. **Unpack the `defaulttheme.jar` file and modify the following files (as needed).**
 - CSS files
 - Image Files
 - Javascript
 - Properties Files

For more information on these files, see the *Theme for Sun Java Web User Interface Components* Tech Note. When modifying the `defaulttheme.jar` file, leave the manifest file intact.

3. **Make a JAR of the files and change the server classpath to point to the new location.**

When packaging the theme into a JAR file, specify an existing manifest file. For example, type `jar cmf META-INF/MANIFEST.MF com` to retain the original manifest file. This command will prevent the `jar` command from auto-generating a new manifest file.

For example, change classpath from `PortalServer-base/lib/defaulttheme.jar` to `PortalServer-DataDir/portals/portal-ID/defaulttheme.jar`. In a multi-portal environment, you can:

- Have all portals point to a single `theme.jar` file.

- Or, have each portal associated with its own `theme.jar` file. For example:
 - The web container serving *portal1* can have its server classpath pointing to `PortalServer-DataDir/portals/portal1/theme/portal1theme.jar` file.
 - The web container serving *portal2* can have its server classpath pointing to `PortalServer-DataDir/portals/portal2/theme/portal2theme.jar` file.
4. **Restart the web container for the changes to take effect and reload the Community Sample in your browser to see the changes.**

Tip – It is not necessary to have the theme in a JAR file format. Alternatively, you can set up the server classpath to include the directory containing the exploded theme hierarchy. This is especially useful during development cycles since it allows you to edit individual files and see the result instantly reflected.

Changes in properties file require a web container restart; changes to Javascript, images, and CSS files will be reflected without restarting the server.

Modifying the Properties Files

This chapter contains the following:

- “communityportlets.properties File” on page 55
- “communitymc.properties File” on page 56
- “communitymgmnt.properties File” on page 57

communityportlets.properties File

The configuration variables defined in the `communityportlets.properties` file, in most cases, need not be modified. To modify the file, log into any system that has portal server installed as superuser and access the file in the `PortalServer-DataDir/portals/portal-ID/config/` directory.

The file contains the following (editable) variables:

`ps-community-create-type`

Specifies the membership and configuration contributor type to use when creating new communities.

`search-server-url`

Specifies the URL for all the community portlets, which are integrated with search, to access search server for submitting, indexing, as well as searching documents.

`community-search-database-prefix`

Specifies a prefix when a database is created for a community. For example:

```
community_contents_portal1
community_discussions_portal1
```

`community_communities_portall`

`search-taxonomy-root`

Specifies the root of the taxonomy. The value can be `ROOT` to use the whole taxonomy tree for categorization. Or, specify any valid full path of a classification to use part of the taxonomy tree.

`dp-priority-base`

Specifies the base value used to set display profile priorities for created communities.

communitymc.properties File

This file configures the community membership and configuration component of the Portal Server software. For the most part, this file contains low level configuration that need not be modified. However, this file must be changed to disable Java DB use in portal.

The file contains the following variables:

<code>manager.package</code>	Specifies the Java package name for manager implementation classes.										
<code>manager.contributors</code>	Specifies the active membership and configuration contributor types in the system. By default, this list includes: <table><tr><td><code>am-global</code></td><td>Access Manager service schema</td></tr><tr><td><code>am-org</code></td><td>Access Manager organizations</td></tr><tr><td><code>am-role</code></td><td>Access Manager roles</td></tr><tr><td><code>am-frole</code></td><td>Access Manager filtered roles</td></tr><tr><td><code>jdo</code></td><td>Java DB, via Java Data Objects (JDO) layer</td></tr></table>	<code>am-global</code>	Access Manager service schema	<code>am-org</code>	Access Manager organizations	<code>am-role</code>	Access Manager roles	<code>am-frole</code>	Access Manager filtered roles	<code>jdo</code>	Java DB, via Java Data Objects (JDO) layer
<code>am-global</code>	Access Manager service schema										
<code>am-org</code>	Access Manager organizations										
<code>am-role</code>	Access Manager roles										
<code>am-frole</code>	Access Manager filtered roles										
<code>jdo</code>	Java DB, via Java Data Objects (JDO) layer										

Each of these contributor types provides membership and configuration data from a different source.

communitymgmnt.properties File

This file contains configuration for the community management SDK component. The community management SDK is not public, and this file should not be modified.

