Sun Java™ System

Portal Server 6
Desktop Customization Guide

2005Q1

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

Part No: 817-7694

Copyright © 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 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 of the U.S. patents listed at http://www.sun.com/patents and one or more additional patents or pending patent applications in the U.S. and in other countries.

THIS PRODUCT CONTAINS CONFIDENTIAL INFORMATION AND TRADE SECRETS OF SUN MICROSYSTEMS, INC. USE, DISCLOSURE OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR EXPRESS WRITTEN PERMISSION OF SUN MICROSYSTEMS, INC.

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 in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, the Duke logo, the Java Coffee Cup logo, the Solaris logo, the SunTone Certified logo and the Sun ONE logo 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 architecture developed by Sun Microsystems, Inc.

Legato and the Legato logo are registered trademarks, and Legato NetWorker, are trademarks or registered trademarks of Legato Systems, Inc. The Netscape Communications Corp logo is a trademark or registered trademark of Netscape Communications Corporation.

The OPEN LOOK and Sun(TM) 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 service manual are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical 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 © 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuels 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 des brevets américains listés à l'adresse http://www.sun.com/patents et un ou des brevets supplémentaires ou des applications de brevet en attente aux Etats - Unis et dans les autres pays.

CE PRODUIT CONTIENT DES INFORMATIONS CONFIDENTIELLES ET DES SECRETS COMMERCIAUX DE SUN MICROSYSTEMS, INC. SON UTILISATION, SA DIVULGATION ET SA REPRODUCTION SONT INTERDITES SANS L AUTORISATION EXPRESSE, ECRITE ET PREALABLE DE SUN MICROSYSTEMS, INC.

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

Des parties de ce produit peuvent être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, le logo Duke, le logo Java Coffee Cup, le logo Solaris, le logo SunTone Certified et le logo Sun[tm] ONE 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.

Legato, le logo Legato, et Legato NetWorker sont des marques de fabrique ou des marques déposées de Legato Systems, Inc. Le logo Netscape Communications Corp est une marque de fabrique ou une marque déposée de Netscape Communications Corporation.

L'interface d'utilisation graphique OPEN LOOK et Sun(TM) 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 license 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 ce manuel d'entretien et les informations qu'il contient sont regis par la legislation americaine en matiere de controle des exportations et peuvent etre soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucleaires, des missiles, des armes biologiques et chimiques ou du nucleaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou reexportations vers des pays sous embargo des Etats-Unis, ou vers des entites figurant sur les listes d'exclusion d'exportation americaines, y compris, mais de maniere non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une facon directe ou indirecte, aux exportations des produits ou des services qui sont regi par la legislation americaine en matiere de controle des exportations et la liste de ressortissants specifiquement designes, 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

List of Procedures 1

Preface 3

Who Should Use This Book 3
Before You Read This Book 4
How This Book Is Organized 4
Conventions Used in This Book 6
Typographic Conventions 6
Symbols 6
Default Paths and File Names 7
Shell Prompts 8
Related Documentation 8
Books in This Documentation Set 8
Other Server Documentation 9
Accessing Sun Resources Online 9
Contacting Sun Technical Support 10
Related Third-Party Web Site References 10
Sun Welcomes Your Comments 10

Chapter 1 Introduction to Customizing the Desktop 11
Types of Desktop Customizations 11
End User Customizations 12
Developer Customizations 12
Administrator Customizations 13
What Are the Areas for Customizing the Desktop? 14
Using the Display Profile 15
Using JavaServer Pages and Template Files 16
Using the Desktop Tag Library 17

Chapter 2 Modifying the Default Sample Portal 19
Introduction 19
Editing the Default Sample Portal Files 20
Changing the Desktop Type 20
Restoring the Default Sample Portal Settings 21
Deriving the Sample Desktop 22
Determining the User's Default Desktop 22

Chapter 3 Creating a New Desktop 25
Developing the Container 25
Editing the Display Profile 26
Developing and Deploying JSP or Template Files 27
Loading the Display Profile at the Appropriate LDAP Nodes 29
(Optional) Creating a Resource Bundle 30
Accessing the Desktop 30
Debugging the Desktop 30

Chapter 4 Customizing Container Tabs 31
Adding a Tab to JSPTabContainer 31
Creating a Tab Within a Tab 36
Stretching a Tab Across an Entire Container 36
Changing the Tab Image for JSP-based Tab Containers 37
Changing the Color of Tabs 37
Making a Tab the Start Tab 38
Adding a Role-Based Tab 38
Adding a Channel to a User-defined Tab 39

Chapter 5 Customizing Channels 41
Customizing Channel Refresh Times and Container Caching 41
Customizing Window Preference 42
Removing a Button 43
Changing the Channel Layout for a Table Container 46
Removing the Title Bar from a Channel 46
Changing the Channel Border Width 47
Customizing the Channel Border 48

Chapter 6 Customizing Instant Messaging 49
Disabling the User from Editing Instant Messaging Server Information 49
Automatically Closing the Instant Messaging Invite Window 52
Customizing Display of Instant Messaging Contacts 52

Chapter 7 Customizing the Anonymous Desktop 55
Configuring Anonymous Authentication 55
Accessing the Anonymous Desktop 58
Disabling the Initial Access Manager Software Login Page and Always Use Anonymous Log In 60
Modifying the Anonymous Banner and Menu Bar 61
Adding the Login Channel to the Anonymous Desktop of a Newly Created Organization 62
Modifying the Default Desktop (Container) for authlessanonymous User 66

Chapter 8 Customizing Authentication 67
Using UNIX Authentication with LoginProvider 67
Configuring LDAP Authentication for UserInfoProvider 68

Chapter 9 Modifying the Desktop Layout 71
Deriving More Desktop Layouts 71
Changing Content Layout to Support Categorizing the Available and Selected Lists 73
Customizing Existing JSPs 73
Writing a New Content Channel 73
Adding New Layouts 74
Changing the Desktop Column Layout 74

Chapter 10 Branding the Desktop 77
Changing the HTML Title (Title That Appears in the Browser) 77
Changing the Logo (Image) in the Banner Header 77
Changing the Header and Footer of the Theme, Content, and Layout Pages 78

Chapter 11 Changing Desktop Colors 81
Changing Desktop Colors 81
Changing the Default Color Scheme for an Organization 83

Chapter 12 Customizing the Global Themes 85
Customization Overview 85
Adding a Theme to the Sample Portal 86
Customizing the Current Themes 86

Chapter 13 Customizing the Service Providers 91
Overview of Customizing the Service Providers 91
Overview of Customizing the Search Provider 91
Overview of Customizing the Discussion Provider 94
Tips for Customizing the Service Providers 96
Debugging the Service Providers 96
Location of JavaServer Pages 96

Modifying JavaServer Pages 96
Accessing Channels Directly 97
Customizing the Search Provider 97
Customizing the Discussion Channels 103
Customizing DiscussionLite Channel 103
Customizing Discussions Channel 105

Chapter 14 Customizing the Desktop End-User Online Help 107
Overview of the Desktop End-User Online Help 107
Location of the Desktop End-User Online Help HTML Files 109
Modifying the Desktop End-User Online Help HTML files 110
Editing An Existing Help File 111
Creating a New Help File 111

Chapter 15 Miscellaneous JSP and Template Information 113
Performing JSP Redirects 113
JSP vs. Theme Color 114
Recompiling JSPs 114
JavaServer Page Caching Information 115
Debugging JSPs 115
Dynamic Template Reloading 116

List of Procedures

To change the Desktop type 20
To Create Customized Organization JSP and Template Files 29
To Access the Desktop 30
To Add a Tab to JSPTabContainer 31
To Stretch a Tab Across an Entire Container 36
To Change the Tab Image for JSP-based Tab Containers 37
To Change the Color of Tabs 37
To make the tab the start tab 38
To Add a Role-based Tab 38
To Customize the Channel Window Preference 42
To Remove a Button From All Channels in a Container 43
To Remove a Button From a Single Channel 44
To Change the Channel Layout for a Table Container 46
To Remove the Title Bar from a Channel 46
To Change the Border Width for all Channels in a Container 47
To Customize Channel Borders to Have Bevelled Edges, Shadows, Curved Corners, and So On . . 48
To Enable Anonymous Log In 56
To Disable Anonymous Log In 56
To Enable Authentication-less (authlessanonymous) Log In 56
To Disable Authentication-less (authlessanonymous) Log In 57
To Access the Anonymous Desktop through the Access Manager Host Name (obj.conf File) 58
To Access the Anonymous Desktop through the Portal Server Host Name (index.html File) 59
To always use Anonymous Log In 60
To Change the Banner for the Anonymous Desktop 61
To Add the Login Channel to the Anonymous Desktop of a Newly Created Organization 62
To Change the Default Channel Name for Authlessanonymous User 66
To Use UNIX Authentication with LoginProvider 67

To Enable End User Password Maintenance for LDAP Authentication 68
To Change the Desktop Column Layout from the Command Line 74
To Change the Desktop Column Layout from the Administration Console 75
To Modify Column Widths Directly (Using Scriptlets) 75
To Change the Logo (image) in the Banner 78
To Change the Header and Footer of the Theme, Content, and Layout Pages 79
To Change the Desktop Colors 81
To Change the Text 87
To Change the Sample Anonymous Desktop Theme 88
To Modify the Default Search Server 97
To Add last-modified to the Search Result Display 98
To Remove content-length from Search Results 99
To Display the Total Number of Documents in the Search Result Status Message 100
To Remove author from the Advanced Search Interface 101
To Add a New Field to Advanced Search 102
To Customize the DiscussionLite Channel Link Display Window 103
To Display DiscussionLite on the Front tab 104
To Display Additional Fields in the List View of Discussions 105
To Modify the Sort Order in List All Discussions Page 105
To Modify viewHits in View Discussion Page 106
To Inherit Classification and readACL 106
To Control Access to Discussions 106
To Create a New Online Help File and to Define the `helpURL` Value Manually 111
To Change the Template Reload Interval 116

Preface

The *Desktop Customization Guide* includes sample customizations and detailed instructions for customizing the software desktop. This preface contains the following sections:

- Who Should Use This Book
- Before You Read This Book
- How This Book Is Organized
- Conventions Used in This Book
- Related Documentation
- Accessing Sun Resources Online
- Contacting Sun Technical Support
- Related Third-Party Web Site References
- Sun Welcomes Your Comments

Who Should Use This Book

The *Desktop Customization Guide* is intended for use by administrators and other individuals responsible for customizing the software Desktop.

Audience for this guide should already understand the following technologies:

- Lightweight Directory Access Protocol (LDAP)
- Java technology
- JavaServer Pages™ (JSP™) technology

- Hypertext Transfer Protocol (HTTP)
- Hypertext Markup Language (HTML)
- Extensible Markup Language (XML)

You must also be familiar with the following software configuration, administration, and customization tasks:

- Sun Java™ System Access Manager
- Sun Java™ System Directory Server
- Sun Java™ System Web Server
- Sun Java™ System Application Server

Before You Read This Book

Portal Server is a component of Sun Java Enterprise System, a software infrastructure that supports enterprise applications distributed across a network or Internet environment. You should be familiar with the documentation provided with Sun Java Enterprise System, which can be accessed online at http://docs.sun.com/coll/entsys_05q1.

Because the Portal Server software components work together with Sun Java™ System Access Manager for user, service, and policy management and authentication, single sign-on, and logging services, you should be familiar with the documentation provided with that product. Access Manager software documentation can be accessed online at http://docs.sun.com/doc.

Because Sun Java™ System Directory Server is used as the data store for primary configuration information and user profile data in a Portal Server deployment, you should be familiar with the documentation provided with that product. Directory Server software documentation can be accessed online at http://docs.sun.com/coll/DirectoryServer_05q1.

How This Book Is Organized

The first chapter of this book provides an overview of the areas of customization. Chapters 2 through 14 provide sample customizations and information for customizing the desktop. Chapter 15 includes essential information about JSPs and desktop templates that you will need when customizing the software.

Table 0-1 How This Book Is Organized

Title	Description
Chapter 1, “Introduction to Customizing the Desktop”	Provides an overview of the possible areas of customization.
Chapter 2, “Modifying the Default Sample Portal”	Provides information on modifying the sample portal to suit your deployment environment.
Chapter 3, “Creating a New Desktop”	Provides instructions for setting up a custom desktop.
Chapter 4, “Customizing Container Tabs”	Provides information on customizing tab containers.
Chapter 5, “Customizing Channels”	Provides information on customizing channels.
Chapter 6, “Customizing Instant Messaging”	Provides information on customizing the Sun Java™ System Instant Messaging server.
Chapter 7, “Customizing the Anonymous Desktop”	Provides information on configuring, accessing, and modifying the anonymous desktop.
Chapter 8, “Customizing Authentication”	Provides information on customizing UNIX and LDAP authentication.
Chapter 9, “Modifying the Desktop Layout”	Provides information on modifying the desktop layout.
Chapter 10, “Branding the Desktop”	Provides information on branding the default desktop.
Chapter 11, “Changing Desktop Colors”	Provides information on changing the desktop colors.
Chapter 12, “Customizing the Global Themes”	Provides information on cutomizing the GlobalThemes.
Chapter 13, “Customizing the Service Providers”	Provides information on customizing the search and discussion providers.
Chapter 14, “Customizing the Desktop End-User Online Help”	Provides information on customizing the desktop end-user online help files.
Chapter 15, “Miscellaneous JSP and Template Information”	Provides miscellaneous information on JSPs and desktop templates.

Conventions Used in This Book

The tables in this section describe the conventions used in this book.

Typographic Conventions

The following table describes the typographic changes used in this book.

Table 2 Typographic Conventions

Typeface	Meaning	Examples
AaBbCc123 (Monospace)	API and language elements, HTML tags, web site URLs, command names, file names, directory path names, onscreen computer output, sample code.	Edit your.login file. Use ls -a to list all files. % You have mail.
AaBbCc123 (Monospace bold)	What you type, when contrasted with onscreen computer output.	% **su** Password:
AaBbCc123 (Italic)	Book titles, new terms, words to be emphasized. A placeholder in a command or path name to be replaced with a real name or value.	Read Chapter 6 in the *User's Guide*. These are called *class* options. Do *not* save the file. The file is located in the *install-dir*/bin directory.

Symbols

The following table describes the symbol conventions used in this book.

Table 3 Symbol Conventions

Symbol	Description	Example	Meaning
[ ]	Contains optional command options.	ls [-l]	The -l option is not required.

Table 3 Symbol Conventions *(Continued)*

Symbol	Description	Example	Meaning
{ | }	Contains a set of choices for a required command option.	`-d {y|n}`	The `-d` option requires that you use either the `y` argument or the `n` argument.
-	Joins simultaneous multiple keystrokes.	Control-A	Press the Control key while you press the A key.
+	Joins consecutive multiple keystrokes.	Ctrl+A+N	Press the Control key, release it, and then press the subsequent keys.
>	Indicates menu item selection in a graphical user interface.	File > New > Templates	From the File menu, choose New. From the New submenu, choose Templates.

Default Paths and File Names

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

Table 0-4 Default Paths and File Names

Term	Description
PortalServer-base	Represents the base installation directory for Sun Java System Portal Server software. The Portal Server software default base installation and product directory depends on your specific platform: Solaris™ systems: /opt/SUNWps
AccessManager-base	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
DirectoryServer-base	Represents the base installation directory for Sun Java System Directory Server software. The Directory Server software default base installation is `/var/opt/mps/serverroot`.
ApplicationServer-base	Represents the base installation directory for Sun Java™ System Application Server software. The Application Server software default base installation is `/opt/SUNWappserver8`.
WebServer-base	Represents the base installation directory for Sun Java™ System Web Server software. The Web Server software default base installation is `/opt/SUNWwbsvr`.

Table 0-4 Default Paths and File Names *(Continued)*

Term	Description
PortalServer-ContentFiles	Represents the directory where JSPs, templates and property files, and tag libraries are installed. By default, this is `/etc/opt/SUNWps`.
PortalServer-SEdb	Represents the Portal Server software Search Engine database. By default, this is `/var/opt/SUNWps`.

Shell Prompts

The following table describes the shell prompts used in this book.

Table 5 Shell Prompts

Shell	Prompt
C shell on UNIX or Linux	*machine-name*%
C shell superuser on UNIX or Linux	*machine-name*#
Bourne shell and Korn shell on UNIX or Linux	$
Bourne shell and Korn shell superuser on UNIX or Linux	#
Windows command line	C:\

Related Documentation

The http://docs.sun.com℠ web site enables you to access Sun technical documentation online. You can browse the archive or search for a specific book title or subject.

Books in This Documentation Set

The following table summarizes the books included in the Portal Server core application documentation set.

Table 6 Books in This Documentation Set

Book Title	Description
Portal Server Administration Guide http://docs.sun.com/db/doc/817-7691	Describes how to administer the Portal Server software using the administration console and the command line.
Portal Server Secure Remote Access Administration Guide http://docs.sun.com/db/doc/817-7693	Describes how to administer Portal Server Secure Remote Access software.
Portal Server Developer's Guide http://docs.sun.com/db/doc/817-7695	Describes how to extend the Portal Server software APIs.
Portal Server Technical Reference Guide http://docs.sun.com/db/doc/817-7696	Provides detailed information on the the Portal Server software concepts.
Portal Server Deployment Planning Guide http://docs.sun.com/db/doc/817-7697	Describes how to plan for and deploy Portal Server software.
Portal Server Migration Guide http://docs.sun.com/db/doc/817-7698	Describes how to migrate from previous releases of the Portal Server software to this release.

Other Server Documentation

For other server documentation, go to the following:

- Directory Server documentation: http://docs.sun.com/coll/DirectoryServer_04q2
- Web Server documentation: http: //docs.sun.com/coll/S1_websvr61_en
- Application Server documentation: http://docs.sun.com/coll/s1_asseu3_en
- Access Manager documentation: http://docs.sun.com/???

Accessing Sun Resources Online

For product downloads, professional services, patches and support, and additional developer information, go to the following:

- Download Center: http://wwws.sun.com/software/download/
- Professional Services: http://www.sun.com/service/sunps/sunone/index.html
- Sun Enterprise Services, Solaris Patches, and Support: http://sunsolve.sun.com/
- Developer Information: http://developers.sun.com/prodtech/index.html

Contacting Sun Technical Support

If you have technical questions about this product that are not answered in the product documentation, go to http://www.sun.com/service/contacting.

Related Third-Party Web Site References

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.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions.

To share your comments, go to http://docs.sun.com and click Send Comments. In the online form, provide the document title and part number. The part number is a seven-digit or nine-digit number that can be found on the title page of the book or at the top of the document. For example, the title of this book is *Sun Java System Portal Server 6 2005Q1 Desktop Customization Guide* and the part number is 817-7694-10.

Chapter 1

Introduction to Customizing the Desktop

This chapter provides an introduction to customizing the Sun Java System Portal Server software Desktop. It describes the different kinds of customizations and who should make those customizations. This chapter also provides an overview of the display profile, sample Desktops included with the product, and how you create and deploy a new Desktop and provider.

