

Desktop Customization Guide

Sun™ ONE Portal Server

Version 6.2

816-6756-10
November 2003

Copyright © 2003 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 © 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plus des brevets américains listés à l'adresse <http://www.sun.com/patents> et un ou les brevets supplémentaires ou les 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 pourront 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.

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 licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

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

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

Contents

About This Guide	11
Who Should Read This Book	11
What You Need to Know	11
How This Book Is Organized	12
Document Conventions Used in This Guide	13
Monospaced Font	13
Bold Monospaced Font	13
Italicized Font	14
Square or Straight Brackets	14
Command-Line Prompts	14
Variables	15
Where to Find Related Information	15
Where to Find This Guide Online	16
Chapter 1 Introduction to Customizing the Desktop	17
Types of Desktop Customizations	17
End User Customizations	18
Developer Customizations	18
Administrator Customizations	19
Using the Display Profile	20
Display Profile Document Structure	23
Modifying the Display Profile	27
Where to Go for More Information	27
Using JavaServer Pages and Template Files	27
The Desktop and JavaServer Pages	28
The Desktop and Template Files	29
Changing the Desktop Type	30

Using the Desktop Tag Library	32
What Is the Sample Portal?	32
Creating a New Desktop	34
Developing the Container	35
Editing the Display Profile	35
Developing JSP or Template Files	36
Deploying JSP or Template Files	37
Loading the Display Profile at the Appropriate LDAP Nodes	39
(Optional) Creating a Resource Bundle	39
Accessing the Desktop	39
Creating and Extending a Provider Class	40
Chapter 2 Display Profile Properties	41
Overview of Display Profile Properties	41
Display Profile Properties	41
Display Profile Property Types	43
Display Profile Global Properties	44
Chapter 3 Container Provider Display Profile Definitions and Properties	47
Display Profile Definitions for Building-Block Container Provider	47
JSPTableContainerProvider	48
JSPTabContainerProvider	49
JSPSingleContainerProvider	50
TemplateTableContainerProvider	52
TemplateTabContainerProvider	53
Display Profile Definitions for Internally Used Container Providers	54
JSPTabCustomTableContainerProvider	54
JSPFrameCustomTableContainerProvider	56
TemplateEditContainerProvider	58
TemplateTabCustomTableContainerProvider	59
PredefinedFrontPageFramePanelContainerProvider	60
PredefinedSamplesFramePanelContainerProvider	62
PredefinedToolsTemplatePanelContainerProvider	64
PredefinedFrontPageTemplatePanelContainerProvider	66
PredefinedSamplesTemplatePanelContainerProvider	67
PredefinedSamplesTabPanelContainerProvider	69
PredefinedFrontPageTabPanelContainerProvider	70
Container Provider Display Profile Properties	72
Available and Selected List	73
Common Properties for Table Container	74
Common Properties for Tab Container	77
Other Container Properties	77

Chapter 4 Leaf Provider Display Profile Definitions and Properties	79
Overview of Display Profile Provider Definitions and Properties	79
Display Profile Definitions and Properties of Leaf Building-Block Providers	80
JSPProvider	80
URLScraperProvider	82
XMLProvider	83
Display Profile Definition for Service Providers	85
SearchProvider	86
DiscussionProvider	88
SubscriptionsProvider	90
Display Profile Definitions for Content Providers	91
AddressBookProvider	92
AppProvider	94
BookmarkProvider	95
CalendarProvider	96
IMProvider	98
LoginProvider	100
LotusNotesAddressBookProvider	102
LotusNotesCalendarProvider	104
LotusNotesMailProvider	106
MailCheckProvider	108
MailProvider	110
MSExchangeAddressBookProvider	112
MSExchangeCalendarProvider	114
MSExchangeMailProvider	116
NotesProvider	118
PersonalNoteProvider	120
SimpleWebServiceProvider	120
SimpleWebServiceConfigurableProvider	122
UserInfoProvider	124
Display Profile Common Properties for Providers	127
isEditable	130
authlessState	131
encoderClassName	131
ConditionalProperties	131
Chapter 5 Display Profile Channel Definitions	133
Overview of Channel Display Profile Definitions	133
Display Profile Definition for Container Channels	133
JSPTableContainer	134
MyFrontPageFramePanelContainer	136
SamplesTabPanelContainer	137
SearchTabPanelContainer	138