Attributes

The community portlets must construct URLs that access and control the community sample in which they are executing. The following attributes define how these community URLs are constructed:

`CommunityParentContainerURLParameter`
(`CommunityParentContainer.selectedChannel`)
Specifies the top level container for the community sample, as well as the URL parameter that causes this container to select a new contained channel.

`CommunityHomeContainerName` (`CommunityHomeContainer`)
Specifies the name of the container that defines the community home page.

`CommunityCreateContainerName`
Specifies the name of the community portlet channel used to create new communities.

In most cases, you will not need to change these values. You may want to change them if, for example, you are designing a new sample from scratch, and are using different container and channel names than the defaults used in the community sample. To help understand how the system uses these attributes, some tokenized example community URLs are provided below:

`http://host:port/portal/dt?`
`CommunityParentContainerURLParameter=CommunityHomeContainerName`
Access community home page.

`http://host:port/portal/dt?`
`CommunityParentContainerURLParameter=x_Container`
Access a community page, for community ID x.

`http://host:port/portal/dt?`
`CommunityParentContainerURLParameter=CommunityCreateContainerName`
Access the community create portlet.

Tokens for Wiki Portlet

The wiki is a portlet on top of JSPWiki. For more information on JSPWiki, see <http://www.jspwiki.org>. The Wiki supports the following configuration tokens in the *PortalServer-DataDir/portals/portal-ID/portletapps/wiki/tokens.properties* file:

`WIKI_REPOSITORY_TYPE`

Specifies JDBC for database (default) file for filesystem.

`WIKI_WORK_DIR`

Specifies the location of temp files and search index.

`WIKI_FILESYSTEM_REPOSITORY_DIR`

Specifies the location of filesystem repository (not used for JDBC repository).

`WIKI_INITIAL_PAGESET`

Specifies the location of default page which is copied to create the initial file system repository (not used for JDBC repository)

`DB_ADMIN_URL`, `DB_ADMIN_USER`, `DB_ADMIN_PASSWORD`,

`DB_ADMIN_DRIVER_CLASS`, and `DB_ADMIN_DRIVER_CLASSPATH`

These are used by the system to create the wiki database repository when the wiki is first configured.

The Wiki template allows for the following runtime tokens in the

PortalServer-DataDir/portals/portal-ID/portletapps/wiki/tokens.properties file:

`jspwiki.applicationName`

Specifies the default application name. It affects the HTML titles and logging, for example. It can be different from the actual web name (<http://my.com/mywiki>) of the application, but usually it is the same.

`jspwiki.pageProvider`

Specifies the provider class to use. For example, the values can be:

`RCSFileProvider`

For simple RCS-based file storage.

`FileSystemProvider` For simple pure file storage with no version information.

`VersioningFileProvider` For simple, non-RCS based versioning storage.

By default, this is set to `JDBCPageProvider`.

`jspwiki.JDBCPageProvider.url`, `jspwiki.JDBCPageProvider.username`,
`jspwiki.JDBCPageProvider.password`,
`jspwiki.JDBCPageProvider.driver`, and
`jspwiki.JDBCPageProvider.cachedConnections`
Specifies the database connection when using `JDBCPageProvider`.

`jspwiki.JDBCPageProvider.continuationEditMinutes`
Specifies the minutes for editing the page before increasing the version of the page. Multiple edits by the same user within this interval will not increment the page version number and will appear as a single edit in the page history.

`jspwiki.JDBCPageProvider.migrateFrom` and
`jspwiki.JDBCAttachmentProvider.migrateFrom`
Specifies the path to the properties files of the imported page and attachment repositories, which are used to build the initial database repository from default data, or to migrate from old versions of JSPWiki.

`jspwiki.fileSystemProvider.pageDir`
Specifies where wiki files are kept for `FileSystemProvider` and `RCSFileProvider`.