This chapter contains the following sections:

- Types of Desktop Customizations
- What Are the Areas for Customizing the Desktop?
- Using the Display Profile
- Using JavaServer Pages and Template Files
- Using the Desktop Tag Library

Types of Desktop Customizations

The Desktop can be customized by end users, administrators, and developers. Though this guide covers only administrator customizations, it includes an overview of end user and developer customizations, and where to go for more information on those customizations. This guide refers to administrator customizations such as changes and modifications made to the Desktop that involve modifications to the display profile, JavaServer Pages™ and template files, search provider, and online help.

End User Customizations

End users can customize the Desktop in the following ways:

- Setting the channel time out
- Selecting column layout from the Layout page
- Moving channels up and down, as well as side to side
- Arranging channels by width
- Resizing the channel window
- Adding and removing certain channels from the Content page
- Customizing channels by using the channel Edit page
- Selecting theme from a set of preset themes or customize the theme by changing color scheme and font type for the Desktop channels
- Creating, removing, and editing tabs

Users customize channels by using the Edit Channel icon for a particular channel (as long as the administrator has made it available). Users customize the look and feel of the Desktop through the Themes page. See the Portal Server software End User Desktop Online Help for more information.

Users can also configure:

- Time zone, language, mail server, and password by using the User Info channel Edit page.
- Personal and mail information by using the Sun Java System Access Manager software administration console. The URL to use is:

  `http://`*`hostname`*`:`*`port`*`/amconsole`

Developer Customizations

The Portal Server software developers can customize the Desktop by creating:

- Provider classes
- Desktop templates
- Tag libraries
- JavaServer Pages

Developers can also use the Provider Application Programming Interface (PAPI) and the Search service APIs to extend the Portal Server software. See the *Portal Server Administration Guide* for more information.

In addition, developers can create new portal services to integrate applications and enable Single Sign-on across multiple applications, as well as implement authentication modules, using the Access Manager software APIs. See the *Access Manager Developer's Guide* for more information.

Administrator Customizations

The Portal Server software administrators can customize the Desktop by:

- Using supplied providers to define additional content channels.
- Creating and customizing the display profile, which involves creating or modifying provider, channel, and container channel objects. When you modify the display profile, you use the appropriate XML tag definitions for providers, channels, and container channels.
- Create new preset themes in the display profile.
- Using supplied JavaServer Pages and template files to modify the user interface.
- Customizing the search provider.
- Customizing the Desktop end user online help.

By performing these customizations, you can arrive at:

- A site-specific look and feel of the Desktop: whether it uses tabs or frames, what channels are available to users and how they are situated out on the Desktop, what applications are available to end users, what kind of online help is available, and so on.
- Different Desktops for different LDAP roles or organizations.
- Desktop behavior based on user roles.

This guide describes these administrator customizations.

What Are the Areas for Customizing the Desktop?

In general, the Portal Server software documentation divides Desktop interface customization into three areas:

Authentication screens You can modify the look and feel of the HTML authentication templates, including images, HTML structure, and color.

NOTE Authentication screens are provided by the Access Manager software. This guide does not describe how to customize the authentication screens. See the *Access Manager Developer's Guide* for that information.

Desktop You can modify the look and feel of the Desktop, customize the top container used by the organization, control Desktop themes, and so on. This chapter describes the Desktop customization tasks.

Files for dynamic content for providers, channels, and containers are located under the `/etc/opt/SUNWps/desktop` directory; that is, the Base Desktop templates are in `/etc/opt/SUNWps/desktop/default` and sample portal templates are in `/etc/opt/SUNWps/desktop/sampleportal` directories.

Some other static resources are located in the *web-container*`/portal` directory. This includes images used on the Desktop, such as channel control buttons, and style sheets.

NOTE References to the locations and contents of files in the `web-apps` directory are for information only. They do not represent the definition of an interface that you can depend on for any future release.

Search channel You can modify the default Search server and add or remove fields from the Advanced search interface. See Chapter 13, "Customizing the Service Providers" for more information.

Using the Display Profile

Much of your work in customizing the Portal Server software involves creating or editing the display profile to provide the kind of Desktop you want for your site. The display profile is an XML document that defines the Desktop structure and content. The display profile Document Type Definition file (DTD) defines valid syntax for the display profile XML documents. See `/etc/opt/SUNWps/dtd/psdp.dtd` for more information.

The hierarchical structuring of the display profile document does not define the visual layering of channel on the portal Desktop. The display profile exists only to provide property values for channels on the Desktop.

The display profile contains definitions that enable you to construct the Desktop. These definitions include providers, channels, containers, and properties. Some of these definitions create the Desktop containers—the frames, tables, and tabs that arrange the content of the Desktop—and others create channels for the Desktop via the respective providers. A display profile provider definition is a template for building channels based on that provider.

NOTE A provider is a Java class responsible for converting the content in a file, or the output of an application or service into the proper format for a channel. A number of providers are shipped with the Portal Server software. As the desktop is imaged, each provider is queried in turn for the content of its associated channel. Some providers are capable of generating multiple channels based upon their configuration.

Using the display profile mechanism, you can create a new portal Desktop or modify the sample portal provided with the product. See Chapter 3, “Creating a New Desktop” and Chapter 2, “Modifying the Default Sample Portal” for more information.

The display profile documents themselves consist of display profile objects. The display profile DTD defines the XML tags that represent the allowable display profile objects.

Like display profile objects are grouped within their appropriate XML tag pairs. That is, providers are grouped within `<Providers></Providers>` tags, channels within `<Channels></Channels>` tags, properties within `<Properties></Properties>` tags.

For a complete discussion of the display profile, how the merging works, and a description of the display profile DTD syntax, see the *Portal Server Technical Reference Guide*.

Using JavaServer Pages and Template Files

To generate the rendered Desktop user interface (what the industry refers to as the "presentation"), The Portal Server software makes use of either JavaServer Pages (JSP™) or template files. JSPs are preferred because they enable a much easier customization process without having to change the provider Java classes. JSPs also provide a way to enable a strict separation of business and presentation logic. Specifically, this means having the business logic in the provider classes and presentation logic in JSPs.

NOTE In general, a three-tier architecture consists of presentation logic, business logic, and the data. Tag libraries or Enterprise JavaBeans™ provide the business logic, a database contains the data, and JavaServer Pages (JSPs) or templates provide the presentation logic. However, this view is based on a "small" system where the entire system is contained in one server, or perhaps only the data is on another server.

The Portal Server software takes the "larger" system view, where all of the product is presentation. The business logic resides in some back end resource server that a content provider accesses. The data is on yet another back end server. Because all of the Portal Server software is presentation, there really is no business logic in the product.

The default set of JSPs and template files are installed in `/etc/opt/SUNWps/desktop/default` directory. The sample portal JSPs and template files are installed in `/etc/opt/SUNWps/desktop/sampleportal` and `/etc/opt/SUNWps/desktop/anonymous` directories. The Desktop Type attribute in the Desktop attributes page of the Access Manager software administration console specifies from what subdirectory to retrieve either the JSP or template files for the Desktop. For more information, see *Portal Server Technical Reference Guide*.

NOTE How the JSPs and Template files are referenced by the providers is provider-specific. Some providers specify the file in the display profile, other providers specify fixed names.

For example, for JSPProvider, there are display profile properties such as contentPage, editPage, and so on, that reference Desktop files under the `/etc/opt/SUNWps/desktop` directory. For other providers, such as BookmarkProvider, the name of the template file is fixed, for example, `display.template` and that name is mentioned in the display profile document for that provider.

Using the Desktop Tag Library

Desktops based on JSPs enable a customization process without the necessity of changing the provider Java classes. The implementation of the JSP-based Desktop uses a tag library which the Portal Server software supplies. Not all Desktop channels need to be JSP-based. There are also channels using HTML-based templates.

A tag library is exposed through Tag Library Descriptors (TLD) files, so tags are in their appropriate functional area. For more information, see the *Portal Server Technical Reference Guide*.

Chapter 2

Modifying the Default Sample Portal

This chapter contains the following sections:

- Introduction
- Editing the Default Sample Portal Files
- Changing the Desktop Type
- Restoring the Default Sample Portal Settings
- Deriving the Sample Desktop
- Determining the User's Default Desktop

NOTE This chapter describes the sample portal that you can choose to install on your system during the Sun Java System Portal Server software installation. See the *Portal Server Technical Reference Guide* for more information on the sample portal. This chapter assumes that the sample portal is installed.

Introduction

The Portal Server software includes a sample portal that demonstrates the Portal Server software features. The sample portal requires customization before deploying the portal because the sample portal does not have the custom content that you may require in your deployment environment.

Editing the Default Sample Portal Files

Do not directly edit any of the files that make up the sample portal (display profile XML, JSP, and template files). Instead, make a copy of the sample portal to a new directory and then modify those copied files. In this way you preserve the integrity of the sample portal. Additionally, if you later apply a patch to the Portal Server, you won't loose any changes you might have made to the sample portal files, as the patch would only overwrite the initially installed sample files.

Changing the Desktop Type

You should also create a custom Desktop type for your users. The Desktop type attribute of the Desktop service is a comma-separated string. It is still a string type, but the Desktop uses it as an ordered Desktop type list. The list is used by the Desktop lookup operation when searching for templates and JSPs. The lookup starts at the first element in the list and each element represents a sub directory under the Desktop template base directory. If a template is not found in the first directory, then it proceeds to the next one in the list. This continues until the item is found (or not), for all Desktop type elements in the list.

If the default directory is not included in the list, it will be added at the end of the list implicitly. For example, if the Desktop type is `sampleportal`, the target template will be searched in the `sampleportal` sub directory, then the `default` sub directory.

By default, if the sample portal is installed, then the Desktop type attribute, `sunPortalDesktopType`, is set to `sampleportal`, meaning files are retrieved from the `sampleportal` subdirectory. If the sample portal is not installed, then the Desktop type attribute value is set to `default`. The authless user is created as part of the sample portal, and the Desktop type for the authless user is set to `anonymous,sampleportal`.

You can define a new set of templates by creating a new directory under the `/etc/opt/SUNWps/desktop/` directory, placing your template files in this directory, and making this directory the Desktop Type attribute for that organization.

➤ **To change the Desktop type**

1. Create a new subdirectory in the `/etc/opt/SUNWps/desktop` directory, or whatever directory `templateBaseDir` specifies in the `desktopconfig.properties` file.

   For example:

```
mkdir /etc/opt/SUNWps/desktop/sesta
```

2. Manually copy only the template files that you wish to modify to the new directory location.

   For example, if your Desktop type will modify `content.jsp` file for JSPProvider, copy this file to `/etc/opt/SUNWps/desktop/sesta/JSPProvider/content.jsp`, and customize the file for the new Desktop type in that location.

   You only need to copy the files that you have changed from the sample installation to the new directory tree. This structure enables you to tell at a glance which files have been modified from the original distribution. It also eliminates the need to back up copies of the original sample files.

3. Use the Access Manager software administration console to change the value of the Desktop Type attribute for the subdirectory created in Step 1.

   As this attribute is dynamic, you need to change it everywhere that it appears (organization, sub-organization, role, and user). Changing the Desktop Type at the organization level will not necessarily be reflected at the user level. This will be the case only if the user has not overwritten the Desktop Type in which case the Desktop Type value will be inherited from the organization level. If the user defines the Desktop Type at the user level, the value will remain the same even if the Desktop Type is changed at the organization level.

   In this example, in the administration console, you would specify `sesta,sampleportal` as the value for the `sunPortalDesktopType` attribute.

For more information on the Desktop attributes, see *Portal Server Technical Reference Guide*. See the *Portal Server Administration Guide* for more information on editing the Desktop attributes.

Restoring the Default Sample Portal Settings

To re-load the default (original) display profile for the sample portal providers, the `dp-providers.xml` and `dp-org.xml` files (in *PortalServer-base*`/samples/desktop` directory) must be reloaded. The `dp-providers.xml` file goes in the global level and the `dp-org.xml` file goes in the organization level. For example, type:

***PortalServer-base*/bin/dpadmin modify -u "uid=amAdmin,ou=People,dc=sesta,dc=com" -w *password* -g *PortalServer-base*/samples/desktop/dp-providers.xml**

```
PortalServer-base/bin/dpadmin modify -u
"uid=amAdmin,ou=People,dc=sesta,dc=com" -w password -d "dc=sesta,dc=com"
PortalServer-base/samples/desktop/dp-org.xml
```

Deriving the Sample Desktop

The container architecture uses the various containers, and JavaServer Pages™ (JSP™) or template files, to display the Desktop, and to build a hierarchy of containers that organize the Desktop content. Channels are defined at the global display profile level and are referenced by all the containers.

Determining the User's Default Desktop

The Portal Server software determines the user's default container by examining the Default Channel Name attribute for the Desktop page in the Access Manager software administration console.

When you install the sample portal, by default, the default channel is set to JSPTabContainer. Users can view this portal by typing:

`http://`*hostname*`:`*port*`/portal/dt`

You can easily view each sample portal Desktop by typing in the appropriate URL.

NOTE The Desktop service uses a dynamic attribute, `SunPortalDesktopDefaultChannelName`, to specify the default channel to execute when accessing the Desktop. This attribute is displayed as Default Channel Name in the administration console. This attribute is only used when the `provider=name` URL parameter is not specified. The default channel service attribute can be overridden by passing to the Desktop servlet the provider parameter and setting it to the name of a channel. Once the `provider=parameter` is passed to the Desktop, this value becomes the new default channel. The service attribute is no longer used for the duration of the current user's session.

Setting the service attribute for the default channel is useful in simple scenarios when you need to set a default channel per organization or per role. In situations where you need to set the default channel based on some programmatic logic, you should use a routing container.

A routing container is a channel that reads a display profile property specifying the user's preferred default Desktop channel to determine the Desktop to present. See the *Portal Server Developer's Guide* for more information on developing a custom routing container.

Chapter 3

Creating a New Desktop

Creating a new Desktop involves the following:

- Developing the Container
- Editing the Display Profile
- Developing and Deploying JSP or Template Files
- Loading the Display Profile at the Appropriate LDAP Nodes
- (Optional) Creating a Resource Bundle
- Accessing the Desktop
- Debugging the Desktop

Developing the Container

You can develop a container by:

1. Defining a `<Container>` element in the display profile that references an existing `<Provider>` element.
2. Defining a `<Provider>` element in the display profile that references an existing provider class. You also must do Step 1.
3. Defining a container provider class that extends an existing container provider such as JSPTableContainerProvider. You also must do Step 2.
4. Defining a container provider class from scratch that extends ContainerProviderAdapter. You also must do Step 2. If you create a container by extending the Provider class, then it also needs to implement the ContainerProvider interface.

NOTE You cannot create a container provider by just extending the Provider class. By definition, a container must implement the ContainerProvider interface. ContainerProviderAdapter does this. See the *Portal Server Developer's Guide* for more information on creating custom container providers.

If you write a new class file, it must reside in the `/etc/opt/SUNWps/desktop/classes` directory. You can change this location by editing the `/etc/opt/SUNWps/desktop/desktopconfig.properties` file.

NOTE You can also use the Access Manager software administration console to manipulate containers.

Editing the Display Profile

You need to edit the display profile XML and modify the following tag:

```
<Provider name="provider" class="provider class">
```

You also need to modify the following tag, which references the provider in the previous sentence:

```
<Container name="container" provider="provider">
```

For JSP files, the `<Properties>` tag for the provider contains the following property tag, which references the JSP Content page:

```
<String name="contentPage" value="value">
```

The `<Properties>` tag for the channel can have values that override the properties set in the `<Provider>` tag. Thus, if desired, you could set the JSP `contentPage` value here. You do not reference template-based providers, or other providers you might develop, in this way.

The `<Available>` and `<Selected>` tags are required for all containers in the display profile.

The JSP-based tab, table, and frametab containers have additional properties requirements.

NOTE There is a distinction between a provider element in the display profile and the Java class for the provider.

Provider element:

```
<Provider name="JSPTableContainer"
class=com.sun.portal.providers.containers.jsp.table.JSPTa
bleContainerProvider>
```

Java class:

```
com.sun.portal.providers.containers.jsp.table.JSPTableCon
tainerProvider
```

You can modify display profile objects by performing one of the following:

- Using the Access Manager software administration console Channel and Container Management link.
- Using the Access Manager software administration console Edit XML link.
- Manually editing an existing display profile document and then loading it at the appropriate LDAP node by using the `dpadmin` modify command.
- Running the `dpadmin` command with the specified XML text changes, on standard input. When adding a new object, use the `add` subcommand. When modifying an existing object, use the `modify` subcommand.
- Creating a new display profile document from scratch and then loading it at the appropriate LDAP node by using the `dpadmin modify` command.

For more information on display profile and the `dpadmin` command, see the *Portal Server Technical Reference Guide*.

Developing and Deploying JSP or Template Files

You can modify existing JSP files (for example, `tabs.jsp`) or template files, or develop with your own. If you install the sample portal, the JSP and template files are located in the `/etc/opt/SUNWps/desktop/sampleportal` directory, in different subdirectories for each container.

See the *Portal Server Technical Reference Guide* for more information on the sample JSP and template files.

For the JSPs, you can find compilation and runtime errors in the desktop debug log at `/var/opt/SUNWam/debug/desktop.debug`. Also, all JSPProvider based Desktop channels have a property called `showExceptions`. By default, this property is set to `false`; setting it to `true` causes the JSP exception to show up as the content of the channel.

When you create a container, you need to create a new subdirectory for your newly created container in the `/etc/opt/SUNWps/desktop/`*desktopType* directory. That is, the newly created container should be placed based on what the *desktopType* is. If sampleportal is installed, then the *desktopType* is, by default, `sampleportal`; so, create a new directory for the container under `sampleportal` so that any JSP and template that is being added can adopt the same look and feel that as defined in the sampleportal. If sampleportal is not installed, and if you have set up a custom *desktopType*, for example, `foo`, then the new container directory must be created directory under `foo`.

Either copy the modified JSP or template files here, or place your newly created files here. If you use a sample container without changing any content or file names, you do not need to create a new subdirectory nor copy any files there. (In the example that follows, a new subdirectory is needed, because a new container is created.)

For example, let's say you create a new container called newSingleContainer whose display profile definition is the following:

```
<Container name="newSingleContainer" provider="JSPSingleContainer">
    <Properties>
        <String name="helpURL" value="desktop/newSingle.html"/>
        <String name="title" value="A new single container"/>
        <String name="contentPage" value="newsinglecontent.jsp"/>
        <Boolean name="isEditable" value="true"/>
        <String name="editType" value="edit_subset"/>
    </Properties>
    <Available/>
    <Selected/>
    <Channels/>
</Container>
```

Because the file specified for the `contentPage` property is different from the `contentPage` value for the provider definition, you need to create a new directory under the `/etc/opt/SUNWps/desktop/default` directory called `newSingleContainer`. You then only need to copy the `newsinglecontent.jsp` file to this new directory. The system is able to locate all other JSPs referenced by the JSPSingleContainer provider.

If desired, rather than customizing the sample portal JSP and template files directly, you can create a separate directory for your organization's customized files, and perform customizations on those files. This preserves the initially installed portal JSP and template files.