MyFrontPageTabPanelContainer	139
SamplesFramePanelContainer	141
JSPTabContainer	141
JSPCreateChannelContainer	143
JSPPresetThemeContainer	144
JSPCustomThemeContainer	145
JSPPopupContainer	146
TemplateTableContainer	147
TemplateTabContainer	148
MyFrontPageTemplatePanelContainer	150
SamplesTemplatePanelContainer	151
ToolsTemplatePanelContainer	152
FrameTabContainer	152
JSPDynamicSingleContainer	154
PredefinedFrontPageTabPanelContainer	155
PredefinedSamplesTabPanelContainer	156
JSPTabCustomTableContainer	157
PredefinedFrontPageTemplatePanelContainer	158
PredefinedSamplesTemplatePanelContainer	159
PredefinedToolsTemplatePanelContainer	160
TemplateTabCustomTableContainer	160
PredefinedFrontPageFramePanelContainer	161
PredefinedSamplesFramePanelContainer	162
FrameCustomTableContainer	163
Display Profile Definition for Leaf Channels	164
Address Book Channel	165
Applications List Channel	166
Bookmark Channel	166
Calendar Channel	167
Discussion Channels	168
Instant Messaging Channel	171
JSP Channel	172
Login Channel	172
Lotus Notes Address Book Channel	173
Lotus Notes Calendar Channel	174
Lotus Notes Mail Channel	174
Mail Channel	175
Mail Check Channel	176
MS Exchange Address Book Channel	177
MS Exchange Calendar Channel	178
MS Exchange Mail Channel	178
Notes Channel	179
Search Channel	180

Sample Simple Web Service Channel	182
Sample Simple Web Service Configurable Channel	182
Subscriptions Channel	183
Sample RSS Channel	184
User Information Channel	185
Sample URL Scraper Channel	186
Sample XML Channel	187
Chapter 6 Understanding the Sample Portal	189
Overview of the Sample Portal	189
Sample Containers for the Sample Desktop	190
Guidelines for Using the Sample Portal	191
Sample Portal Installation Directories	191
Desktop Theme Link	192
JSP-based Tab Container (JSPTabContainer)	192
JSPTabContainer Architecture	194
JSPTabContainer Actions	195
JSPTabContainer Edit Pages	197
Container Display Profile	197
JSP Files Used by JSPTabContainer	198
JSP-based Table Container (JSPTableContainer)	199
JSPTableContainer Architecture	200
JSPTableContainer Actions	201
Container's Display Profile	203
JSP Files Used by JSPTableContainer	204
Frame-based Container (FrameTabContainer)	204
FrameTabContainer Architecture	205
FrameTabContainer Actions	206
FrameTabContainer Edit Pages	207
Container's Display Profile	208
JSP Files Used by FrameTabContainer	209
Template-based Containers (TemplateTabContainer and TemplateTableContainer)	210
TemplateTabContainer	210
Chapter 7 Customizing the Desktop Interface	211
Desktop Customization Overview	211
What Are the Areas for Customizing the Desktop?	212
Channel Refresh Times and Container Caching	213
Contained Containers	214
Guidelines for Using the dpadmin Command	214
Adding Containers and Channels to the Desktop	215
Customizing Container Tabs	216