`jspwiki.workDir`
Specifies the JSPWiki working directory. The working directory is used to cache things like Lucene search results.

`jspwiki.attachmentProvider`
Specifies the attachment provider to use. You can specify `BasicAttachmentProvider` for a simple, flat file versioning provider or leave the value empty to disable the functionality. By default, this is set to `JDBCAttachmentProvider`.

`jspwiki.basicAttachmentProvider.storageDir`
If you specifies the `BasicAttachmentProvider` as the attachment provider, this specifies where to store the files the user has uploaded.

`jspwiki.baseURL`
Specifies the BaseURL which can be used to rewrite all of JSPWiki's internal references.

`jspwiki.translatorReader.allowHTML`
Specifies whether raw HTML is allowed as Wiki input. By default, this is set to true. You can also specify yes or no, or on or off.

`jspwiki.templateDir`
Specifies the default template used by the Wiki engine. The templates live in *wiki-webapp-location/templates/template-name* directory. JSPWiki will attempt to

find two basic templates (`ViewTemplate` and `EditTemplate`) from that directory. The default JSPWiki template is called `default` which is provided as a reference only. The wiki portlet uses templates called `wikicomunity` and `wikiportlet`.

`jspwiki.lockExpiryTime`

Specifies the number of minutes a person can lock a page while editing it.

`log4j.logLevel`

Specifies the logging level.

`log4j.appender.FileLog.File`

Specifies the log file location. The default log location is dynamically set to *PortalServer-DataDir/logs/instance-name/jspwiki.log*.

Tokens for Shared Tasks Portlet

Shared Tasks defines the following tokens in the *PortalServer-DataDir/portals/portal-ID/portletapps/sharedtasks/tokens.properties* file:

SSOADAPTER_TEMPLATE_NAME	Default value is CTY-SHARED-TASKS. Specifies the name of SSO Adapter name. This SSO Adapter contains the required configuration information for the portlet functioning. This SSO Adapter is per-portlet application; that is, the name should be different for each deployed portlet instance. For example, HACKERS_COMMUNITY_TASKS
SSOADAPTER_TEMPLATE_DESC	Default value is CTY-SHARED-TASKS. Contains the description for SSO Meta Adapter. This needs to be the same as SSOADAPTER_TEMPLATE_NAME for the template to be editable from the management console. For example, HACKERS_COMMUNITY_TASKS.
SSOADAPTER_ORGANIZATION_DN	Specifies the distinguished node of organization under which the SSO Adapter is loaded. The user retrieves the SSOAdapter from the Organization the user belongs to.

DA_SERVER_HOST	<p>For example, o=CommunitySample,dc=siroe,dc=example,dc=com.</p> <p>Specifies the fully qualified name of the system where Delegated Administration server is running.</p> <p>For example, daserver.blue.planet.com.</p>
DA_SERVER_PORT	<p>Default value is 80.</p> <p>Specifies the port number of the Delegated Administration server.</p> <p>For example, 80.</p>
DA_DOMAIN_SEPARATOR	<p>Default value is @.</p> <p>Specifies the user ID and Domain Separater character.</p> <p>For example, @.</p>
DA_PROXY_ADMIN_USER_ID	<p>Specifies the admin user for Delegated Administration that can be used to perform proxy authentication of users.</p> <p>For example, admin.</p>
DA_PROXY_ADMIN_USER_PASSWORD	<p>Specifies the password for DA_ADMIN_USER_ID.</p> <p>For example, abc123.</p>
DA_DEFAULT_DOMAIN	<p>Specifies the DNS Domain Name (sunPreferredDomain) of the organization where the CTY-PROXY-USER would be provisioned. The CTY-PROXY-USERS can be created in a different organization than the community users, in which case it should be manageable by Delegated Administrator.</p> <p>For example, blue.planet.com.</p>
IS_HOSTED_DOMAIN_ENV	<p>Default value is false.</p> <p>Specifies whether the portlet should use a hosted domains environment.</p>

Note – If the Delegated Administration server and Calendar Server are configured to use hosted domains setup, the value needs to be set to true.