➤ **To Create Customized Organization JSP and Template Files**

1. Change directories to the Desktop JSP or template directory.

   For example,

   ```
   cd /etc/opt/SUNWps/desktop
   ```

2. Create a new directory for your organization's JSPs and templates.

   For example,

   ```
   mkdir sesta
   ```

3. Copy the JSPs and templates that you wish to modify into the new directory location, maintaining the same directory structure.

   For example, if your new Desktop type will modify `/etc/opt/SUNWps/desktop/default/JSPProvider/content.jsp`, copy this file to `/etc/opt/SUNWps/desktop/sesta/JSPProvider/content.jsp`, and customize the file for the new Desktop type in that location.

4. Customize the JSPs templates in the `sesta` directory as required.

5. Change the dynamic Desktop Type attribute in the Access Manager software administration console to use the newly created directory.

   See "Changing the Desktop Type" on page 20 for more information.

Loading the Display Profile at the Appropriate LDAP Nodes

Load the display profile at the appropriate LDAP node(s) by using the `dpadmin` command. You can also use the Edit Display Profile XML text box in the Access Manager software administration console (as long as you are not using Netscape 4.7x) or the Upload link.

See the *Portal Server Administration Guide* for more information on the `dpadmin` command, and the Edit Display Profile XML text box and Upload link.

(Optional) Creating a Resource Bundle

If you created a new provider, you may need to create a resource bundle file with the same name as the provider.

Accessing the Desktop

Access the Desktop in one of the following ways.

➤ **To Access the Desktop**

1. Use a specific container or channel reference by using the provider argument to the Desktop login URL:

   `http://hostname:port/portal/dt?provider=`*`providername`*

   where *providername* is one of the providers. For example, to access the JSP-based tab Desktop, in a browser, type:

   `http://hostname:port/portal/dt?provider=JSPTabContainer`

2. If no channel is referenced, the Desktop looks in the session for the last channel or container that was displayed. (This is stored in the session.)

3. If no channel is stored in the session, the Desktop looks in a Desktop service attribute for the top-level container to display (Default Channel Name attribute). This happens after an initial login.

Once this top-level container is determined, that container draws the containers or channels that it references (through the Selected list), until all of its leaves have been reached.

Debugging the Desktop

Use the following to help debug the Desktop environment:

The Desktop debug file is located at:

`/var/opt/SUNWam/debug/desktop.debug`

If you get an error message page on the Desktop, you can view the source to look at the stack trace. (For example, in Netscape Navigator, select Page Source from the View menu.)

Chapter 4

Customizing Container Tabs

This chapter provides a variety of tasks to customize the container tabs. It contains the following sections:

- Adding a Tab to JSPTabContainer
- Creating a Tab Within a Tab
- Stretching a Tab Across an Entire Container
- Changing the Tab Image for JSP-based Tab Containers
- Changing the Color of Tabs
- Making a Tab the Start Tab
- Adding a Role-Based Tab
- Adding a Channel to a User-defined Tab

NOTE The order that you list the tabs in the display profile is the order that tabs are displayed in the Desktop. So, to make a tab the first tab in the user's Desktop, you need to move it to be first in the selected list in the display profile.

Adding a Tab to JSPTabContainer

A tab can be any container type, but, by default, the sample portal uses table container. To add a new tab, you must first define the container, then register that container in JSPTabContainer, which "houses" the tabs.

➤ **To Add a Tab to JSPTabContainer**

1. Create the necessary display profile.

a. Define the new collection within <Collection name="TabProperties"> in JSPTabContainer, for example:

```
...
    <Collection name="NewTabPanelContainer">
        <Boolean name="removable" value="false"/>
        <Boolean name="renamable" value="true"/>
        <Boolean name="predefined" value="true"/>
    </Collection>
    ...
</Collection>
...
```

b. Add entries to the <Available> and <Selected> tags, for example:

```
...
    <Available>
        <Reference value="NewTabPanelContainer"/>
        ...
    </Available>
...
    <Selected>
        <Reference value="NewTabPanelContainer"/>
        ...
    </Selected>
...
```

c. Define a container for NewTabPanelContainer, for example:

```
<Container name="NewTabPanelContainer"
provider="JSPTableContainerProvider">
    <Properties>
        <String name="title" value="New Container Channel"/>
        <String name="contentPage" value="tabtable.jsp"/>
        <String name="description" value="This is a test for front table
containers"/>
        <String name="Desktop-fontFace1" value="Sans-serif"/>
        <Collection name="categories">
            <String value="Personal Channels"/>
            <String value="Sample Channels"/>
        </Collection>
        <Collection name="Personal Channels">
            <String value="UserInfo"/>
```

```
            <String value="MailCheck"/>
        </Collection>
        <Collection name="Sample Channels">
            <String value="SampleJSP"/>
            <String value="SampleXML"/>
        </Collection>
    </Properties>
    <Available>
        <Reference value="UserInfo"/>
        <Reference value="MailCheck"/>
        <Reference value="SampleJSP"/>
        <Reference value="SampleXML"/>
    </Available>
    <Selected>
        <Reference value="UserInfo"/>
        <Reference value="MailCheck"/>
        <Reference value="SampleJSP"/>
        <Reference value="SampleXML"/>
    </Selected>
    <Channels>
        ...
    </Channels>
</Container>
```

d. If predefined property value is true in the `TabProperties` collection (Step a on page 32), then it is recommended to define a Provider for the container channel which is meant to be used as a predefined tab.

For example:

Code Example 4-1 PredefinedNewTabPanelContainerProvider Display Profile Definition

```
<Provider name="PredefinedNewTabPanelContainerProvider"
class="com.sun.portal.providers.containers.jsp.table.JSPTableContainerProvider"
version="2">
    <Properties>
        <ConditionalProperties condition="locale" value="en" >
            <ConditionalProperties condition="locale" value="US" >
                <String name="title" value="New Sample"/>
                <String name="description" value="New Tab"/>
            </ConditionalProperties>
        </ConditionalProperties>
        <String name="title" value="New Sample"/>
        <String name="description" value="New Tab"/>
        <String name="contentPage" value="tabtable.jsp"/>
        <String name="presetThemeChannel" value="JSPPresetThemeContainer" advanced="true"/>
        <String name="customThemeChannel" value="JSPCustomThemeContainer" advanced="true"/>
        <String name="parentTabContainer" value="JSPTabContainer" advanced="true"/>
        <String name="Desktop-fontFace1" value="Sans-serif"/>
```

Code Example 4-1 PredefinedNewTabPanelContainerProvider Display Profile Definition *(Continued)*

```
        <String name="refreshTime" value="" advanced="true"/>
        <String name="width" value="thin" advanced="true"/>
        <String name="fontFace1" value="Sans-serif"/>
        <String name="productName" value="Sun Java System Portal Server"/>
        <String name="maximizedChannel" value=""/>
        <Integer name="timeout" value ="240"/>
        <Integer name="layout" value="1"/>
        <Boolean name="showExceptions" value="false"/>
        <Boolean name="parallelChannelsInit" value="false"/>
        <Boolean name="refreshParentContainerOnly" value="false" advanced="true"/>
        <Boolean name="isEditable" value="true" advanced="true"/>
        <String name="editType" value="edit_complete" advanced="true"/>
        <String name="editContainerName" value="JSPEditContainer" advanced="true"/>
        <Integer name="thin_popup_height" value="200"/>
        <Integer name="thin_popup_width" value="500"/>
        <Integer name="thick_popup_height" value="300"/>
        <Integer name="thick_popup_width" value="600"/>
        <Integer name="fullwidth_popup_height" value="500"/>
        <Integer name="fullwidth_popup_width" value="600"/>
        <Boolean name="defaultChannelIsMinimizable" value="true"/>
        <Boolean name="defaultChannelIsMaximizable" value="true"/>
        <Boolean name="defaultChannelIsMinimized" value="false" advanced="true"/>
        <Boolean name="defaultChannelIsDetached" value="false" advanced="true"/>
        <Boolean name="defaultChannelIsDetachable" value="true"/>
        <Boolean name="defaultChannelIsRemovable" value="true"/>
        <Boolean name="defaultChannelHasFrame" value="true" advanced="true"/>
        <Boolean name="defaultChannelIsMovable" value="true"/>
        <Boolean name="defaultBorderlessChannel" value="false" advanced="true"/>
        <String name="defaultChannelColumn" value="1" advanced="true"/>
        <String name="defaultChannelRow" value="1" advanced="true"/>
        <Collection name="categories">
            <String value="Sample Channels"/>
        </Collection>
        <Collection name="Sample Channels">
            <String value="SampleRSS"/>
            <String value="SampleURLScraper"/>
            <String value="Notes"/>
            <String value="SampleSimpleWebService"/>
        </Collection>
        <Collection name="channelsColumn" advanced="true">
            <String name="SampleURLScraper" value="2"/>
            <String name="Notes" value="2"/>
            <String name="SampleSimpleWebService" value="2"/>
        </Collection>
        <Collection name="channelsRow" advanced="true">
            <String name="SampleURLScraper" value="2"/>
            <String name="Notes" value="3"/>
            <String name="SampleSimpleWebService" value="4"/>
        </Collection>
        <Collection name="channelsIsMinimized" advanced="true"/>
        <Collection name="channelsIsDetached" advanced="true"/>
        <Collection name="channelsHasFrame" advanced="true"/>
        <Collection name="channelsIsMinimizable"/>
        <Collection name="channelsIsMaximizable"/>
```

Code Example 4-1 PredefinedNewTabPanelContainerProvider Display Profile Definition *(Continued)*

```
        <Collection name="channelsIsMovable"/>
        <Collection name="channelsIsRemovable"/>
        <Collection name="channelsIsDetachable"/>
        <Collection name="borderlessChannels"/>
    </Properties>
</Provider>
```

e. Define the container channel based on the PredefinedNewTabPanelContainerProvider.

When the user creates a new tab based on the predefined tab, all the properties for this tab are picked up from the Provider definition. For example:

Code Example 4-2 PredefinedNewTabPanelContainer Channel Properties

```
<Container name="PredefinedNewTabPanelContainer"
provider="PredefinedNewTabPanelContainerProvider">
    <Properties/>
    <Available>
        <Reference value="SampleRSS"/>
        <Reference value="SampleURLScraper"/>
        <Reference value="Notes"/>
        <Reference value="SampleSimpleWebService"/>
    </Available>
    <Selected>
        <Reference value="SampleRSS"/>
        <Reference value="SampleURLScraper"/>
        <Reference value="Notes"/>
        <Reference value="SampleSimpleWebService"/>
    </Selected>
    <Channels>
    </Channels>
</Container>
```

2. Load the display profile into LDAP by using the `dpadmin` command.
3. Bring up the Desktop and verify that the tab was added.

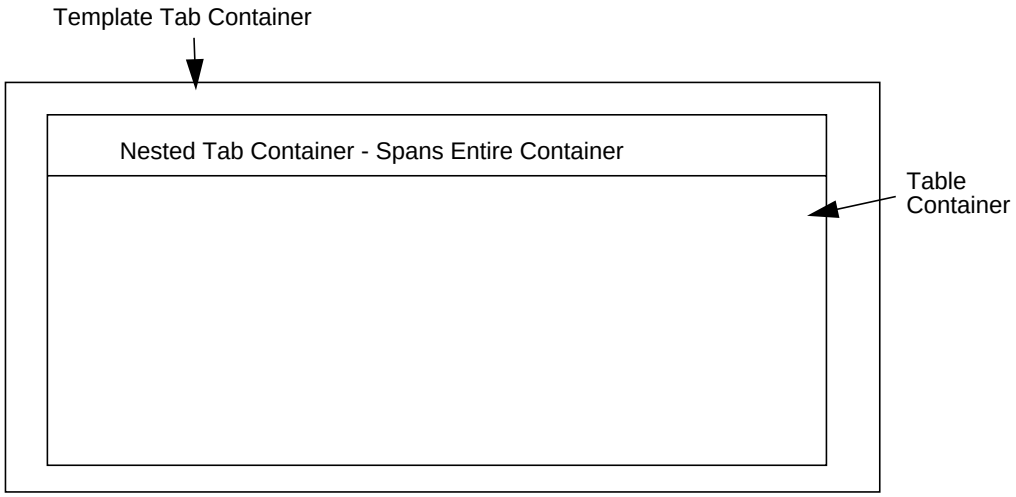
Creating a Tab Within a Tab

This is similar to "Adding a Tab to JSPTabContainer" on page 31, except that instead of defining the tab based on JSPTableContainerProvider, you base the new tab on JSPTabContainerProvider.

Stretching a Tab Across an Entire Container

Figure 4-1 shows an example of a tab that spans an entire container. In this figure, the top-level container is a template tab container. Nested within it is a table container, and nested within that is a tab container, in which the tab spans the entire container.

Figure 4-1 Tab Spanning Entire Container



➤ **To Stretch a Tab Across an Entire Container**

1. Edit the display profile and make the width of the Nested Tab Container full_top in the Table Container so that it stretches across the entire page.
2. Load the display profile into LDAP by using the `dpadmin` command.

Changing the Tab Image for JSP-based Tab Containers

You can customize the look of tabs as they use images.

➤ **To Change the Tab Image for JSP-based Tab Containers**

1. Log in to the Access Manager administration console.
2. Select Services from your Organization View pull-down menu and select Portal Desktop.
3. Select Edit XML to directly edit the display profile XML fragment. Or,
   a. Select Manage Channels and Containers Link and Edit Properties for Display Profile.
   b. Select Global Themes and the theme you wish to modify.
4. Change the value of the `tabNotchImage` property to the new image name.

   By default, the value for this property is `tabNotch.gif`.
5. Copy the new image into *PortalServer-base*`/web-src/desktop/tabs/images` directory.
6. Run the *PortalServer-base*`/bin/deploy redeploy -deploy_admin_password` *password* command to deploy the new image.
7. Reload the Desktop to verify the change.

Changing the Color of Tabs

The background color of tabs are part of the themes.

➤ **To Change the Color of Tabs**

1. Log in to the Access Manager administration console.
2. Select Services from your Organization View pull-down menu and select Portal Desktop.
3. Select Edit XML to directly edit the display profile XML fragment. Or,
   a. Select Container and Channel Management Link and Edit Properties for Display Profile.

b. Select Edit Collection for Global Themes and the theme you wish to modify.

4. Change the value of the `titleBarColor` property to change the color of the selected tab and / or change the value of `tabColor` property to change the color of an unselected tab.

   The selected tab background color is the same as the title bar color.

5. Reload the Desktop to verify the change.

Making a Tab the Start Tab

The “Start tab” is the tab that is highlighted when user first logs in.

➤ **To make the tab the start tab**

1. Edit the display profile for the appropriate container.
2. Change the `startTab` property to the tab to highlight when the user logs in. For example:

   ```
   <String name="startTab" value="MyFrontPageTabPanelContainer"/>
   ```

3. Load the display profile into LDAP by using the `dpadmin` command.

Adding a Role-Based Tab

When you have a display profile that provides a set of JSP nested tabs, and you want a user who belongs to this role and another role to add an additional sub-tab, you can merge multiple display profiles to accomplish this. In the additional display profile, you specify the additional sub-tab in the like-named container for JSPTabContainer with `merge=fuse` and JSPTableContainer for the new tab.

➤ **To Add a Role-based Tab**

1. To add a role-based tab for a user, define a role level display profile, which has the JSPTabContainer definition.
2. Add the role-based tab to the available and selected list with merge=fuse in the channel definition for the table container.

   When the user is added to this role, the new tab is visible. The following display profile XML fragments show the role definitions.

```
role1
<Container name="JSPTableContainer" provider="JSPTableContainerProvider"
merge="fuse">
    <Properties> ... </Properties>
    <Available> ... </Available>
    <Selected merge="fuse">
        <Reference value="Outages"/>
        <Reference value="SolarisAdmin"/>
        <Reference value="AdminTipoftheDay"/>
    </Selected>
</Container>
role2
<Container name="Front" provider="front" merge="fuse">
    <Properties> ... </Properties>
    <Available> ... </Available>
    <Selected merge="fuse">
        <Reference value="Benefits"/>
        <Reference value="EmployeeNews"/>
    </Selected>
</Container>
```

The user belonging to both role1 and role2 receives the following display profile:

```
Container name="JSPTableContainer" provider="JSPTableContainerProvider"
merge="fuse">
    <Properties> ... </Properties>
    <Available> ... </Available>
    <Selected merge="fuse">
        <Reference value="Outages"/>
        <Reference value="SolarisAdmin"/>
        <Reference value="AdminTipoftheDay"/>
        <Reference value="Benefits"/>
        <Reference value="EmployeeNews"/>
    </Selected>
</Container>
```

Adding a Channel to a User-defined Tab

Users can add a new tab to their Desktop by using the Tabs link and then by clicking the Make a New Tab link. The channel list that gets displayed on the content page which is shown when the user selects to create a new tab from scratch is picked up from the JSPTabCustomTableContainer's Available list.

Chapter 5

Customizing Channels

This chapter describes how to customize channels. This chapter contains the following sections:

- Customizing Channel Refresh Times and Container Caching
- Customizing Window Preference
- Removing a Button
- Changing the Channel Layout for a Table Container
- Removing the Title Bar from a Channel
- Changing the Channel Border Width
- Customizing the Channel Border

See the *Portal Server Administration Guide* for instructions on adding a channel to the Desktop.

Customizing Channel Refresh Times and Container Caching

The `refreshTime` property controls how often a channel's content is reloaded. When `refreshTime` is set to `0` (the default) for the container, the browser refresh (or reload) causes the page to be reloaded and the `getContent()` method is called again for every channel.

The following applies to a single channel:

- It is not possible to refresh only the content of the single channel within a container because a channel is an HTML table cell.

- It is possible to use the `DesktopURL()` method in the PAPI. The provider can use `getDesktopURL()` to get the Desktop servlet's URL, append arguments to it, and generate a new URL (or link).

The following applies to controlling and configuring container caching:

- Use the `refreshTime` property for the container along with the `refreshTime` for individual channels within the container.
- If the `refreshTime` for the container is blank, it is calculated to be the minimum time for all of the contained channels. If you want to override that calculated time, set a `refreshTime` for the container and then the content for the whole container will be cached.

NOTE If you have a large number of channels, utilize the provider caching by setting the `refreshTime` to a large number so that the portal page can use cached content. This makes sense when most of your channels have static content. The way the `refreshTime` works is if the container's `refreshTime` is set, it will use it. If `refreshTime` is set to an empty string, it will try to get and use the minimum of the `refreshTime` of its selected channels.

Customizing Window Preference

For channels that include links that launch another browser, you can control how this browser window is opened.

➤ **To Customize the Channel Window Preference**

1. Define the display profile (either for the channel, to make the change for only that channel, or for the provider, to make the change for every channel that uses the provider) so that it includes the windowPref property.

   For example:

```
<Properties>
    ...
    <String name="windowPref" value="all_new"/>
    ...
</Properties>
```

The values are:

- `all_new` (New window is opened for every link)
- `one_new` (All links open on the same new window)
- `same` (Desktop window)

2. Load the display profile into LDAP by using the `dpadmin` command.

NOTE The intelligence has to be built with the help of JavaScript for that particular channel. This is supported by BookmarkProvider only.