To Add a Tab to JSPTabContainer	216
Creating a Tab Within a Tab	220
To Stretch a Tab Across an Entire Container	220
To Change the Tab Image for JSP-based Tab Containers	221
To Change the Color of Tabs	222
To Make a Tab the Start Tab	222
Setting a Role Tab	223
Adding a Channel to a User-defined Tab	224
Customizing Channels	224
To Customize Window Preference	224
To Remove a Button From All Channels in a Container	225
To Remove a Button From a Single Channel	227
To Disable the User from Editing Instant Messaging Server Information	228
To Automatically Close the Instant Messaging Invite Window	230
To Customize Display of Instant Messaging Contacts	231
Creating Customized Organization JSP and Template Files	232
Customizing the Anonymous Desktop	233
Configuring Anonymous Authentication	233
To Enable Anonymous Log In	234
To Disable Anonymous Log In	234
To Enable Authentication-less (authlessanonymous) Log In	235
To Disable Authentication-less (authlessanonymous) Log In	236
To Access the Anonymous Desktop Through the Identity Server Host Name (obj.conf File) ..	236
To Access the Anonymous Desktop Through the Portal Server Host Name (index.html File) ..	238
To Disable the Initial Sun ONE Identity Server Software Login Page and Always Use Anonymous Log In	238
To Modify the Anonymous Banner and Menu Bar	239
To Add the Login Channel to the Anonymous Desktop of a Newly Created Organization ...	240
Customizing Authentication	244
To Use Unix Authentication with LoginProvider	244
To Modify the Default Desktop (Container) for authlessanonymous User	245
Modifying the Desktop Layout	245
Deriving More Desktop Layouts	245
To Derive More Desktop Layouts	246
Changing Content Layout to Support Categorizing the Available and Selected Lists	247
Adding New Layouts	248
To Change the Desktop Column Layout (Command Line)	248
To Change the Desktop Column Layout (Sun ONE Identity Server Software Administration Console)	249
To Change the Channel Layout for a Table Container	250
To Remove the Title Bar from a Channel	250
To Change the Channel Border Width	251
To Customize the Channel Border	252

To Modify Column Widths Directly (Using Scriptlets)	252
Branding the Desktop	253
To Change the HTML Title (Title That Appears in the Browser)	253
To Change the Logo (Image) in the Banner Header	254
To Change the Header and Footer of the Theme, Content, and Layout Pages	255
Changing Desktop Colors	256
To Change Desktop Colors	256
To Change the Default Color Scheme for an Organization	258
Customizing the Global Themes	259
Customization Overview	259
GlobalThemes Display Profile Definition	260
Theme Properties	267
Glossary of Terms	270
Adding and Customizing Global Themes	271
Miscellaneous JSP Information	274
Performing Redirects	275
JSP vs. Theme Color	275
Recompiling JSPs	276
JavaServer Page Caching Information	276
Desktop Debug Information	277
Debugging JSPs	277
Chapter 8 Customizing the Service Providers	279
Overview of Customizing the Service Providers	279
Overview of Customizing the Search Provider	279
Overview of Customizing the Discussion Provider	281
Tips for Customizing the Service Providers	283
Debugging the Service Providers	283
Location of JavaServer Pages	284
Modifying JavaServer Pages	284
Accessing Channels Directly	284
Customizing the Search Provider	285
To Modify the Default Search Server	285
To Modify the Default Search Server:	285
To Add last-modified to the Search Result Display	286
To Remove content-length from Search Results	287
To Display the Total Number of Documents in the Search Result Status Message	288
To Remove author from the Advanced Search Interface	289
To Add a New Field to Advanced Search	290
Customizing the Discussion Channels	292
Customizing DiscussionLite Channel	292
Customizing Discussions Channel	294