CALENDAR_SERVER_HOST

For example, false

Specifies the fully qualified name of the system where Calendar Server is running.

For example,
calendar.blue.planet.com.

CALENDAR_SERVER_PORT

Default value is 3080.

Specifies the port Number of Calendar Server.

For example, 3080.

CALENDAR_PROXY_ADMIN_UID

Specifies the admin user for Calendar Server that can be used to perform proxy authentication of users.

For example, calmaster.

CALENDAR_PROXY_ADMIN_PASSWORD

Specifies the password for
CALENDAR_PROXY_ADMIN_UID.

For example, calmaster.

AUTO_DELETE_CTY_PROXY_USER_CAL

Default value is false.

Specifies whether the portlet should delete the CTY-PROXY-USER when a community is deleted. The CTY-PROXY-USER is the community proxy user, a user created by the Shared Tasks portlet

For example, false.

AM_ADMINISTRATOR_DN

Specifies the distinguished name of the administrator for Sun Java System Access Manager software.

For example,
uid=amAdmin,ou=People,dc=blue,dc=planet,dc=c

AM_ADMINISTRATOR_PASSWORD

Specifies the password for
AM_ADMINISTRATOR_DN.

For example, abc123.

AMADMIN_COMMAND

Default value is
/opt/SUNWam/bin/amadmin.

Specifies the fully qualified path name to
the amadmin CLI of Access Manager.

For example,
/opt/sun/am/bin/amadmin.

Tokens for Shared Events Portlet

Shared Events defines the following tokens in the *PortalServer-DataDir/portals/portal-ID/portletapps/sharedevents/tokens.properties* file:

`SSOADAPTER_TEMPLATE_NAME`

Default value is `CTY-SHARED-EVENTS`.

Specifies the name of SSO Adapter name. This SSO Adapter contains the required configuration information for the portlet functioning. This SSO Adapter is per-portlet application; that is, the name should be different for each deployed portlet instance.

For example, `HACKERS_COMMUNITY_EVENTS`

`SSOADAPTER_TEMPLATE_DESC`

Default value is `CTY-SHARED-EVENTS`.

Contains the description for SSO Meta Adapter. This needs to be the same as `SSOADAPTER_TEMPLATE_NAME` for the template to be editable from the Access Manager console.

For example, `HACKERS_COMMUNITY_EVENTS`.

`SSOADAPTER_ORGANIZATION_DN`

Specifies the distinguished node of organization under which the SSO Adapter is loaded. The user retrieves the SSOAdapter from the Organization that the user belongs to.

For example, `o=CommunitySample,dc=siroe,dc=example,dc=com`.

`DA_SERVER_HOST`

Specifies the fully qualified name of the system where Delegated Administration server is running.

For example, `daserver.blue.planet.com`.

DA_SERVER_PORT
Default value is 80.

Specifies the port number of the Delegated Administration server.

For example, 80.

DA_DOMAIN_SEPARATOR
Default value is @.

Specifies the user ID and Domain Separator character.

For example, @.

DA_PROXY_ADMIN_USER_ID
Specifies the admin user for Delegated Administration that can be used to perform proxy authentication of users.

For example, admin.

DA_PROXY_ADMIN_USER_PASSWORD
Specifies the password for DA_ADMIN_USER_ID.

For example, abc123.

DA_DEFAULT_DOMAIN
Specifies the DNS Domain Name (`sunPreferredDomain`) of the organization where the CTY-PROXY-USER would be provisioned. The CTY-PROXY-USERS can be created in a different organization than the community users, in which case it should be manageable by Delegated Administrator.

For example, blue.planet.com.

IS_HOSTED_DOMAIN_ENV
Default value is false.

Specifies whether the portlet should use a hosted domains environment.

Note – If the Delegated Administration server and Calendar Server are configured to use hosted domains setup, the value needs to be set to `true`.

For example, false

CALENDAR_SERVER_HOST
Specifies the fully qualified name of the system where Calendar Server is running.

For example, calendar.blue.planet.com.

CALENDAR_SERVER_PORT
Default value is 3080.

Specifies the port Number of Calendar Server.