Removing a Button

➤ **To Remove a Button From All Channels in a Container**

1. Find the container you want to work with. If you are working with one of the sample portals, you need to modify the appropriate "contained" container, which is part of the top-level container. See the *Portal Server Technical Reference Guide* for a list of the contained containers.

2. Add the appropriate property (within the `<Properties> </Properties>`) tags from Table 5-1 to the container's display profile for the button you want to remove. This two column table lists the button in the first column and the property to hide the button in the second column.

   The order of the buttons in this table corresponds to the order they appear in the channel, from left to right: Minimize, Maximize, Help, Edit, Detach, and Remove.

Table 5-1 Channel Buttons and Corresponding Properties

Button	Property to Hide the Button
Minimize	`<Boolean name="defaultChannelIsMinimizable" value="false"/>`
Maximize	`<Boolean name="defaultChannelIsMaximizable" value="false"/>`
Help	`<String name="helpURL" value=""/>`
Edit	`<Boolean name="isEditable" value="false"/>`
Detach	`<Boolean name="defaultChannelIsDetachable" value="false"/>`

Table 5-1 Channel Buttons and Corresponding Properties *(Continued)*

Button	Property to Hide the Button
Remove	`<Boolean name="defaultChannelIsRemovable" value="false"/>`

NOTE For the Help and Edit buttons, You must insert the respective property for each channel. You cannot insert the property within the container's `<Properties> </Properties>` tags.

Make sure the following properties are not defined in the container:

```
<Collection name="channelsIsRemovable">..</Collection>
<Collection name="channelsIsMinimizable"/>..</Collection>
<Collection name="channelsIsMaximizable"/>..</Collection>
<Collection name="channelsIsDetachable"/>..</Collection>
```

3. Load the display profile into LDAP by using the `dpadmin` command.

➤ To Remove a Button From a Single Channel

1. For the channel from which you want to remove a button, add the appropriate property to a `Collection` tag in the container that contains the channel. See Table 5-2, for the button you want to remove. This two column table lists the button in the first column and the property to hide the button in the second column

   The order of the buttons in this table corresponds to the order they appear in the channel, from left to right: Minimize, Maximize, Help, Edit, Detach, and Remove.

Table 5-2 Channel Buttons and Corresponding Properties

Button	Property to Hide the Button

Minimize	`<Collection name="channelsIsMinimizable">`
	`<Boolean name="channelname" value="false"/>`
	`</Collection>`

Table 5-2 Channel Buttons and Corresponding Properties *(Continued)*

Button	Property to Hide the Button

Maximize	`<Collection name="channelsIsMaximizable">`
	`<Boolean name="channelname" value="false"/>`
	`</Collection>`

Detach	`<Collection name="channelsIsDetachable">`
	`<Boolean name="channelname" value="false"/>`
	`</Collection>`

Remove	`<Collection name="channelsIsRemovable">`
	`<Boolean name="channelname" value="false"/>`
	`</Collection>`

2. For the channel in which you want to remove a button, add the appropriate property to a Collection tag in the controlling container.

   For example, use the following XML to hide the Remove button for the Sample JSP channel in the JSP table container, MyFrontPageTabPanelContainer, whose container is JSPTabContainer.

```
<Container name="MyFrontPageFramePanelContainer"
provider="JSPTableContainerProvider">
    <Properties>
        ...
         <Collection name="channelsIsRemovable">
             <Boolean name="SampleJSP" value="false"/>
         </Collection>
    </Properties>
    ...
```

3. Load the display profile into LDAP by using the `dpadmin` command.

Changing the Channel Layout for a Table Container

➤ **To Change the Channel Layout for a Table Container**

1. You can change the layout for a particular table container by modifying (or adding) the following property in the table container's display profile:

   ```
   <Integer name="layout" value="value"/>
   ```

   where value is:

   1 = Thin-wide, two columns

   2 = Wide-thin, two columns

   3 = Thin-wide-thin, three columns

2. Reload the display profile to LDAP by running the `dpadmin` with the `modify` command, loading it at the top-most node in the directory by using the `-g` option.

   For example:

   ```
   dpadmin modify -u "uid=amAdmin,ou=People,dc=sesta,dc=com" -w password
   -g dp-providers.xml
   ```

NOTE You can also use the Sun Java System Access Manager software administration console Channel and Container Management link for the Desktop service to change the channel layout.

Removing the Title Bar from a Channel

➤ **To Remove the Title Bar from a Channel**

1. Add the following to the table container display profile in which the channel is present.

```
<Collection name="channelsHasFrame">
<Boolean name="channelname" value="false"/>
</Collection>
```

2. Load the display profile into LDAP by using the `dpadmin` command.

   See "Editing the Display Profile."

Changing the Channel Border Width

➤ **To Change the Border Width for all Channels in a Container**

1. Change directories to the appropriate table container (JSPFrameCustomTableContainerProvider, JSPTabCustomTableContainerProvider, or JSPTableContainerProvider) for which you want to change the border width.

   For example:

   ```
   cd /etc/opt/SUNWps/desktop/default/JSPTableContainerProvider
   ```

2. Edit the `leafWrapper.jsp` file.

   Change the cellpadding to not use the `borderWidth` property from theme. The line to change is:

   ```
   CELLPADDING=<dttheme:getAttribute name="borderWidth"/>
   ```

   For example, you might use `CELLPADDING=2` in place of this line.

3. Run the `touch` command on the `tablecolumn.jsp` file, which includes `leafWrapper.jsp`.

   ```
   touch tablecolumn.jsp
   ```

NOTE You can also change the `borderWidth` property for the GlobalThemes Collection. This changes the width of the channel borders for a theme. Users can then select the theme from the Themes page. (This method removes the need to change JSPs.) After making the change to the display profile, which in the sample portal is `dp-org.xml`, you need to upload it to LDAP.

Customizing the Channel Border

➤ **To Customize Channel Borders to Have Bevelled Edges, Shadows, Curved Corners, and So On**

1. Change directories to the appropriate table container (JSPFrameCustomTableContainerProvider, JSPTabCustomTableContainerProvider, or JSPTableContainerProvider) for which you want to customize the border width.

   For example:

   ```
   cd /etc/opt/SUNWps/desktop/default/JSPTableContainerProvider
   ```

2. Edit leafWrapper.jsp.

   Change the appropriate HTML. You change border width and color by modifying the following lines:

   ```
   CELLPADDING=<dttheme:getAttribute name="borderWidth"/>
   BGCOLOR="<dttheme:getAttribute name="borderColor"/>"
   ```

   To add curved borders, shadows, and so on, you use a combination of image maps and tables. While possible, such an example is beyond the scope of this document.

3. Run the `touch` command on the `tablecolumn.jsp` file, which includes `leafWrapper.jsp`.

   ```
   touch tablecolumn.jsp
   ```

Chapter 6

Customizing Instant Messaging

This chapter contains the following sections:

- Disabling the User from Editing Instant Messaging Server Information
- Automatically Closing the Instant Messaging Invite Window
- Customizing Display of Instant Messaging Contacts

Disabling the User from Editing Instant Messaging Server Information

When a site has only a single Sun Java System Instant Messaging server, there is no need for the end-users to be able to edit the server information in the channel edit page. The server information can be removed by editing the `IMEdit.jsp` file so that part of the page is not displayed.

Code Example 6-1 contains the Instant Messaging Server Information (snippet) in the `IMEdit.jsp` file.

Code Example 6-1 Instant Messaging Server Information in IMEdit.jsp File

```
...
<%-- ------------------- Instant Messaging Server Information --------------%>
<tr>
    <td width="100%" bgcolor="#333366" colspan="3">
        <font size="+1" face="<%=fontFace%>" color="#ffffff"><b>Instant Messaging Server
Information</b></font>
    </td>
</tr>
<tr>
    <td colspan="2" width="70%"></td>
    <td valign="top" rowspan="5">
```

Code Example 6-1 Instant Messaging Server Information in IMEdit.jsp File *(Continued)*

```
        <table border="0" cellpadding="2" cellspacing="1" width="100%" bgcolor="#eeeeee">
            <tbody>
                <tr>
                    <td valign="top" bgcolor="#dcdcdc" height="15"
                        <font face="<%=fontFace%>"><b>Why?</b></font>
                    </td>
                </tr>
                <tr>
                    <td valign="top">
                        <font face="<%=fontFace%>" size="-1">In order to access a Sun Java
System Instant Messaging server, you must enter the information needed to contact the
server.</font>
                    </td>
                </tr>
            </tbody>
        </table>
    </td>
</tr>
<tr>
    <td valign="middle" align="right"><font face="<%=fontFace%>" color="#000000"><label
for="commonname">
        <b>Server:</b>
    </label></font><br></td>
    <td valign="middle">
        <input type="text" name="server" size="25" maxlength="40"
value="<dtpc:getStringProperty key="server"/>">
    </td>
</tr>
<tr>
    <td valign="middle" align="right"><font face="<%=fontFace%>" color="#000000"><label
for="commonname">
        <b>Server Port:</b>
    </label></font><br></td>
    <td valign="middle">
        <input type="text" name="port" size="25" maxlength="40"
value="<dtpc:getStringProperty key="port"/>">
    </td>
</tr>
<tr>
    <td valign="middle" align="right"><font face="<%=fontFace%>" color="#000000"><label
for="commonname">
        <b>Multiplexor:</b>
    </label></font><br></td>
    <td valign="middle">
        <input type="text" name="mux" size="25" maxlength="40"
value="<dtpc:getStringProperty key="mux"/>">
    </td>
</tr>
<tr>
    <td valign="middle" align="right"><font face="<%=fontFace%>" color="#000000"><label
for="commonname">
        <b>Multiplexor Port:</b>
```

Code Example 6-1 Instant Messaging Server Information in IMEdit.jsp File *(Continued)*

```
    </label></font><br></td>
    <td valign="middle">
        <input type="text" name="muxport" size="25" maxlength="40"
value="<dtpc:getStringProperty key="muxport"/>">
    </td>
</tr>
<dtpc:getStringProperty id="authMethod" key="authMethod"/>
<jx:if test="$authMethod == 'ldap'">
<tr>
    <td valign="middle" align="right"><font face="<%=fontFace%>" color="#000000"><label
for="commonname">
        <b>Username:</b>
    </label></font><br></td>
    <td valign="middle">
        <input type="text" name="username" size="25" maxlength="40"
value="<dtpc:getStringProperty key="username"/>">
    </td>
</tr>
<tr>
    <td valign="middle" align="right"><font face="<%=fontFace%>" color="#000000"><label
for="commonname">
        <b>Password:</b>
    </label></font><br></td>
    <td valign="middle">
        <input type="password" name="password" size="25" maxlength="40" value="<%=
JSPProvider.getDummyPassword() %>">
    </td>
</tr>
</jx:if>
...
```

The change to this file depends on the type of authentication that is being used for the channel. With Sun Java System Access Manager software authentication (`authMethod=idsvr`), the entire Instant Messaging server Information section can be removed. With LDAP authentication (`authMethod=ldap`), the `username` and `password` fields are still necessary. So, remove only the `server` and `port` fields and keep the rest of the Instant Messaging Server Information section.

Automatically Closing the Instant Messaging Invite Window

The `invite.jsp` file generates the content for the popup window that is created when a user is invited to a conference in an instant messaging client that is already running. It is necessary to open this window (the window is opened before the server checks if the instant messaging client is running), but the window can be closed automatically. The `invite.jsp` file has the Javascript code for doing this commented out (see Code Example 6-2). Merely uncomment this Javascript code (shown in Code Example 6-2 in bold) to cause the window to close automatically.

Code Example 6-2 Instant Messaging `invite.jsp` File

```
<%@ taglib uri="/tld/jx.tld" prefix="jx" %>
<%@ taglib uri="/tld/im.tld" prefix="im" %>
<%@ taglib uri="/tld/desktop.tld" prefix="dt" %>
<%@ taglib uri="/tld/desktopProviderContext.tld" prefix="dtpc" %>
<dt:obtainChannel channel="$JSPProvider">
<html>
    <head>
        <title>Sun Java System Instant Messenger</title>
    </head>
    <body bgcolor="#ffffff">
        <p>
            User <%= request.getParameter("username") %> is being invited to join a
conference using the IM client that you already have running.
        <p>
            <center>
                <form>
                    <input type=button value="Ok" onClick="window.close();">
                </form>
            </center>
        <%-- uncomment this script if you want the window to close automatically
             <script>window.close();</script>
        --%>
```

Customizing Display of Instant Messaging Contacts

By default, the `IMContent.jsp` file only displays presence information for on-line contacts. There is code in the JSP file for displaying presence information for all contacts in the selected contact group(s). If you wish to display all of this information, uncomment the code (shown in Code Example 6-3 on page 53) in `IMContent.jsp` file.

Code Example 6-3 Instant Messaging Offline Contact Information in `IMContent.jsp` File

```
...
<%--Offline States - Uncomment this if you want users to see offline contacts -
<jx:when test="$p=='CLOSED'">
<jx:expr value="$anchortag"/>
    <img src="im/images/Offline_13Padded.gif" alt="Offline - Click to chat" border=0
align=absmiddle></a>
    <im:getContactName/></br>
</jx:when>
<jx:when test="$p=='AWAY'">
<jx:expr value="$anchortag"/>
    <img src="im/images/Away_13Padded.gif" alt="Away - Click to chat" border=0
align=absmiddle></a>
    <im:getContactName/></br>
</jx:when>
<jx:when test="$p=='FORWARDED'">
<jx:expr value="$anchortag"/>
    <img src="im/images/Forwarded_13Padded.gif" alt="Offline - Click to chat" border=0
align=absmiddle></a>
    <im:getContactName/></br>
</jx:when>
<jx:when test="$p=='OTHER'">
<jx:expr value="$anchortag"/>
    <img src="im/images/Offline_13Padded.gif" alt="Other - Click to chat" border=0
align=absmiddle></a>
    <im:getContactName/></br>
</jx:when>
<jx:otherwise>
<jx:expr value="$anchortag"/>
    <img src="im/images/Offline_13Padded.gif" alt="<%=p%> - Click to chat" border=0
align=absmiddle></a>
    <im:getContactName/></br>
</jx:otherwise>
--  End of offline states --%>
```

Customizing the Anonymous Desktop

This chapter describes customizations you can make for the anonymous Desktop. This chapter contains the following sections:

- Configuring Anonymous Authentication
- Accessing the Anonymous Desktop
- Disabling the Initial Access Manager Software Login Page and Always Use Anonymous Log In
- Modifying the Anonymous Banner and Menu Bar
- Adding the Login Channel to the Anonymous Desktop of a Newly Created Organization

When you install the sample portal, a copy of the anonymous Desktop display profile is located in the *PortalServer-base*/`samples/desktop/dp-anon.xml` file, with the support files located in the `/etc/opt/SUNWps/desktop/anonymous` directory.

Configuring Anonymous Authentication

Sun Java System Portal Server software supports two methods for implementing anonymous authentication:

- Authentication-less User ID attributes - Users accessing the Desktop URL are granted access to the default Desktop.
- Anonymous user session - Users select Anonymous from the Authentication menu, log in as the user anonymous, and are granted access to the Desktop.

When you install Portal Server software, by default the installation program enables anonymous authentication to the Desktop of the default organization using the Authentication-less User ID attributes. To implement this feature, the installation program creates a user account, authlessanonymous, and sets up access for this user within the following two Desktop Services global attributes:

- Authorized Authentication-less User IDs
- Default Authentication-less User ID

This section describes how to enable and disable both types of anonymous authentication. See the Administering Users And Services chapter in the *Portal Server Administration Guide* for more information on enabling and disabling anonymous authentication.

➤ To Enable Anonymous Log In

1. Log in to the Sun Java System Access Manager software administration console as administrator.
2. Register the Anonymous service for the selected organization and create its template.
3. Add Anonymous to the Authentication menu in the Core service (for the selected organization).
4. Create the anonymous user account for the selected organization.

➤ To Disable Anonymous Log In

1. Log in to the Access Manager software administration console as administrator.
2. Unregister the Anonymous service for the selected organization.
3. Remove Anonymous from the Authentication menu in the Core service (for the selected organization).
4. Remove the anonymous user account for the selected organization.

➤ To Enable Authentication-less (authlessanonymous) Log In

1. Log in to the Access Manager software administration console as administrator.
2. Create the auhthlessanonymous account with a password of authlessanonymous for the selected organization.
3. Select the Service Configuration tab.

4. Click on the Desktop node.

   The Desktop attributes page appears in the data pane.

5. Add the following value to the Authorized Authentication-less user IDs attribute:

   ```
   uid=authlessanonymous,ou=People,dc=organization|authlessanonymous
   ```

   Substitute the appropriate organization name for organization.

6. Set the Default Authentication-less user ID attribute to the following:

   ```
   uid=authlessanonymous,ou=People,dc=organization
   ```

   Substitute the appropriate organization name for organization.

7. Log out from the Access Manager software administration console.

8. Verify that authentication-less authentication works. That is, close all current browsers and start a new browser with the following URL:

   ```
   http://hostname:port/portal/dt
   ```

   The anonymous Desktop will be displayed.

➤ **To Disable Authentication-less (authlessanonymous) Log In**

By default, the sample portal is registered for Authentication-less (authlessanonymous) authentication. This is different from Anonymous authentication, which the sample portal, by default, is not registered for. The Anonymous Desktop uses Portal Server software for authentication; the Authless Desktop does not pass through the authentication process at all and is handled internally in the Desktop servlet.

To disable authentication-less log in:

1. Log in to the Access Manager software administration console as administrator.

2. Select the Service Configuration tab.

3. Click on the Desktop node.

   The Desktop attributes page is displayed in the data pane.

4. Remove the value(s) from the Authorized Authentication-less user IDs attribute.

5. Remove the value from the Default Authentication-less user ID attribute so that it is blank.

6. Log out from the Access Manager software administration console.
7. Verify that you cannot reach the Anonymous Desktop. That is, close all current browsers and start a new browser with the following URL:

   `http://hostname:port/portal/dt`

   The Anonymous Desktop should not appear. Instead, the Login page, presenting the various authentication modules you have configured, should appear.

Accessing the Anonymous Desktop

➤ **To Access the Anonymous Desktop through the Access Manager Host Name (obj.conf File)**

To enable users to access the Anonymous Desktop without typing the fully qualified domain name, you need to modify the *Web-Container-Instance*`/config/obj.conf` file.

1. Edit the web server's `obj.conf` file.
2. After the line Object `name=default`, which is at the top, add the following lines, depending on whether you want authentication-less (authless anonymous) or anonymous access.

   For authentication-less (authlessanonymous):

```
NameTrans fn="redirect" from="/index.html"
url="http://hostname:port/portal/dt?desktop.suid=uid=authlessanonymous,ou=Peop
le,dc=organization"
```

   For anonymous:

```
NameTrans fn="redirect" from="/index.html"
url=http://hostname:port/amserver/login?org=organization&module=Anonymous
```

   For a specific organization:

```
NameTrans fn="redirect" from="/index.html"
url="http://hostname:port/amserver/login?organization
```

Make sure psservername is the fully qualified domain name of your Portal Server software host, and organization is the name of the appropriate Access Manager software organization.

NOTE For anonymous, you must also make sure that only Anonymous authentication is enabled in the Access Manager software administration console.