Chapter 9 Customizing the Desktop End-User Online Help	297
Overview of the Desktop End-User Online Help	297
Location of the Desktop End-User Online Help HTML files	299
Modifying the Desktop End-User Online Help HTML files	301
To Edit An Existing Help File	301
To Create a New Help File	301
Appendix A Desktop Template and Tag Reference	305
Overview of Desktop Templates and Tags	305
How the Desktop Template Tags Work	306
Template Lookup	306
Kinds of Tags Used in the Desktop	308
Desktop Templates in the default Directory and Tag Reference	308
AddressBookProvider	309
AppProvider	311
BookmarkProvider	311
CalendarProvider	313
error	328
LoginProvider	331
LotusNotesAddressBookProvider	331
LotusNotesCalendarProvider	334
LotusNotesMailProvider	349
MailCheckProvider	351
MailProvider	352
MSExchangeAddressBookProvider	355
MSExchangeCalendarProvider	357
MSExchangeMailProvider	372
TemplateTabContainerProvider	374
TemplateTableContainerProvider	377
UserInfoProvider	389
default	390
Desktop Templates in the sampleportal Directory and Tag Reference	394
MyFrontPageTemplatePanelContainer	395
PredefinedFrontPageTemplatePanelContainerProvider	396
PredefinedSamplesTemplatePanelContainerProvider	408
SamplesTemplatePanelContainer	420
TemplateTabContainerProvider	420
TemplateTabCustomTableContainerProvider	421
TemplateTableContainer	422
ToolsTemplatePanelContainer	423
sampleportal	423

Appendix B JavaServer Pages Tag Library Reference	427
Tag Library Overview	427
Desktop Tag Library Hierarchy	429
Using Tags in JSPs	430
Using the Desktop Tag Library in Your Application	433
Example Tab Page with Selected Channels	435
Tag Reference	436
Attributes and Return Values	436
Tag Library Exceptions	437
Tag Overview	438
Context Setup Tags	438
Validator Tags	439
Normal Tags	439
Search Tags	452
Instant Messaging Tags	460
Appendix C JavaServer Pages Reference	463
Anonymous Desktop JSPs	463
FrameTabContainer JSPs	464
JSPDynamicSingleContainer JSPs	464
JSPTabContainer JSPs	464
JSPTableContainer JSPs	465
PredefinedFrontPageFramePanelContainerProvider JSPs	466
PredefinedFrontPageTabPanelContainerProvider JSPs	467
PredefinedSamplesFramePanelContainerProvider JSPs	468
PredefinedSamplesTabPanelContainerProvider JSPs	469
Desktop JSPs in the default Directory	469
DiscussionLite JSPs	470
Discussions JSP	471
DiscussionsProvider JSPs	473
DummyChannel JSPs	474
IMProvider JSPs	475
JSPContentContainer JSPs	475
JSPDynamicSingleContainer	476
JSPEditContainer JSPs	476
JSPFrameCustomTableContainerProvider JSPs	476
JSPLayoutContainer JSPs	477
JSPProvider JSPs	478
JSPSingleContainerProvider JSPs	478
JSPTabContainerProvider JSPs	479
JSPTabCustomTableContainerProvider JSPs	480
JSPTableContainerProvider JSPs	481
Miscellaneous JSPs	482

SampleSimpleWebService JSPs	483
SampleSimpleWebServiceConfigurable JSPs	484
Search JSPs	484
SearchProvider JSPs	487
SimpleWebServiceConfigurableProvider JSPs	489
SimpleWebServiceProvider JSPs	490
Subscriptions JSPs	490
SubscriptionsProvider JSPs	491
TabJSPEditContainer JSPs	492
JSPs in the sampleportal Directory	492
FrameTabContainer JSPs	493
JSPContentContainer JSPs	494
JSPCreateChannelContainer JSPs	494
JSPCustomThemeContainer JSPs	495
JSPDynamicSingleContainer JSPs	495
JSPEditContainer JSPs	496
JSPFrameCustomTableContainerProvider JSPs	496
JSPLayoutContainer JSPs	497
JSPPopupContainer JSPs	497
JSPPresetThemeContainer JSPs	497
JSPSingleContainer JSPs	498
JSPTabContainer JSPs	498
JSPTabCustomTableContainerProvider JSPs	499
JSPTableContainerProvider JSPs	499
Miscellaneous JSPs	500
PredefinedFrontPageFramePanelContainerProvider	502
PredefinedFrontPageTabPanelContainerProvider	503
PredefinedSamplesFramePanelContainerProvider	504
PredefinedSamplesTabPanelContainerProvider	506
SampleJSP JSPs	507
SampleSimpleWebService JSPs	508