For example, 3080.

CALENDAR_PROXY_ADMIN_UID

Specifies the admin user for Calendar Server that can be used to perform proxy authentication of users.

For example, calmaster.

CALENDAR_PROXY_ADMIN_PASSWORD

Specifies the password for CALENDAR_PROXY_ADMIN_UID.

For example, calmaster.

AUTO_DELETE_CTY_PROXY_USER_CAL

Default value is false.

Specifies whether the portlet should delete the CTY-PROXY-USER when a community is deleted.

For example, false.

CREATE_EVENTS_IN_CTY_MEMBERS_CAL

Default value is true.

Specifies whether the portlet should create the calendar event in all the member's community calendars whenever a community event is created.

For example, true.

AM_ADMINISTRATOR_DN

Specifies the distinguished name of the administrator for Sun Java System Access Manager software.

For example, uid=amAdmin, ou=People, dc=blue, dc=planet, dc=com

AM_ADMINISTRATOR_PASSWORD

Specifies the password for AM_ADMINISTRATOR_DN.

For example, abc123.

AMADMIN_COMMAND

Default value is /opt/SUNWam/bin/amadmin.

Specifies the fully qualified path name to the amadmin CLI of Access Manager.

For example, /opt/sun/am/bin/amadmin.

Tokens for Surveys and Polls Portlet

The Surveys and Polls portlets include the following tokens in the *PortalServer-base/portletapps/surveys/tokens.properties* file:

<code>%DERBY_LIB_DIR%</code>	Specifies the library where Java DB class files <code>derbyclient.jar</code> are located.
<code>%DERBY_HOST%</code>	Specifies the host server that host the Java DB.
<code>%DERBY_PORT%</code>	Specifies the port that the Java DB server host listens to for database requests.
<code>%PORTAL_ID%</code>	Specifies the Portal's ID.

The Surveys and Polls portlets `tokens.properties` file in the *PortalServer-DataDir/portals/portal-ID/portletapps/surveys* directory includes the following tokens layered in 3 sections:

Tokens for `src/conf/common.dbadmin`

Tokens in this section include:

`DB_ADMIN_DRIVER_CLASS` and `DB_ADMIN_DRIVER_CLASSPATH`
Java DB class file and classpath.

`DB_ADMIN_URL`
Java DB host name and domain, and the port on which derby server instance is listening.

`DB_ADMIN_USER` and `DB_ADMIN_PASSWORD`
Java DB administrator name and password.

Tokens for `src/conf/surveydb.datasource` and `web/WEB-INF/sun-web.xml`

Tokens in this section include:

`DB_JNDI_NAME`
Specifies the JNDI resource name used by the survey web application to lookup connection to the Java DB supporting the database back end.

DB_URL=jdbc:derby://%DERBY_HOST%:%DERBY_PORT%/surveydb_%PORTAL_ID%
Java DB host name and domain, the port on which the Java DB server instance is listening, and portal ID (for example, myPortal).

DB_DRIVER_CLASS, DB_USER, DB_PASSWORD, DB_MAX_ACTIVE, DB_MAX_IDLE,
DB_MAX_WAIT, DB_DATASOURCE, and DB_DATASOURCE_IBMWAS5
Tokens for src/conf/server.classpath.

Tokens for src/conf/server.classpath
Token for this section include DB_DRIVER_JAR.

APPLICATION_TYPE
Specifies the type of application. By default, this is set to SURVEY.

Tokens for File Sharing Portlet

The File sharing portlet includes the following tokens in the *PortalServer-DataDir/portals/portal-ID/portletapps/Filesharing/tokens.properties* file:

REPOSITORY_TYPE	Specifies the type of file repository. By default, only file system is supported, and so the default value is <code>FileSystem</code> .
FILE_SYSTEM_REPOSITORY_ROOT_DIR	Specifies the absolute directory for the file repository. By default, this is set to <code>/export/filesharing/repository</code> .
UPLOAD_FILES_TEMP_DIR	Specifies the absolute directory for file upload staging. By default, this is set to <code>/export/filesharing/temp</code> .