3. Save the file and restart Portal Server software.

   ```
   /etc/init.d/amserver start
   ```

   Users can now view the Anonymous Desktop by typing the Portal Server software host name in their browser. The fully qualified domain name is no longer required.

➤ To Access the Anonymous Desktop through the Portal Server Host Name (index.html File)

To access the Desktop login page using a URL in the following form http://*psservername*, add some JavaScript™ to the web server's index.html file.

1. Add the following Javascript to the index.html file.

```
<HTML>
    <HEAD>
        <SCRIPT>
            document.location.href="/portal/dt?desktop.suid=uid=authlessano
nymous,ou=People,dc=organization,dc=com" <-- for authless anonymous
        </SCRIPT>
    </HEAD>
</HTML>
```

This example assumes that /portal/dt is the user's redirect URL.

2. Verify that you can now access the Desktop by just typing the server name in the browser.

Disabling the Initial Access Manager Software Login Page and Always Use Anonymous Log In

➤ **To always use Anonymous Log In**

1. Log in to the Access Manager software administration console as administrator.
2. Navigate to the default organization or sub-organization.
3. Choose Services from the View menu.
4. Click the Properties icon next to Core.
5. For the Authentication Menu, make sure Anonymous is selected and deselect all other entries.
6. Click Save.
7. Create the anonymous user. With the desired organization selected, choose Users from the Show menu.
8. Click New.
9. Select the services for the anonymous user.

   Typically, you select Desktop and NetMail.
10. Type in the Create User screen with the following information.
    - UserID - anonymous
    - First Name - (blank)
    - Last Name - anonymous
    - Full Name - anonymous
    - Password - anonymous
    - Password (confirm) - anonymous

11. Click the create button to create the user.

    When users type the URL to access the portal server in a browser, the anonymous Desktop comes up, bypassing the Access Manager software login page. This Desktop will have the login channel, where users can log in if desired.

Modifying the Anonymous Banner and Menu Bar

To change the banner for the Anonymous Desktop, you need to modify the `/etc/opt/SUNWps/desktop/anonymous/banner.template` file. To modify the menu bar, you need to modify the `/etc/opt/SUNWps/desktop/anonymous/menubar.template` file.

➤ **To Change the Banner for the Anonymous Desktop**

1. Edit the `banner.template` file.
2. Make your modifications.

   For example, you could change the following line to a background color or image of your choice:

```
<td bgcolor="#333366"><img src="[surl:/images/productName.gif]" width="274
" height="38" alt="Sun Java System Portal Server"></td>
```

   Replace `[surl:/images/productName.jpg]` with a reference to an alternate image. For example, if you use *AccessManager-base*`/public_html/images/newimage.gif`, then use `/images/newimage.gif` as your replacement text. The `[surl:]` tag references image files from the Portal Server software web application archive. Your own custom images need to be placed elsewhere, so the `[surl:]` tag is not used.

3. Place your file in the appropriate directory.

   You can place your custom image files under the web server document root or you can deploy them in a custom web application archive. See the web server documentation for information on how to deploy a web application archive.

   You could also make a new `banner.template` file to replace the default one.

4. Modify the `menubar.template` file. You could also make a new `menubar.template` file to replace the default one.

Adding the Login Channel to the Anonymous Desktop of a Newly Created Organization

The default organization in the sample portal is configured with the login channel on the Anonymous Desktop. This enables new users who do not already have a membership user account to sign up for a membership user account. The login channel is also the only way a user can log in when anonymous is the sole authentication module selected.

As you add new organizations, you might want to set up the login channel on the Anonymous Desktop of the new organization.

➤ **To Add the Login Channel to the Anonymous Desktop of a Newly Created Organization**

1. Use the Access Manager software administration console to create the new organization (this example uses `company22.com` as the initial organization and `sesta.com` as the new one), register the appropriate services (Core, Membership, LDAP, Desktop, NetMail, User, and so on), create the service templates, and assign policies to execute Desktop and NetMail.

   See the *Portal Server Administration Guide* for details.

TIP Make sure that the Desktop policy contains the rule to execute the Desktop, and that in the Core service you add Membership to the Authentication Menu.

2. In the Access Manager software administration console, choose Organizations from the View menu in the Identity Management tab.
3. Navigate to the newly created organization.
4. Create a user account for the authless session.
   a. Choose Users from the View menu then click New.
   b. Select Desktop and NetMail for services then click Next.

      The Create User page opens in the data pane.

c. Type values for the required fields. This example uses authlessanonymous as the user ID and authlessanonymous as the password.

d. When done click Create.

The authlessanonymous user ID appears in the list of users.

5. Add the authlessanonymous user ID to the list of authorized users for the global Desktop service.

a. Choose Service Configuration tab.

b. Click the Properties arrow icon next to Portal Desktop.

The Desktop attributes page opens in the data pane.

c. Type the following for the Authorized Authentication-less User IDs attribute:

```
uid=authlessanonymous,ou=People,dc=sesta,dc=com|authlessanonymous
```

d. Click Add.

e. Click Save.

6. Load the display profile for the organization by using the `dpadmin` command.

This example uses the `dp-org.xml` file as the display profile for the new organization, `sesta.com`.

```
/opt/SUNWps/bin/dpadmin add -u "uid=amAdmin,ou=People,dc=sesta,dc=com" -w
password -d "dc=sesta,dc=com" /opt/SUNWps/samples/desktop/dp-org.xml
```

7. Copy the sample anonymous display profile, `dp-anon.xml`, to a new file.

For example,

```
<Reference value="Login"/>
...
<String name="Login" value="1"/>
...
<String value="Login"/>
...
<Boolean name="Login" value="false"/>
...
<Channel name="Login" provider="LoginProvider">
```

You do not want to modify the sample `dp-anon.xml` file, as you may want to have it as a backup in case need it for reloading that for your default organization.

8. Edit the `dp-anon-sesta.xml` display profile file to change every instance of the Login channel to LoginSesta.

The lines of the `dp-anon-sesta.xml` display profile to be changed look like this:

```
<Reference value="Login"/>
...
<String name="Login" value="1"/>
...
<String value="Login"/>
...
<Boolean name="Login" value="false"/>
...
<Channel name="Login" provider="LoginProvider">
```

NOTE Do not change LoginProvider to LoginSestaProvider. The provider name must stay the same.

9. Load the anonymous display profile for the authless user ID by using the `dpadmin` command.

```
/opt/SUNWps/bin/dpadmin add -u "uid=amAdmin,ou=People,dc=sesta,dc=com" -w
password -d "uid=authlessanonymous,ou=People,dc=sesta,dc=com"
dp-anon-sesta.xml
```

10. Create the channel templates for the new login channel.

a. Change directories to the `/etc/opt/SUNWps/desktop/`*desktoptype* directory.

```
cd /etc/opt/SUNWps/desktop/desktoptype
```

b. Copy the `Login` directory contents to a new directory, `LoginSesta`.

```
cp -r Login LoginSesta
```

c. Change directories to the `LoginSesta` directory.

```
cd /etc/opt/SUNWps/desktop/desktoptype/LoginSesta
```

d. Change the Form action value from `/amserver/login` to `/amserver/login?org=sesta.com` in all the display template files (`display.html`, `display_AuthLDAP.html`, and `display_AuthUnix.html`).

e. Change the "Sign me up" URL from `<A HREF="/amserver/login?module...>` to `<A HREF="amserver/login?org=sesta.com&module...>` in all the display template files.

11. Set the Desktop type for the authless user.

a. In the Access Manager software administration console, select the newly created organization.

b. Choose Users from the View menu.

c. Click the Properties arrow icon next to the authlessanonymous user ID.

d. Select Edit at the end of the Desktop line in the data pane.

e. In the popup window, type anonymous in the Desktop Type field and select Customize in the drop-down menu next to the text field.

f. Click Save.

g. Access the authless anonymous Desktop for the new organization by typing the following URL:

```
http://psserver:port/portal/dt?desktop.suid=uid=authlessanonymous,ou=People,dc=sesta,dc=com
```

Modifying the Default Desktop (Container) for authlessanonymous User

To change the default channel name for authlessanonymous user from JSPTabContainer to another container, for example, JSPTableContainer, perform the following:

➤ **To Change the Default Channel Name for Authlessanonymous User**

1. Log in to the administration console and select Users View for your organization.
2. Select authlessanonymous and Portal Desktop from the View pull-down menu for authlessanonymous users.
3. Select the Edit link.
4. Change the Default Channel Name and select Customize from the pull-down menu.
5. Select Save.
6. Validate the change to the Desktop.

Chapter 8

Customizing Authentication

Sun Java System Portal Server supports a number of authentication schemes, including LDAP, anonymous, membership, UNIX, and more. See the *Portal Server Administration Guide* for information on configuring authentication, as well as users and roles.

This chapter contains instructions for:

- Using UNIX Authentication with LoginProvider
- Configuring LDAP Authentication for UserInfoProvider

Using UNIX Authentication with LoginProvider

➤ **To Use UNIX Authentication with LoginProvider**

1. Change directories to the default/Login directory.

   For example:

   ```
   cd /etc/opt/SUNWps/desktop/default/Login
   ```

2. Copy the `display_UnixAuth.html` file to `display.html`.

   For example,

   ```
   cp display_AuthUnix.html display.html
   ```

3. Register and enable UNIX authentication service for the organization.

   See the *Portal Server Administration Guide* for details.

4. Add Unix to the Non Interactive Modules in the Core service.

NOTE To use LDAP authentication, the authentication module is already enabled for the default organization. You only need to copy `display_AuthLDAP.html` to `display.html`.

Configuring LDAP Authentication for UserInfoProvider

Out of the box, the UserInfo channel allows the user to edit and maintain their Membership password (change their own password). To change the user's authentication module to only LDAP, the administrator has to customize the UserInfoProvider to acknowledge LDAP authenticated users.

➤ **To Enable End User Password Maintenance for LDAP Authentication**

1. Create an LDAP `passwordHandler` template. The template name format is `passwordHandler-`*authType*`.template`.

   You can copy an existing template in the Userinfo template directory. For example,

   ```
   cd /etc/opt/SUNWps/desktop/default/UserInfo/html
   ```

   ```
   cp passwordHandler-Membership.template passwordHandler-LDAP.template
   ```

2. Optionally, modify the descriptive text within `passwordHandler-`*authType*`.template`.

   For example, in the `passwordHandler-LDAP.template` file, change the `Membership` to `LDAP`.

3. Add the authentication module name to the channel's `authTypes` display profile Collection.

   Use the `dpadmin` utility to add the entry to the UserInfoProvider `<Provider>` element. For example:

   a. Type ***PortalServer-base*/bin/dpadmin list -u *uid* -w *password* -g > provider.xml**.

      Make a backup copy of provider.xml. For example, type **`cp provider.xml provider-original.xml`**.

   b. Add the entry `LDAP` to the `authTypes` collection for the UserInfoProvider in the `provider.xml` file as shown (in bold) below:

```
<Collection name="authTypes" advanced="true">
    <String value="Membership"/>
    <String value="LDAP"/>
</Collection>
```

Here, based on the user's SSOToken authentication type, the appropriate authType will be used.

c. Import the modified display profile document. For example, type

***PortalServer-base*/bin/dpadmin modify -u *uid* -w *password* -g provider.xml**

4. Restart the web container.
5. Access the portal desktop as an LDAP authenticated user and edit the user info channel.

   Verify that the password field is displayed.
6. Modify the user's password and select finished
7. Logout and login to the Desktop with the new credentials.

Chapter 9

Modifying the Desktop Layout

This chapter describes how to modify the channel arrangement in the Desktop. It contains the following sections:

- Deriving More Desktop Layouts
- Changing Content Layout to Support Categorizing the Available and Selected Lists
- Adding New Layouts
- Changing the Desktop Column Layout

Deriving More Desktop Layouts

The Desktop Layout page provides a way for users to set the arrangement of the channels by moving them up or down, and right or left. The Desktop Layout page also provides users with the option to set column layout, where they can arrange columns by channel width.

Channel widths are defined as thin, wide, full_top, and full_bottom. A thin channel takes less Desktop area than a wide channel. A full_top channel spans the entire Desktop width above all the other channels. A full_bottom channel spans the entire Desktop width below all other channels. The available layouts, which use different combinations of channel widths, are:

- thin-wide (two columns)
- wide-thin (two columns)
- thin-wide-thin (three columns)
- full_top (one column spans the width of the Desktop at the top of the page)

- full_bottom (one column spans the width of the Desktop at the bottom of the page)

You can derive more Desktop layouts from the existing layouts by modifying display profile properties and the JSPs for the table container, when one of the contained channel's width is specified as either full_top or full_bottom.

For example, you could come up with the following:

- full_top-thin-wide / full_top-wide-thin / full_top-thin-wide-thin
- thin-wide-full_bottom
- fulltop-thin-wide-full_bottom (This layout is a combination of the first two with full_top at the top and full_bottom at the bottom.)

To do so involves modifying the appropriate display profile. That is, to derive more desktop layouts, use the appropriate display profile for the desired layout. After making a change to the display profile, load the display profile into LDAP by using the `dpadmin` command.

To use a full_top-thin-wide / full_top-wide-thin / full_top-thin-wide-thin layout, modify a channel's width in the display profile as follows:

```
<Channel name="Search" provider="SearchProvider">
    <Properties>
        <String name="title" value="Search"/>
        <String name="description" value="This is a search provider example"
/>
        <String name="searchServer" value=""/>
         <String name=”width” value=”full_top”/>
    </Properties>
</Channel>
```

To use a thin-wide-full_bottom layout, modify a channel's width in the display profile as follows:

```
<Channel name="Search" provider="SearchProvider">
    <Properties>
        <String name="title" value="Search"/>
        <String name="description" value="This is a search provider example"
/>
        <String name="searchServer" value=""/>
```

```
        <String  name="width" value="full_bottom"/>
    </Properties>
</Channel>
```

To use a full_top-thin-wide-full_bottom layout, modify the width of one of the channels as full_bottom and one of the other channels as full_top in the display profile.

Changing Content Layout to Support Categorizing the Available and Selected Lists

For sites with a great number of channels, and a need to categorize and sub-categorize these into a hierarchical kind of structure—to make channel selection easier to navigate, and channels easier to find—customization options include:

- Customizing Existing JSPs
- Writing a New Content Channel

Customizing Existing JSPs

You can customize `contentedit.jsp`, which generates the Content page, and `contentdoedit.jsp`, which processes the page. The Content page uses categories to group channels into different sections. If you want this page to look different, for example, to use a pull-down menu instead of showing all the channels for one category, to save on page space, you can customize the two JSPs mentioned. (The main reason for a JSP-based content and layout channels is to support ease of customization.)

Writing a New Content Channel

You can write a new content channel by extending JSPSingleContainerProvider to support your site's needs. You would then only need to modify the Content link in `table.jsp` under JSPTableContainerProvider to point to this new content channel. You can also create your own JSP to do whatever specific implementation you need, and change the link from `table.jsp` to `new.jsp`.

Adding New Layouts

The following example is intended to show some of the customization possibilities for the Desktop. Details are not provided. See the *Portal Server Developer's Guide* for more information on how to add new layouts.

In this scenario, the Desktop has three rows. The first row contains one full-width channel; the second row contains 2 channels, a thin plus a thick channel; and the third row contains 3 thin channels of equal width.

To create such a Desktop requires a custom container channel, created from the JSPs for the table container (JSPTableContainer).

To enable the Layout link to work with this container, you need a new layout channel with a customized JSP for editing and processing the Edit page. You build this by starting from the `layoutedit.jsp` and `layoutdoedit.jsp` files.

You also need to create a new custom table container provider by extending JSPTableContainerProvider.

Changing the Desktop Column Layout

➤ **To Change the Desktop Column Layout from the Command Line**

1. Modify the `width` property to change the layout of the channel column. Specify one of the following:
   - thin - The channel appears in one of the thin columns.
   - thick - The channel appears in one of the wide columns.
   - full_top - The channel appears at the top spanning the entire horizontal space.
   - full_bottom - The channel appears at the bottom spanning the entire horizontal space.

   For example, the following specifies that a channel appears at the top spanning the entire horizontal space:

   ```
   <String name="width" value="full_top"/>
   ```

2. Load the display profile into LDAP by using the `dpadmin` command

   See "Editing the Display Profile."

➤ To Change the Desktop Column Layout from the Administration Console

1. Log in to the Access Manager software administration console as administrator.
2. Navigate to User Management by choosing View User Management.
3. Select the appropriate organization or suborganization.
4. Choose Services from the Show menu.
5. Select the properties arrow next to Desktop in the navigation pane.

   The Desktop attributes appear in the data pane.
6. Click Channel and Container Management.

   The Channels page appears, with the container path set at the root. The defined channels appear in a list.
7. Select the Edit link beside the channel to be modified.

   The Properties page appears.
8. Modify the width property.

   The possible values are thin, thick, full_top, and full_bottom.
9. Click Save.

➤ To Modify Column Widths Directly (Using Scriptlets)

To modify the column widths directly, you can add the following code to the `JSPTableContainerProvider/toptable.jsp` file.

```
<%@ page import="com.sun.portal.search.providers.util.Layout"%>
<%
    JSPTableContainerProvider tcp =
(JSPTableContainerProvider)pageContext.getAttribute("JSPProvider");
    int layout = tcp.getLayout();
    int centerWidth = -1;
    int rightWidth = -1;
    int leftWidth = -1;
    switch (layout) {
        case Layout.LAYOUT_THIN_THICK:
            leftWidth = 40;
            rightWidth = 60;
            break;
        case Layout.LAYOUT_THICK_THIN:
            rightWidth = 40;
            leftWidth = 60;
            break;
```

```
        case Layout.LAYOUT_THIN_THICK_THIN:
            rightWidth = 30;
            centerWidth= 40;
            leftWidth = 30;
            break;
        default:
            rightWidth = 40;
            leftWidth = 60;
            break;
    }
%>
and then replace the widths
<!-- BEGIN LEFT CHANNELS -->
<TD WIDTH="<%=leftWidth%>" VALIGN=TOP>
<!-- BEGIN CENTER CHANNELS -->
<TD WIDTH="<%=centerWidth%"  VALIGN=TOP>
<!-- BEGIN RIGHT CHANNELS -->
<TD WIDTH="<%=rightWidth%>"  VALIGN=TOP>
```

Chapter 10

Branding the Desktop

This chapter describes how to brand the Desktop with your site's logo and name. It contains the following sections:

- Changing the HTML Title (Title That Appears in the Browser)
- Changing the Logo (Image) in the Banner Header
- Changing the Header and Footer of the Theme, Content, and Layout Pages

Changing the HTML Title (Title That Appears in the Browser)

The title is in the `productName` property in the display profile definition for all the providers and channels. Edit this property to change the HTML title.

Changing the Logo (Image) in the Banner Header

The logo image is defined in the themes in the display profile. The related theme properties are:

`brandImage`	The brand image on the left of the header.
`brandImage2`	The brand image in the center of the header; if there's no need to have the second image, then use `spacer.gif`.
`brandImageBgColor`	The background color for the left image file.
`brandImage2BgColor`	The background color for the center image file.

`brandImageWidth`	The width of the left image file.
`previewImage`	The image that is displayed in the Theme / preset Themes page.