About This Guide

This guide explains how to customize the Sun™ ONE Portal Server Release 6.2 software Desktop. Sun ONE Portal Server software provides a platform to create portals for your organization's integrated data, knowledge management, and applications. The Sun ONE Portal Server software platform offers a complete infrastructure solution for building and deploying all types of portals, including business-to-business, business-to-employee, and business-to-consumer.

This preface includes the following sections:

- [Who Should Read This Book](#)
- [What You Need to Know](#)
- [How This Book Is Organized](#)
- [Document Conventions Used in This Guide](#)
- [Where to Find Related Information](#)
- [Where to Find This Guide Online](#)

Who Should Read This Book

You should read this guide if you are responsible for customizing the Sun ONE Portal Server software Desktop at your site.

What You Need to Know

Before you customize Sun ONE Portal Server software, you must be familiar with the following concepts:

- Basic Solaris™ administrative procedures
- HTML
- JavaServer Pages™
- LDAP
- XML

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

- Sun™ ONE Identity Server software
- Sun™ ONE Directory Server software
- Sun™ ONE Web Server software
- Sun™ ONE Application Server software

How This Book Is Organized

This book contains the following chapters and appendices:

About This Guide (this chapter)

Chapter 1, “Introduction to Customizing the Desktop” This chapter describes the Sun ONE Portal Server software Desktop, the different kinds of customizations, who would make those customizations, and the actual Desktop files that you customize.

Chapter 2, “Display Profile Properties” This chapter supplies information on the display profile definitions and properties that ship with Sun ONE Portal Server software.

Chapter 3, “Container Provider Display Profile Definitions and Properties” This chapter supplies information on the container channel default display profile definitions that ship with Sun ONE Portal Server software.

Chapter 4, “Leaf Provider Display Profile Definitions and Properties” This chapter describes the leaf provider default display profile definitions and provider specific properties.

Chapter 5, “Display Profile Channel Definitions” This chapter describes the default display profile definitions and properties of the channels that ship with Sun ONE Portal Server software.

Chapter 6, “Understanding the Sample Portal” This chapter describes the sample portal that you can choose to install on your system during the Sun ONE Portal Server software installation.

Chapter 7, “Customizing the Desktop Interface” This chapter describes common customization tasks for modifying the look and feel of the Desktop.

Chapter 8, “Customizing the Service Providers” This chapter describes common customization tasks for modifying the Search provider.

Chapter 9, “Customizing the Desktop End-User Online Help” This chapter supplies information on customizing the Sun ONE Portal Server software Desktop online help.

Appendix A, “Desktop Template and Tag Reference” This appendix describes the Desktop templates files and serves as a reference for the tags used by the template files.

Appendix B, “JavaServer Pages Tag Library Reference” This appendix describes the Desktop custom tag library and serves as a reference for the tags available within the library.

Appendix C, “JavaServer Pages Reference” This appendix describes the various JavaServer Pages used by Sun ONE Portal Server.

Document Conventions Used in This Guide

Monospaced Font

Monospaced font is used for any text that appears on the computer screen or text that you should type. It is also used for file names, distinguished names, functions, and examples.

Bold Monospaced Font

Bold monospaced font is used to represent text within a code example that you should type. For example, you might see something like this:

```
./pssetup  
*****
```

Sun(TM) One Portal Server (6.1 release)

Installation log at /var/sadm/install/logs/ipsinstall.13343/install.log

This product will run without a license. However, you must either purchase a Binary Code License from, or accept the terms of a Binary Software Evaluation license with, Sun Microsystems, to legally use this product.