➤ **To Change the Logo (image) in the Banner**

1. Log in to the Sun Java System Access Manager administration console.
2. Select Services from your Organization View pull-down menu and select Portal Desktop.
3. Select Edit XML to directly edit the display profile XML fragment. Or,
   a. Select Container and Channel Management Link and Edit Properties for Display Profile.
   b. Select Edit Collection for Global Themes and the theme you wish to modify.
4. Modify the relevant theme properties.
5. Copy the new image into *PortalServer-base*`/web-src/images` directory.
6. Run *PortalServer-base*`/bin/deploy redeploy -deploy_admin_password` *password* command to deploy the new image.

   The images will be deployed to the *web_container_install_root*`/portal_web_application_install_root/images` directory.
7. Reload the Desktop to verify the change.

Changing the Header and Footer of the Theme, Content, and Layout Pages

Depending on the Desktop, the particular header and footer files for in the Theme, Content, and Layout pages are determined by the container that contains the Theme, Content, and Layout pages.

For example, when you access the Content page for JSPTabContainer, JSPContentContainer is the container that is used to include the header and footer files that the tab container is using. The `contentedit.jsp` file, located in the JSPContentContainer directory, uses logic, based on the container, to access the appropriate header and footer files.

To brand the header and footer of the Theme, Content, and Layout pages:

Use Table 10-1 to determine the appropriate header and footer JSP files to edit. This two column table lists the Desktop containers in the first column and the corresponding header and footer files in the second column.

Table 10-1 Header and Footer Files for Theme, Content, and Layout Pages

Desktop Container	Header and Footer Files for Theme, Content, and Layout Pages
FrameTabContainer	`framePreferenceHeader.jsp` and `framePreferenceMenubar.jsp`
JSPSingleContainer	`singlePreferenceHeader.jsp` and `singlePreferenceMenubar.jsp`
JSPTabContainer	`tabPreferenceHeader.jsp` and `tabPreferenceMenubar.jsp`
JSPTableContainer	`tablePreferenceHeader.jsp` and `tablePreferenceMenubar.jsp`

Containers not listed in Table 10-1 use `defaultHeader.jsp` and `defaultMenubar.jsp` files. These two files are actually the same as `singlePreferenceHeader.jsp` and `singlePreferenceMenubar.jsp` files. If you want a default look and feel for the container's header and menubar, customize these two JSPs. Currently, the sample portal does not use `defaultHeader.jsp` and `defaultMenubar.jsp` files.

➤ To Change the Header and Footer of the Theme, Content, and Layout Pages

1. Change to the appropriate directory.

   That is, change to `/etc/opt/SUNWps/desktop/sampleportal` (if sampleportal is installed) or change to the specific desktop type subdirectory associated with the target user or organization.

2. Edit the JSP files.

   For example, change the HTML title and logo in the header file, and change the product name in the footer.

3. Run the `touch` command.

   For example, type **`touch *.jsp`**.

4. Reload the Desktop to verify the change.

Chapter 11

Changing Desktop Colors

This chapter describes how to change the color for various Desktop components, such as header, footer, font color in the header and footer, and so on. This chapter contains the following:

- Changing Desktop Colors
- Changing the Default Color Scheme for an Organization

Changing Desktop Colors

Most of these colors are part of the global theme attributes. See *Portal Server Technical Reference Guide* for more information.

➤ **To Change the Desktop Colors**

1. Use Table 11-1 on page 82 to determine what you want to change and what file you need to change. This two column table lists the Desktop component to customize in the first column, the files to edit in the second column, and the corresponding theme attribute in the third column.

Table 11-1 Files to Customize to Change Desktop Colors

Desktop Component to Customize	File to Edit in /etc/opt/SUNWps/desktop/default/	Theme Attribute
Header background color	JSP-based containers: • ./JSPSingleContainerProvider/header.jsp • ./JSPTabContainer/header.jsp • ./JSPTableContainerProvider/header.jsp • ./PredefinedFrontPageFramePanelContainerProvider/header.jsp • ./PredefinedFrontPageTabPanelContainerProvider/header.jsp • ./PredefinedSamplesTabPanelContainerProvider/header.jsp Frame-based containers: • ./FrameTabContainer/banner.jsp • ./PredefinedSamplesFramePanelContainerProvider/header.jsp	brandBgColor
Footer background color	JSP-based containers: • ./JSPSingleContainerProvider/menubar.jsp • ./JSPTabContainer/menubar.jsp • ./JSPTableContainerProvider/menubar.jsp • ./PredefinedFrontPageFramePanelContainerProvider/menubar.jsp • ./PredefinedFrontPageTabPanelContainerProvider/menubar.jsp • ./PredefinedSamplesTabPanelContainerProvider/menubar.jsp Frame-based containers: • ./FrameTabContainer/menubar.jsp • ./PredefinedSamplesFramePanelContainerProvider/menubar.jsp	brandBgColor
Font color in the header and footer	The related JSPs are the `header.jsp` and `menubar.jsp` that listed in header background color and footer background color.	headerFontColor

Table 11-1 Files to Customize to Change Desktop Colors *(Continued)*

Desktop Component to Customize	File to Edit in /etc/opt/SUNWps/desktop/default/	Theme Attribute
Selected tab color	JSP-based containers: ./JSPTabContainer/selectedTab.jsp Frame-based containers: ./FrameTabContainer/selectedTab.jsp	titleBarColor
Content Page color	./JSPContentContainer/contentLayoutBar.jsp ./JSPEditContainer/contentLayoutBar.jsp ./JSPLayoutContainer/contentLayoutBar.jsp ./TabJSPEditContainer/contentLayoutBar.jsp	(none)
Layout Page color	./JSPLayoutContainer/layoutedit.jsp	(none)
Desktop body	./JSPTableContainerProvider/tabtable.jsp	tableBgColor

2. Edit the appropriate file.

   In almost all case, make modifications to the `bgcolor=value` statement to change the color. In the case of the font color in the header and footer, change the color inside of the `FONT` tag for the specific link.

3. In the directory where you make the change, run the following command:

   ```
   touch *.jsp
   ```

   (Or, if you know the parent JSP file, just run the `touch` command on that file.)

4. Reload the Desktop.

Changing the Default Color Scheme for an Organization

There are two ways in which to provide a new color scheme and layout for an organization:

- Define a new set of templates - You can define a new set of templates in `/etc/opt/SUNWps/desktop/new/` and make this directory (new) the Desktop Type attribute for that organization. See the *Portal Server Administration Guide* for more details.

- Define a new theme - In the display profile, you can define your own theme in the GlobalThemes collection. See Chapter 12, “Customizing the Global Themes” for more information.

Chapter 12

Customizing the Global Themes

This chapter contains the following:

- Customization Overview
- When you add (or customize) a global theme, all channels see the change, as themes are a global property for all channels.

Customization Overview

There are two levels of customization for the themes:

The number of themes and theme attributes are configurable by the administrators. Theme and theme attributes are display profile properties; so they can be modified through the Sun Java System Access Manager software administration console, or they can be edited in the display profile directly. The theme properties are defined as global properties in the organization level in the sample portal. So, when a new theme is created, all users in the organization will see it.

The end user can select one of the preset themes that are defined by the administrator, or customize some theme attribute values inside of the theme page in the Desktop. When the theme changes, it applies to all the containers in the Desktop, and also, the changed property will be stored in the user level display profile.

There are tag library functions defined to allow JSPs to retrieve the theme related values from the display profile. Behind the scene, the tag library functions use the Theme Java class to get the theme properties. For more information on the Theme Java class, please see the Java docs for `com.sun.portal.providers.context.Theme`.

When you add (or customize) a global theme, all channels see the change, as themes are a global property for all channels.

See the *Portal Server Technical Reference Guide* for more information on the Global themes

Adding a Theme to the Sample Portal

1. Develop the display profile XML definition for the new theme and ensure that the new collection has all thirty eight (38) properties defined in the display profile.

   The collection can be added either using the Access Manager administration console, or via the `dpadmin` command.

2. Copy new images into the *PortalServer-base*`/web-src`. That is:

   - `activeBulletImage`, `inactiveBulletImage`, `brandImage`, `brandImage2`, `previewImage`: these images must be copied in to the *PortalServer-base*`/web-src/images` directory.
   - `helpImage`, `removeImage`, `minimizeImage`, `maximizeImage`, `normalImage`, `attachImage`, `detachImage`, `editImage`: these images must be copied in to the *PortalServer-base*`/web-src/desktop/images` directory.
   - `tabNotchImage` must be copied in to the *PortalServer-base*`/web-src/desktop/tabs/images` directory.

3. Run the `deploy` command to deploy the image files. For example, type:

   PortalServer-base`/bin/deploy redeploy -deploy_admin_password` *password*

4. Verify that the new theme shows up on the Desktop's Theme page.

Customizing the Current Themes

Change the theme values in the display profile. You can modify the theme properties from the administration console, or by using the `dpadmin` command to load the XML fragment.

Changing in the administration console is easier, since you want to pin point some specific properties, and change the values.

In the sample portal, the *PortalServer-base*`/samples/desktop/dp-org.xml` file defines eight themes. For more information, see the display profile fragment for GlobalThemes in the `dp-org.xml` file.

➤ To Change the Text

The font families and font sizes are combined and defined in the following theme attributes:

- `headerText`
- `titleText`
- `contentLayoutText`

The value of these attributes is actually a class defined in the desktop *Web-Container-Instance*`/desktop/style.css` file. In the Desktop `style.css` file, there are predefined font family + font size as follows:

```
.sansSerif12Font { font-family: sans-serif; font-size: 12pt }
.sansSerif11Font { font-family: sans-serif; font-size: 11pt }
.sansSerif10Font { font-family: sans-serif; font-size: 10pt }
.sansSerif9Font { font-family: sans-serif; font-size: 9pt }
.sansSerif8Font { font-family: sans-serif; font-size: 8pt }
.sansSerif6Font { font-family: sans-serif; font-size: 6pt }
.monospace12Font { font-family: monospace; font-size: 12pt }
.monospace11Font { font-family: monospace; font-size: 11pt }
.monospace10Font { font-family: monospace; font-size: 10pt }
.monospace9Font { font-family: monospace; font-size: 9pt }
.monospace8Font { font-family: monospace; font-size: 8pt }
.monospace6Font { font-family: monospace; font-size: 6pt }
.serif12Font { font-family: serif; font-size: 12pt }
.serif11Font { font-family: serif; font-size: 11pt }
.serif10Font { font-family: serif; font-size: 10pt }
.serif9Font { font-family: serif; font-size: 9pt }
.serif8Font { font-family: serif; font-size: 8pt }
.serif6Font { font-family: serif; font-size: 6pt }
.verdana12Font { font-family: verdana; font-size: 12pt }
.verdana11Font { font-family: verdana; font-size: 11pt }
.verdana10Font { font-family: verdana; font-size: 10pt }
.verdana9Font { font-family: verdana; font-size: 9pt }
.verdana8Font { font-family: verdana; font-size: 8pt }
.verdana6Font { font-family: verdana; font-size: 6pt }
```

To change the font for the header text, title text, and content and layout text, please use one of the predefined class name, or, add new class definition in the `style.css` file, and then use it.

➤ To Change the Sample Anonymous Desktop Theme

Anonymous users cannot choose their own theme as this is determined by the display profile. However, an administrator who has permission to edit the anonymous user's display profile can change the theme for the anonymous Desktop.

1. Create a temporary file containing the theme definition.

   For example,

   ```
   vi newtheme.xml
   ```

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
    <Collection name="GlobalThemes">
        <Collection name="NeonTheme">
            <String name="description" value="Neon"/>
            <String name="bgColor" value="white"/>
            <String name="titleBarColor" value="#CCFF33"/>
            <String name="fontColor" value="black"/>
            <String name="borderColor" value="#CCFF33"/>
            <String name="borderWidth" value="1"/>
            <String name="fontFace" value="Sans-serif"/>
            <String name="fontSize" value="3"/>
            <String name="activeBulletImage" value="neon_bullet_on.gif"/>
            <String name="inactiveBulletImage"
value="neon_bullet_off.gif"/>
            <String name="brandImage" value="Neon_000000.gif"/>
            <String name="brandImage2" value="spacer.gif"/>
            <String name="brandImageBgColor" value="#000000"/>
            <String name="brandImage2BgColor" value="#000000"/>
            <String name="brandBgColor" value="#000000"/>
            <String name="headerBgColor" value="#000000"/>
            <String name="brandImageWidth" value="243"/>
            <String name="headerFontColor" value="#FFFFFF"/>
            <String name="headerText" value="sansSerif11Font"/>
            <String name="tabNotchImage" value="black_tabend.gif"/>
            <String name="tabColor" value="#FF0000"/>
            <String name="tabFontColor" value="#000000"/>
            <String name="contentLayoutLinkColor" value="#000000"/>
            <String name="contentLayoutText" value="sansSerif10Font"/>
            <String name="linkSeparatorColor" value="#FF0000"/>
            <String name="tableBgColor" value="#FFFFFF"/>
            <String name="titleFontColor" value="#000000"/>
            <String name="titleText" value="sansSerif10Font"/>
            <String name="channelHighlightColor" value="#ffffff"/>
            <String name="channelLinkColor" value="#3a2eb5"/>
            <String name="previewImage" value="Neon_preview.gif"/>
            <String name="helpImage" value="b_help.gif"/>
            <String name="removeImage" value="b_remove.gif"/>
            <String name="minimizeImage" value="b_minimize.gif"/>
            <String name="maximizeImage" value="b_maximize.gif"/>
            <String name="normalizeImage" value="b_normal.gif"/>
```

```
            <String name="attachImage" value="b_attach.gif"/>
            <String name="detachImage" value="b_new_window.gif"/>
            <String name="editImage" value="b_edit.gif"/>
        </Collection>
    </Collection>
    <String name="UserTheme" value="NeonTheme" propagate="false"/>
```

2. Run the dpadmin command with the add sub-command to load the display profile.

   For example, to load the XML fragment to the anonymous user's node,

   PortalServer-base/bin/dpadmin add -u "uid=amAdmin,ou=People,o=sesta.com,o=isp" -w *password* -d "uid=anonymous,ou=people,o=sesta.com,o=isp" newtheme.xml

3. Log in to the administration console and change the global property. That is:

   a. Select Users from the View pull-down menu.

   b. Select Authlessanonymous

   c. Select View->Portal Desktop on the right panel and select the Edit link.

      A popup window displayed.

   d. Select Channel and Container Management.

   e. Select Display Profile: Edit Properties and change User Theme to "NeonTheme".

   f. Click on Save to save the settings.

4. Log out of the administration console.

Chapter 13

Customizing the Service Providers

This chapter provides common customization tasks for modifying the Search provider and Discussion provider.

This chapter contains the following sections:

- Overview of Customizing the Service Providers
- Tips for Customizing the Service Providers
- Customizing the Search Provider
- Customizing the Discussion Channels

Overview of Customizing the Service Providers

The Sun Java System Portal Server software includes a search service and discussion service provider.

Overview of Customizing the Search Provider

The Search provider (SearchProvider) furnishes a basic reference user interface that contains both search and browse functionality. Search functionality includes basic search mode, and advanced search for more complex searches. You can perform specific field searches in advanced search mode. For example, while in advanced mode, you can search within the title, URL, last modified date, author, and so on.

SearchProvider provides a link for category browsing. In addition, you can create a taxonomy for the Search Engine along with category filter rules. You can browse through the taxonomy tree and view documents within a category through the Search provider interface.

Search Provider Design

The Search provider uses JSPProvider to access the Portal Server back end services. The Search provider users JavaServer Pages™ (JSP™) helper tag libraries to avoid using Java™ scriptlets. The `searchServer` is a global service list type attribute that is configured and updated at installation time. The Search provider is responsible for directing the search request to the appropriate back end Search Engine server.

See the *Portal Server Technical Reference Guide* for the display profile properties you can set for the provider.

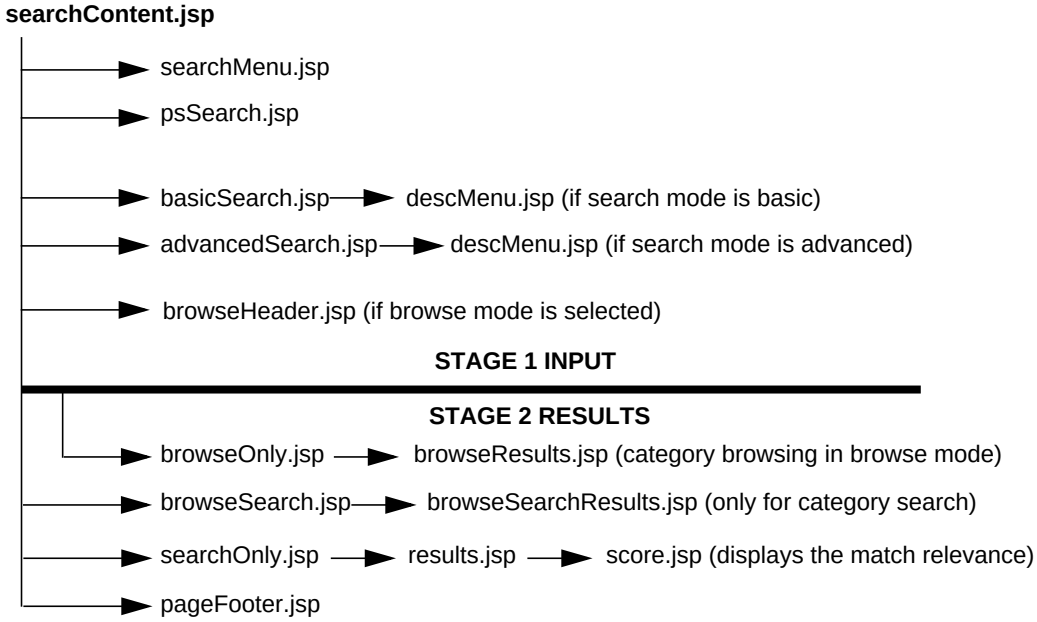
Search JavaServer Pages and Tag Libraries

The Search provider consists of two stages—input form and results—and the JSPs used by the Search provider fall into one of those two stages.

Figure 13-1 on page 93 explains the JSP layout for `searchContent.jsp`. In the input stage (Stage 1), `searchContent.jsp` makes use of `searchMenu.jsp` and `psSearch.jsp` to set up the initial interface. The `basicSearch.jsp` file is used for a basic search and `advancedSearch.jsp` file for an advanced search. The description menu— that is, the Full, Brief, and Title menus—is displayed for both basic and advanced searches by `descMenu.jsp` file. The `browseHeader.jsp` file defines the browse interface.

In the results stage (Stage 2), one of three JSPs is used: `browseOnly.jsp` file sets and executes the parameters for category browsing using the Search tag library, and includes the `browseResults.jsp` page; `browseSearch.jsp` file sets and executes the parameters for searching and browsing within categories using the Search tag library and includes `browseSearchResults.jsp`; and `searchOnly.jsp` file sets and executes the parameters for search using search the Search tag library and includes `results.jsp` and `score.jsp` for the match relevance. The `pageFooter.jsp` file displays the list of pages, Next, and Previous links.

Figure 13-1 JSP Layout for `searchContent.jsp`



See the *Portal Server Technical Reference Guide* for more information on the Search JSP files.

The Search JSPs use the following tag libraries, which ship with the Portal Server software:

- Jakarta tag library, for any generic tags
- Desktop tag library, for the Portal Server software related information
- Search tag library to cover all Search Engine server access functionality and to provide a tag-based wrapper for the existing public search and SOIF API

See the *Portal Server Technical Reference Guide* for more information.

Overview of Customizing the Discussion Provider

The DiscussionProvider is JSPProvider based and hence customizable. It uses the Desktop themes. It retrieves data from the back end Search service using search taglibs and API. The discussions and comments are stored as separate Resource Descriptors (RDs) in the discussion database.

The DiscussionProvider includes features such as discussion threads, starting discussions based on documents or new topics, searching discussions, and rating discussions. By default, the Discussions channel is available on the sample portal for anonymous users. However, an anonymous user cannot subscribe to a discussion or edit the Discussion channel.

The DiscussionProvider supports a full view (via the Discussions channel) and a lite view (via the DiscussionLite channel.) It has the following main functions:

- Start a new discussion from the discussion channel.
- Start a new discussion based on web documents from the search channel.
- Add a comment to an existing discussion or post a reply to an existing discussion.
- Rate all discussions and comments. Note that the displayed ratings are based on an algorithm such that the rating for any comment goes up gradually. For example, a comment has to be rated important three times before it is marked as important.
- Search all discussions and search within a discussion. These functions are routed to the search provider. The `displaySearch` property can be disabled if the search feature is not required in the discussion channel. Users can also search by rating in Advance Search.
- Authenticated users can choose to subscribe to a particular discussion by selecting the subscribe link. The request is handled by the SubscriptionProvider. The `displaySubscription` property can be disabled if the feature is not required. By default, the value is true.

A Discussion Lite view retrieves main posts sorted by last-modified date and has pagination so users can access older discussions. View discussion displays each discussion subtree. The main item is displayed in detail and the subtree is displayed below the main item. View discussion includes:

- Several filters on the page. A document display can be based on filters such as document rating (irrelevant, routine, interesting, important, and must read).

- Display preference can be set to threaded or flat display.

Expansion threshold helps to control displayed items in the subtree. The users can choose to expand only highly rated documents, or expand all or collapse all. Default value is collapse all. Expand all will expand all the filtered comments. It will also show a description of the discussion, provide a menu for rating the discussion, and allow the user to post a reply.

Discussion Service Channels

The DiscussionLite channel and the Discussions channel are based on the DiscussionProvider.

DiscussionLite Channel

The DiscussionLite channel displays the top twenty discussion titles (which can be reconfigured) and the date. The discussions are sorted by creation date (last modified) and the newest discussion is displayed first. The DiscussionLite channel view has links to view each discussion, view all discussions, and start a new discussion. All these links target the Discussions channel which gets displayed in the JSPDynamicSingleContainer.

Properties for this channel can be configured from the administration console. By default, there are no user editable properties for this channel.

Discussions Channel

The Discussions channel includes a full view that:

- Shows detailed descriptions for the top eight discussions sorted in descending order. This can be reconfigured via the channel edit page.
- Includes pagination so that users can see all the discussions.
- Supports search. The search returns discussion and comment results.

Discussion JavaServer Pages and Tag Libraries

Similar to the search channel JSPs, the discussion channel JSPs have a query portion, a display portion, and use Desktop themes.

For more information on the channel specific JSP files, see the *Portal Server Technical Reference Guide*.

Tips for Customizing the Service Providers

This section provides some basic tips for customizing the search and discussion providers.

Debugging the Service Providers

The Portal Server software provides files to help debug the Search and Discussion providers.

The following directory contains various search log files:

`/var/opt/SUNWps/https-psservername/portal`

The following search log file records the search query sent to the Search Engine by the Search server:

`/var/opt/SUNWps/https-psservername/portal/logs/rdm.log`

See the *Portal Server Administration Guide* for more information about the Search log files.

Location of JavaServer Pages

JavaServer Pages for the Search channel are in the `/etc/opt/SUNWps/desktop/default/Search` directory.

JavaServer Pages for the DiscussionLite channel are in the `/etc/opt/SUNWps/desktop/default/DiscussionLite` directory.

JavaServer Pages for the Discussions channel are in the `/etc/opt/SUNWps/desktop/default/Discussions` directory.

Modifying JavaServer Pages

When you modify statically included JavaServer Pages, be sure to run the touch command, otherwise no changes are reflected. You need to either run the touch command on the top-level JSP file or on all JSP files. For example,

```
touch searchContent.jsp
```

or

```
touch *.jsp
```

See also "JavaServer Page Caching Information" on page 115 and "Recompiling JSPs" on page 114.

Accessing Channels Directly

You can access the search channel directly at the following URL:

```
http://server:port/portal/dt?provider=JSPDynamicSingleContainer&JSPDynamicSingleContainer.selectedChannel=Search&last=false&action=content
```

Modify all the links to use these extra parameters in the URL. For example, edit `searchMenu.jsp` file as follows:

```
<nobr>  <a class=noUnderline href="<%=dpurl%>?mode=basic">Basic Search</a>  </nobr>
```

Replace the bold portion with:

```
http://server:port/portal/dt?provider=JSPDynamicSingleContainer&JSPDynamicSingleContainer.selectedChannel=Search&last=false&action=content
```

You can access the Discussion channel directly at the following URL:

```
http://server:port/portal/dt?provider=JSPDynamicSingleContainer&JSPDynamicSingleContainer.selectedChannel=Discussions&last=false&action=content
```

Customizing the Search Provider

This section describes how to perform some common customizations on the Search provider.

➤ **To Modify the Default Search Server**

When you install the Portal Server software, the Search provider is linked back to the Search server by default. That is to say, the Search provider can connect to any Search server, but by default the value for the `searchServer` property is set to the following:

```
http://server:port/portal/search
```

The `searchServer` property is normally initialized to the default portal server instance during the sample portal installation but this property is not set when you create a new organization or a new custom search channel. See *Portal Server Administration Guide* for information on configuring the `searchServer` property for the Search provider.

1. Log in to the Sun Java System Access Manager software administration console as administrator.
2. Choose Organizations from the Show menu in User Management.

   All created organizations display in the navigation pane.
3. Navigate to the organization or sub-organization that you want to configure Search for.
4. Choose Services from the Show menu.
5. Click the properties arrow next to Desktop in the Navigation pane.
6. Click Channel and Container Management.

   The Channels page appears.
7. In the Channels section, click the Edit link for Search.

   The Edit Channel page appears for the Search channel.
8. Edit the `searchServer` property with the Search server's name.
9. Click Save.

➤ To Add last-modified to the Search Result Display

1. Modify `searchOnly.jsp` file by adding last-modified to the list of `viewAttributes`.

   For example:

```
<search:setViewAttributes
viewAttributes="hl-url,hl-title,hl-description,score,content-length,hl-clas
sification,last-modified"/>
```

2. Modify `results.jsp` file to display the last-modified date for document results using the SOIF `getValue` tag.

   For example:

```
<% if (formbean.getDescription().equals("full")) { %>
    <FONT color=<%=tFontColor%> face=<%=tFontFace%>><search:getValue
soifAttribute="description" escape="false"/></FONT><BR>
    <FONT color=#707070 face=<%=tFontFace%>><search:getURL
escape="true"/><BR>
    <search:getValue soifAttribute="content-length" id="sz"/>
    <search:getValue soifAttribute="last-modified"/><BR>
<% } %>
```

3. Run the `touch` command.

   For example, type **`touch *.jsp`**.

4. Reload the Desktop to verify the change.

➤ **To Remove content-length from Search Results**

1. (Optional) Modify `searchOnly.jsp` file by removing content-length from the list of viewAttributes.

   The line to modify is the following:

```
<search:setViewAttributes viewAttributes=
"hl-url,hl-title,hl-description,score,content-length,classification
hl-classification"/>
```

   Remove content-length from this line.

2. Modify `results.jsp` file by removing the line that displays the content-length.

   The line to modify is the following:

```
<% } else if (formbean.getDescription().equals("full")) { %>
    <FONT size=-1 color=<%=tFontColor%>
face=<%=tFontFace%>><search:getValue soifAttribute="hl-description"
escape="false"/></FONT><BR>
    <FONT size=-1 color=#707070 face=<%=tFontFace%>><search:getValue
soifAttribute="hl-url" escape="false"/><BR>
    <search:getValue soifAttribute="content-length" id="sz"/>
```

3. Remove `<search:getValue soifAttribute="content-length" id="sz"/>` from this file.
4. Run the `touch` command.

   For example, type **`touch *.jsp`**.
5. Reload the Desktop to verify the change.

➤ **To Display the Total Number of Documents in the Search Result Status Message**

In this procedures, Steps 1 and 2 are independent of each other. If desired, run the `touch` command after Step 1 to see the results.

1. Modify `results.jsp` by changing the search status line to add the `<search:getTotalDocuments/>` tag.

   For example:

```
<NOBR><B>Document matches <search:getFirstHit/> - <search:getToHit/> (of
<search:getHitCount/>)</B> out of <search:getTotalDocuments/></NOBR><BR>
```

   This results in the following display:

   `Document matches 1 - 6 (of 6) out of 37`
2. Change `browseResults.jsp` by adding the `<search:getTotalDocuments/>` tag to the search status.

   For example:

```
<FONT color="<%=tFontColor%>" face="<%=tFontFace%>"
size="-1"><b>Subcategories <search:getFirstHit/> - <search:getToHit/> (of
<search:getHitCount/>)</B> out of <search:getTotalDocuments/></FONT><br>
```

   This results in the following display:

   `Category matches 1 - 2 (of 2) out of 86`
3. Run the `touch` command.

   For example, type **`touch *.jsp`**.

4. Reload the Desktop to verify the change.

➤ **To Remove author from the Advanced Search Interface**

1. Comment our or remove the author related HTML from the `advancedSearch.jsp` file.

   For example:

```
<!--     -->
<!-- To disclude the "author" row, remark out the following section -->
<!--     -->
<TR>
    <td valign=middle align=right height=40><FONT color=<%=tFontColor%>
face=<%=tFontFace%>><nobr> <LABEL FOR="advAuthor">Author</LABEL>
    <SELECT NAME="authorOp">
    <OPTION VALUE=<%=SearchContext.CONTAIN%>
<%=formbean.authorOpSelection(SearchContext.CONTAIN)%>>does</OPTION>
    <OPTION VALUE=<%=SearchContext.NOTCONTAIN%>
<%=formbean.authorOpSelection(SearchContext.NOTCONTAIN)%>>does not</OPTION>
    </SELECT>contain </FONT></nobr></TD>
    <td valign=middle align=left height=40><INPUT TYPE="text"
NAME="authorVal" id="advAuthor" VALUE=
"<%=SearchContext.htmlEncode(formbean.getAuthorVal())%>"></TD>
</TR>
```

2. Comment out author- related lines in `advQuery.jsp` file.

```
// h = new HashMap();
    // h.put(SearchContext.OPERAND, "author");
    // h.put(SearchContext.OPERATION, formbean.getAuthorOp());
    // h.put(SearchContext.VALUE, formbean.getAuthorVal());
    // l.add(h);

    h = new HashMap();
    h.put(SearchContext.OPERAND, "title");
```

3. Run the `touch` command.

   For example, type **`touch *.jsp`**.

4. Reload the Desktop to verify the change.

➤ To Add a New Field to Advanced Search

1. Uncomment the keywords section in `advancedSearch.jsp` file.

```
<!--     -->
<!-- To Include the "Keywords" row, unremark the following section -->
<!--
<TR>
<td valign=middle align=right height=40><FONT color=<%=tFontColor%>
face=<%=tFontFace%>><nobr> <LABEL FOR="advKeywords">Keywords</LABEL>
<SELECT NAME="keywordsOp">
<OPTION VALUE=<%=SearchContext.CONTAIN%>
<%=formbean.keywordsOpSelection(SearchContext.CONTAIN)%>>does</OPTION>
<OPTION VALUE=<%=SearchContext.NOTCONTAIN%>
<%=formbean.keywordsOpSelection(SearchContext.NOTCONTAIN)%>>does
not</OPTION>
</SELECT>contain </FONT></nobr></TD>
<td valign=middle align=left height=40><INPUT TYPE="text" NAME="keywordsVal"
id="advKeywords" VALUE=
"<%=SearchContext.htmlEncode(formbean.getKeywordsVal())%>"></TD>
</TR>
-->
```

   Remove the `<!--` and `-->` comment marks from this section.

2. Add the keywords to `advQuery.jsp` file.

```
h = new HashMap();
h.put(SearchContext.OPERAND, "keywords");
h.put(SearchContext.OPERATION, formbean.getKeywordsOp());
h.put(SearchContext.VALUE, formbean.getKeywordsVal());
l.add(h);
```

3. Run the `touch` command.

   For example, type **`touch *.jsp`**.

4. Reload the Desktop to verify the change.

Customizing the Discussion Channels

Customizing DiscussionLite Channel

➤ **To Customize the DiscussionLite Channel Link Display Window**

1. Change directory to `/etc/opt/SUNWps/desktop/default/DiscussionLite` directory and edit the following JSP files.

   - `display.jsp` - In this file, comment out or delete the following line:

```
<a
href="<%=desktopPathInfo%>?last=false&Discussions_dmode=vl&did=<%=Encoder.u
rlEncode(url)%>"><B><search:getValue soifAttribute="title" escape="true"
truncate="28"/></B></a>
```

   - `discussionLiteContent.jsp` - In this file, comment out or delete the following lines. :

```
Map pathInfo = new HashMap();
pathInfo.put("action", "content");
pathInfo.put("provider", "JSPDynamicSingleContainer");
pathInfo.put("JSPDynamicSingleContainer.selectedChannel", "Discussions");
pathInfo.put("last", "false");
pageContext.setAttribute("pathInfo", pathInfo);
<dtpc:getDesktopURL id="desktopPathInfo" pathinfo="$pathInfo"/>
```

2. Replace all occurrences of `desktopPathInfo` with `dt`.

   The `desktopPathInfo` ensures that links are always displayed in the Discussions channel in a JSPDynamicSingleContainer. Remove this if you want links to be displayed in the Discussions Channels on the same tab. For example, replace the following line:

```
<td align=center><font size="-1"><a target="ps_main" href=
"<%=desktopPathInfo%>?Discussions_dmode=cmt">New Discussion</a></font>
```

with the following:

```
<td align=center><font size="-1"><a target="ps_main" href=
"dt?Discussions_dmode=cmt">New Discussion</a></font>
```

This procedure will work only if Discussions and DiscussionLite are displayed on the same tab as in the sample portal. DiscussionLite links will be displayed in the Discussions channel on the right side on the collaboration tab.

➤ **To Display DiscussionLite on the Front tab**

1. Modify MyFrontPageTabContainer and add DiscussionLite to the available and selected lists. For example:

   In the administration console:

   a. Log in and select Services (from the View pull-down menu) for your organization.

   b. Select Portal Desktop.

   c. Select Channel and Container Management link and MyFrontPageTabContainer from the list of containers.

      The page to edit the MyFrontPageTabContainer is displayed.

   d. Select DiscussionLite from the Existing Channels list and select the Add arrow pointing towards Available and Visible list.

   e. Select Save and log out.

   From the command line, edit `dp-org.xml` file to include DiscussionLite in the available and selected list. For example, add the text shown below in bold:

```
<Container name="MyFrontPageTabPanelContainer"
provider="PredefinedFrontPageTabPanelContainerProvider">
    <Properties> ...</Properties>
    <Available>
         <Reference value="DiscussionLite”/>
    </Available>
    <Selected>
         <Reference value="DiscussionLite”/>
    </Selected>
    ...
</Container>
```

2. Modify the DiscussionLite channel display profile `isEditable` property and set it to `true`.
3. Log in and verify.

Customizing Discussions Channel

➤ **To Display Additional Fields in the List View of Discussions**

1. Change directories to `/etc/opt/SUNWps/desktop/default/Discussions`.
2. Modify `query.jsp` file and add `xxx` field to `viewAttributes`.

   For example, add content-length as follows:

```
<search:setViewAttributes viewAttributes=
"url,title,description,rd-rating,author,last-modified,rd-last-changed,rd-re
ference-id,
rd-num-rating,rd-sum-rating,rd-peak-rating,rd-reference-url,content-lengt
h"/>
```

3. Add the new fields in `fullDiscussionDisplay.jsp` file wherever appropriate.

   For example:

```
<search:getValue soifAttribute="content-length"/>
```

➤ **To Modify the Sort Order in List All Discussions Page**

By default, discussions are sorted by the last-modified date/time. That is, discussions are displayed in a descending order with the latest or most recent discussion shown first.

To modify the sort order in list All Discussions page, modify the `viewOrder` property in `fullDiscussion.jsp` file. For example, you can reset the value below to `author` or `rd-last-changed`:

```
<jx:set var="viewOrder" value="-last-modified"/>
```

➤ To Modify viewHits in View Discussion Page

```
<jx:set var="hitNumber" value="500"/>
```

By default, the `viewHits` property is set to 500. Make this -1 if you want all the comments to be displayed or reduce this number to improve performance. For example, if the value is 200, only the first 200 comments will be displayed in this case (includes comments and sub-comments.) The `hitNumber` for view discussion page can be reset in the `viewDiscussion.jsp` file.

➤ To Inherit Classification and readACL

If you have classified only the parent discussion manually or modified the access control for the parent discussion, you may want to inherit those values in discussion replies as follows:

1. Change directories to `/etc/opt/SUNWps/desktop/default/Discussions`.
2. Edit `feedbackProcess.jsp` file and modify the values of `inheritClassification` and `inheritReadACL`.

   By default, these are set to `false`. Reset them to `true` if you want comments to inherit the parent's classification field and `readACL` field. Note that comments are automatically protected in this case.
3. Save the file.

➤ To Control Access to Discussions

This can be accomplished by one of the following two ways:

1. Modify the `dbname` property in the display profile for Discussions and DiscussionLite channel for each role to point to a different database.

   In this case users in one role cannot view discussions created by users in a different role.
2. Or modify the `ReadACL` of the parent discussion as it gets submitted in the database and set `inheritReadACL` to `true`.

   That is, you must search for the discussion first or search for `rd-reference-id <contains> ROOT` and modify the `readACL` field for discussions.

Chapter 14

Customizing the Desktop End-User Online Help

This chapter provides information on customizing the Sun Java System Portal Server Desktop online help.

This chapter contains the following sections:

- Overview of the Desktop End-User Online Help
- Location of the Desktop End-User Online Help HTML Files
- Modifying the Desktop End-User Online Help HTML files

Overview of the Desktop End-User Online Help

The Desktop end-user online help is a collection of HTML files that is referenced in the display profile. Each provider, including the various container providers, has a display profile entry for a corresponding help file. In the display profile, each provider definition has the default value for the help file. If a channel that uses the provider has a different help file, the `helpURL` property can be defined in the channel definition also, overriding the provider's value.

The display profile entries for the online help are defined in the provider properties. The entries define the string name, `helpURL`, and its value.

The `helpURL` property is a conditional property. Multiple values can be associated with the `helpURL` property and the display profile API returns the proper value depending on the client type and locale. If your portal server is configured to serve multiple clients (such as HTML, WML) in multiple locales (such as english, french), the `helpURL` property will allow you to set up multiple help files based on the type of client and type of locale you are serving.

For example, Code Example 14-1 specifies different help files for different locales and different client type.

Code Example 14-1 HelpURL Property in Display Profile Definition

```
<ConditionalProperties condition="client" value="WML">
    <ConditionalProperties condition="locale" value="en">
        <String name="helpURL" value="en/wml/help.wml"/>
    </ConditionalProperties>
    <ConditionalProperties condition="locale" value="fr">
        <String name="helpURL" value="fr/wml/help.wml"/>
    </ConditionalProperties>
</ConditionalProperties>
<ConditionalProperties condition="client" value="HTML">
    <ConditionalProperties condition="locale" value="en">
        <String name="helpURL" value="en/html/help.html"/>
    </ConditionalProperties>
    <ConditionalProperties condition="locale" value="fr">
        <String name="helpURL" value="fr/html/help.html"/>
    </ConditionalProperties>
</ConditionalProperties>
```

The value of the `helpURL` property can be either a relative path or an absolute path. The location in Code Example 14-1 is relative to the Desktop static content root.

The static content root is the install directory of static content. By default the static content root is:

PortalServer-base`/web-apps/https-`*psserver*`/portal`

NOTE References to the locations and contents of files in the web-apps directory are for information only. They do not represent the definition of an interface that you can depend on for any future release.

The relative path will be generated as:

static_content_root/doc_root/locale/helpURL_value

The doc root is defined in the display profile also. For example, if the doc root is `docs`, and the user's locale is `en_US`, with the `helpURL` value, the final value for the help location will be:

PortalServer-base`/web-apps/https-`*psserver*`/portal/docs/en_US/desktop/userinfo.htm`

The following is an example of an absolute URL that defines a help file location. Use a similar format for using an absolute URL to define the location of an online help file.

```
<String name="helpURL" value="http://sesta.com/docs/desktop/userinfo.htm"/>
```

Location of the Desktop End-User Online Help HTML Files

The source location of the Desktop end-user online help files for the sample Desktop is:

PortalServer-base/web-src/docs/locale/desktop

The following is a list of help files included with the sample portal. In the table, the first column lists the names of the help files. The second column contains a brief description of the help file.

addressbook.html	Describes the address book channel
bkmark.htm	Describes the bookmark channel.
calendar.html	Describes the calendar channel.
content.htm	Describes the Content link on the Desktop.
discussions.htm	Describes the discussions channel.
fdesktop.htm	Provides an overview of using the Desktop (for frames-based Desktop).
glossary.htm	Provides a glossary for the online help.
help.htm	Describes the Help link on the Desktop.
home.htm	Describes the Home link on the Desktop.
imchan.htm	Describes the instant messaging channel.
ix.htm	Provides an index for the online help.
jspchann.htm	Describes the JavaServer Pages™ channel.
layout.htm	Describes the Layout link on the Desktop.
login.htm	Describes the membership login channel.

logout.htm	Describes the Logout link on the Desktop.
mailhelp.html	Describes the mail channel.
mlchck.htm	Describes the mail check channel.
myapps.htm	Describes the My Applications channel.
notes.htm	Describes the notes channel.
options.htm	Describes the Options link on the Desktop.
ParSample.html	Provides an example of how the Portal PAR command-line utility is used to import and export a PAR file.
rsschann.htm	Describes the RSS channel.
search.htm	Describes the search channel.
sections.htm	Section help file (describes frames-based Desktop sections).
subscriptions.htm	Describes the subscriptions channel.
tabs.htm	Provides an overview of using a tabbed Desktop interface.
theme.htm	Describes the Theme link on the Desktop.
topics.htm	Provides a table of contents for the online help.
urlscrpr.htm	Describes the sample URL scraper channel.
usechann.htm	Provides an overview of using the Desktop channels.
usedesk.htm	Provides an overview of using the Desktop.
userinfo.htm	Describes the user information channel.
webchann.htm	Describes the simple web service channel.
xmlchann.htm	Describes the XML channel.

Modifying the Desktop End-User Online Help HTML files

You can customize the end-user online help by editing the existing HTML online help files or by creating new HTML files.

Editing An Existing Help File

You can edit an existing help file to customize the content to meet specific requirements of your organization. For example, you can remove or change the `SunONE.jpg` image or meta text that is currently displayed on the sample help files, or replace the help file completely with a file of the same name.

After modifying the file, run the *PortalServer-base*`/bin/deploy redeploy -deploy_admin_password` *password* command to deploy the file into the web-app location.

This method of modifying the online help files is useful if you use the sample providers that are shipped with the Portal Server software.

For example, if you use the UserInfoProvider that ships with the Sun Java System Portal Server product, the display profile for that provider already defines the `helpURL` value as `desktop/userinfo.htm`. By editing the help file `userinfo.htm`, no changes to the display profile are necessary.

See"Location of the Desktop End-User Online Help HTML Files" on page 109 for the online help file names and descriptions, and where they are installed on your system.

Creating a New Help File

You can create a new help file by creating a new HTML file. This method of customizing the online help is useful if, for example, you add a new provider to the Desktop. When you create a new help file, you must modify the display profile to contain the new `helpURL` value. You can either manually edit the display profile or use the Access Manager software administration console.

➤ **To Create a New Online Help File and to Define the `helpURL` Value Manually**

1. Create an HTML file for the provider you want to document.
2. Place your file in the appropriate directory.

   You can place your custom help files under the web server document root, in the directory specified as root by the display profile:

   PortalServer-base`/web-src/docs/locale/desktop`

   Or, you can deploy them in a custom web application archive. See the web server documentation for information on how to deploy a web application archive.

3. Run the *PortalServer-base*/bin/deploy redeploy -deploy_admin_password *password* command to deploy the file.
4. Define the helpURL value for that file in the display profile.

   To define the helpURL value for a new online help file, use the format described in the section "Overview of the Desktop End-User Online Help" on page 107.
5. Use the dpadmin command to load the display profile into LDAP.
6. Verify that the new help file is displayed correctly.

To create a new online help file and to define the helpURL value in the Access Manager software administration console, do the following:

1. Create an HTML file for the provider you want to document.
2. Place your file in the appropriate directory.

   You can place your custom help files under the web server document root, in the directory specified as root by the display profile:

   PortalServer-base/web-src/docs/locale/desktop

   Or, you can deploy them in a custom web application archive. See the web server documentation for information on how to deploy a web application archive.
3. Run the *PortalServer-base*/bin/deploy redeploy -deploy_admin_password *password* command to deploy the file.
4. Login to the Access Manager software administration console as the administrator.
5. Choose Services from the View pull-down menu for your organization.
6. Choose the arrow next to Portal Desktop.
7. Choose the Channel and Container Management link.
8. In the channels area, choose Edit Properties for the channel you want to edit.
9. Enter the helpURL Value.
10. Choose Save.
11. Verify that the new help file is displayed correctly.

Chapter 15

Miscellaneous JSP and Template Information

This chapter contains the following sections:

- Performing JSP Redirects
- JSP vs. Theme Color
- Recompiling JSPs
- JavaServer Page Caching Information
- Debugging JSPs
- Dynamic Template Reloading

This chapter contains miscellaneous information on using JSPs and templates when customizing the Desktop.

Performing JSP Redirects

To perform redirects from the `doedit.jsp` page back to the `edit.jsp` page, use the following URL:

```
response.sendRedirect(p.getProviderContext().getDesktopURL(request).toStrin
g() + "?action=edit&provider=ipsdtJSPEditChannel" + "&targetprovider=" +
p.getName() + "&reedit=true");
```

To return back to the Edit page, use the following, which does not hardcode the editchannel name.

```
String editChannel = request.getParameter("editChannelName");
response.sendRedirect(p.getProviderContext().getDesktopURL(request).toStrin
g() + "?action=edit&provider=<%=editChannel%>&targetprovider=" +
p.getName() + "&redit=true");
```

JSP vs. Theme Color

The colors of a theme can be changed in two places: in the JSP or template file that uses the theme tag, and in the display profile using the GlobalThemes attribute.

If you change the color value in the JSP or template file, the Desktop uses this value and not what is in the GlobalThemes attribute.

For example, if you manually change the following background color in a JSP file

```
BGCOLOR="<dttheme:getAttribute name="borderColor"/>"
```

to

```
BGCOLOR="#FFFFFF"
```

then the Desktop shows white for border color, and the theme color is not used. There is no way the theme or the Desktop can detect this.

Recompiling JSPs

Every time you make a modification to a JSP file, you need to recompile. You do this by running the `touch` command on the modified container's top-level JSP file.

For example, type **`touch tab.jsp`**.

NOTE A typical desktop will include content from several JSPs. However, if you make a modification to a non-toplevel JSP that has been included in a top level JSP, the included JSP will not be recomplied. The end result is your desktop changes will never be reflected.

If the top-level JSP is touched, the JSP engine recompiles all the relevant JSPs. If you cannot find the top level JSP, run the `touch` command on all JSPs in the directory, for example,

```
touch *.jsp
```

This modifies all JSPs, including the top-level JSP.

See JavaServer Page Caching Information for information on how to find the top-level (parent) JSP.

JavaServer Page Caching Information

When the system compiles JavaServer Pages™ (JSP™), the result is only one Java™ class per parent JSP. There is no compiled class for the static included JSPs. Thus, when you change an included JSP, you need to run the touch command on the parent JSP to recompile the parent JSP with the changed JSP. The Portal Server software JSP engine checks the last modification time on the compiled class and the JSP file to see if the JSP needs to be recompiled. In this way, the change takes effect immediately.

To find a JSP's parent JSP, search in the JSP for the string `<%@ include file="filename.jsp" %>`. Some JSPs are dynamically included by using the `<jsp:include page="header.jsp" flush="true"/>` syntax instead of `<%@ include file="header.jsp" %>`. This syntax compiles `header.jsp` and generates a separate Java class.

The path to cached JSPs is constructed in such a way so that the compiled JSPs do not conflict with each other in multi-server instances, when multiple Desktop types contain the same JSPs and for multiple clientTypes and locales. So when JSPs are dynamically included, the touch command does not need to be run for the parent JSP.

Debugging JSPs

The JSP classes are created at:

/var/opt/SUNWps/https-psservername/portal/tmp/_jsps

You can find compilation and runtime errors in the Desktop debug log at:

/var/opt/SUNWam/debug/desktop.debug

Also, all JSPProvider based Desktop channels have a property called showExceptions. This property, by default, is set to false; setting it to true causes the JSP exception to show up as the content of the channel.

Dynamic Template Reloading

If you make changes to the Desktop templates, note that these templates are dynamically reloaded. The reload interval is by default set to thirty seconds. You can change the reload interval in the desktopconfig.properties file at /etc/opt/SUNWps/desktop directory.

➤ **To Change the Template Reload Interval**

1. Log in to the Portal Server host and change directories to /etc/opt/SUNWps/desktop.
2. Open the desktopconfig.properties file and reset the templateScanInterval property value.
3. Save and close the file.

Index

A

Anonymous Desktop
accessing 58–60
adding the Login channel 62–65
changing the theme 88
configuring to always use anonymous log in 60
customization ??–66
disabling authlessanonymous log in 57
disabling log in 56
enabling authlessanonymous log in 56
enabling log in 56
modifying the banner and meny bar 61
user session 55
Attributes
DesktopType 16, 20
Authentication
configuring anonymous authentication 55
configuring LDAP authentication for UserInfoProvider 68
configuring to always use anonymous log in 60
disabling authentication-less log In 57
enabling authentication-less login 56
LDAP 68
screen customization 14
using UNIX with LoginProvider 67
Authlessanonymous
accessing the Desktop through the Access Manager host name 58
disabling log in 57
enabling log in 56
modifying the default Desktop for the user 66
Available list
categorizing 73

B

Banner
changing the header image 77
modifying anonymous Desktop banner 61
Border,channel 47
Buttons,channel 44

C

Caching
container 42
JavaServer Pages (JSPs) 115
provider 42
Channel
accessing directly 97
changing the border width 47
changing the layout for a table container 46
controlling and configuring container caching 42
customizing refresh times 41

customizing the border 48
customizing window preference 42
dynamic content 14
removing a button 43–??
removing a title bar 46
widths 71
Colors,Desktop
changing 81
changing the default scheme 83
Column
modifying the layout 74
modifying the width 75
Container
creating 28
deploying JSPs and template files 28
dynamic content 14
how-to stretch a tab 36
modifying the default Desktop for authlessanonymous user 66
removing a button from all channels 43
Content page,Desktop
changing the header and footer 78
header and footer files 79
Content,dynamic 14
Customization
Desktop
areas of 14
search channel 14

D

Debug file,Desktop 30
Desktop
adding new layouts 74
base templates 14
changing column layout 74
changing content layout 73
content page customization 78
debugging 30
deriving more layouts 71
how-to access 30
layout page customization 78
modifications to layout ??–76
DesktopType attribute
changing 20
purpose 16
DiscussionProvider/channel
channel customization 103
customization overview 94
discussion threads 94
DiscussionLite channel
customization 103
customize link display window 103
displaying on the front tab 104
introduction 95
discussions 94
Discussions channel
accessing directly 97
controlling access to discussions 106
customization 105
displaying additional fields in the list view 105
inheriting classification and readACL 106
introduction 95
modifying the Sort Order 105
modifying `viewHits` 106
displaying DiscussionLite on the front tab 104
full view 94
JavaServer Pages
introduction 95
lite view 94
rating discussions 94
searching discussions 94
tag libraries
introduction 95
Display Profile
definition
introduction 15
document
structure 15
Document Type Definition (*see* DTD,Display Profile)
editing 26
introduction 15
loading 29
logo image properties 77
Document Type Definition (*see* DTD,Display Profile)
documentation
overview 8
Documents,Display Profile

structure 15
dp-org.xml 21
dp-providers.xml 21
DTD,Display Profile 15

E

Errors
JSP compilation and runtime 28
Exception,JSP 28

F

File
Desktop color files 82
Desktop debug 30
Desktop online Help 109
header and footer files for Theme, Content, and Layout pages 79
Font
changing families and sizes 87

G

GlobalThemes
changing font sizes and font families 87
changing the anonymous Desktop theme 88
changing the color 114
changing the page header and footer 78
customization ??–89
customization overview 85
header and footer files 79
logo image 77

H

Help,Desktop online
customization overview 107
HTML file
customization 110–112
listing 109
How-to
access the anonymous Desktop through the Access Manager host name 58
access the anonymous Desktop through the Portal Server host name 59
add a Channel to a User-defined tab 39
add a role-based tab 38
add a tab to JSPTabContainer 31–35
add new layouts 74
add the Login channel to the anonymous Desktop 62–65
change the anonymous Desktop theme 88
change the background color of tabs 37
change the channel border width 47
change the channel layout for a table container 46
change the colors of a theme 114
change the default color scheme for an organization 83
change the Desktop colors 81
change the Desktop type 20
change the font families and font sizes 87
change the header and footer in the Theme, Content, and Layout pages 78
change the Tab Image 37
change the tab image for JSP-based tab containers 37
change the text 87
closing the Instant Messaging invite window 52
configure anonymous authentication 55, 55–58
configure to always use the anonymous log in 60
create a tab within a tab 36
customize container caching 41
customize JSPs for restructuring content layout 73
customize the channel border 48
customize the display of Instant Messaging contacts 52
customize the GlobalThemes 86
customize window preference 42
debug the Desktop 30
derive more Desktop layouts 71
disable the user from editing instant messaging server information 49

load the Display Profile 29
make the tab the start tab 38
modify the anonymous banner and menu bar 61
modify the default Desktop for the authlessanonymous user 66
perform JSP redirects 113
remove a button from a single channel 44
remove a button from all channels in a container 43
remove the title bar from a channel 46
stretch a tab across an entire container 36
write a new content channel 73
HTML files, Desktop online help 109

I

Image
changing the banner header logo 77
changing the tab image for JSP-based tab containers 37
`index.html` file,Portal Server 59
Instant Messaging
closing the invite window 52
customizations ??–53
customizing the display of contacts 52
disabling the user from editing the server information 49

J

Jakarta tag library 93
JavaServer Pages (see JSPs,Desktop)
JSPFrameCustomTableContainerProvider
customization 47, 48
JSPs,Desktop
caching 115
changing the theme color 114
compilation errors 28
customizing for restructuring content layout 73
debugging 115
default location 16
deploying 27
exception 28
for sample portal 16
installed location 96
introduction 16
introduction to SearchProvider 92
modifying 96
performing redirects 113
recompile 114
runtime errors 28
JSPSingleContainerProvider 73
JSPTabContainer
adding a role-based tab 38
changing the tab image 37
how-to add a tab 31–35
JSPTabCustomTableContainerProvider
customization 47, 48
JSPTableContainerProvider 47, 48
modifying the Content link 73

L

Layout
adding new layouts 74
changing column layout 74
changing content layout 73
changing the page header and footer 78
Layout page,Desktop
changing the header and footer 78
header and footer files 79
List,available and selected
categorizing 73
LoginProvider / channel
adding the channel to the anonymous Desktop 62, 62–65
using UNIX authentication 67
Logo image
Display Profile properties 77
Look and feel customization,Desktop 14

M

Menu bar
  modifying the anonymous Desktop menu bar 61

O

`obj.conf` file,Access Manager 58
Online help,Desktop
  customization 107–112
  file customization 110
  file location 109
  overview 107

P

Pages,Desktop 78
PredefinedNewTabPanelContainerProvider 35
Property,Display Profile
  channel button 44
Provider
  definition 15
  dynamic content 14

R

Refresh times,channel 41
`refreshTime` property,channel 41
Resource bundle,creating 30
Resources,static 14

S

Sample portal
  adding a theme 86
  Desktop
    default channel 22
    viewing 22
  editing the files 20
  restoring default settings 21
  templates 14
SearchProvider/channel
  accessing channel directly 97
  adding a new field 102
  channel customization 14
  customization
    adding last-modified 98
    displaying the total number of documents 100
    modifying the default search server 97
    overview 91
    removing author 101
    removing content-length 99
  design 92
  JavaServer Pages
    introduction 92
    layout 93
  log files 96
  tag libraries 92
Selected list
  categorizing 73
Service provider
  customization 91–106
  customization overview 91
  customization tips 96
  debugging 96
Static resources 14
Structure,Display Profile document 15

T

Table container
  changing the channel layout 46
Tabs,Desktop
  adding a channel to a user-defined tab 39
  adding a role-based tab 38
  changing the back-ground color 37
  changing the tab image 37
  making a tab the start tab 38
  strectching a tab 36
Tag libraries
  introduction 16

introduction to SearchProvider 92
Jakarta 93
Templates
base Desktop 14
deploying 27
installed location 16
introduction 16
sample portal 14
Title,HTML 77
Top container,customizing 14

U

UserInfoProvider / channel
configuring LDAP authentication 68

W

Widths
channel width 71
modifying column width 75

X

XML files,Display Profile
`dp-org.xml` 21
`dp-providers.xml` 21