Do you accept? yes/[no]

In this example, `./pssetup` is what you would type from the command line and the rest is what would appear as a result.

Italicized Font

An italicized font is used to represent text that you enter using information that is unique to your installation (for example, variables). It is used for server paths and names and account IDs.

Square or Straight Brackets

Square (or straight) brackets [] are used to enclose optional parameters. For example, in the Sun ONE Portal Server software documentation, you will see the usage for the `dpsadmin` command described as follows:

```
dpsadmin list|modify|add|remove [command-specific options]
```

The presence of [command-specific] indicates that there are optional parameters that may be added to the `dpsadmin` command.

Command-Line Prompts

Command-line prompts (for example, `%` for a C-Shell, or `$` for a Korn or Bourne shell) are not displayed in the examples. Depending on which operating system environment you are using, you will see a variety of different command-line prompts. However, you should enter the command as it appears in the document unless specifically noted otherwise.

Variables

Table 0-1 is a two column table that describes the common variables used in this document. The first column lists the variables, and the second column provides a description of how the variables are used.

<i>portal-server-install-root</i>	The Sun ONE Portal Server installation directory. For example, /opt.
<i>web-server-install-root</i>	For example: <ul style="list-style-type: none">• Sun ONE Web Server /opt/SUNWwbsvr• Sun ONE Application Server /opt/SUNWappserver7• BEA WebLogic Server 6.1 /opt/BEA/wlserver6.1• WebSphere Application Server /opt/WebSphere/AppServer
<i>directory-server-install-root</i>	The Sun ONE Directory Server installation directory. For example, /usr/sunone/servers.
<i>identity-server-install-root</i>	The Sun ONE Identity Server installation directory. For example, /opt/IS6.1.
<i>UserID</i>	User identification. For example, root or nobody.

Where to Find Related Information

In addition to this guide, Sun ONE Portal Server software comes with supplementary information for administrators as well as documentation for developers. Use the following URL to see all the Sun ONE Portal Server software documentation:

<http://docs.sun.com/prod/slportalsrv>

Listed below are the additional documents that are available:

- *Sun ONE Portal Server 6.2 Administrator's Guide*
- *Sun ONE Portal Server, Secure Remote Access 6.2 Administrator's Guide*
- *Sun ONE Portal Server 6.2 Developer's Guide*

- *Sun ONE Portal Server 6.2 Deployment Guide*
- *Sun ONE Portal Server 6.2 Installation Guide*
- *Sun ONE Portal Server 6.2 Migration Guide*
- *Sun ONE Portal Server 6.2 Release Notes*

Where to Find This Guide Online

You can find the *Sun ONE Portal Server 6.2 Desktop Customization Guide* online in PDF and HTML formats. This book can be found at the following URL:

<http://docs.sun.com/prod/slportalsrv>

Introduction to Customizing the Desktop

This chapter provides an introduction to customizing the Sun™ ONE 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](#)
- [Using the Display Profile](#)
- [Using JavaServer Pages and Template Files](#)
- [What Is the Sample Portal?](#)
- [Creating a New Desktop](#)
- [Creating and Extending a Provider Class](#)

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 Sun ONE 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™ ONE Identity Server software administration console. The URL to use is:

`http://hostname:port/amconsole`

Developer Customizations

Sun ONE 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 Sun ONE Portal Server software. See the *Sun ONE Portal Server 6.2 Developer's 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 Sun ONE Identity Server software APIs. See the *Sun ONE Identity Server 6.1 Programmer's Guide* for more information.

Administrator Customizations

Sun ONE 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.

Using the Display Profile

Much of your work in customizing Sun ONE 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 *portal-server-install-root/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. 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.

Sun ONE Portal Server software distinguishes between building-block providers, which you can extend using the Sun ONE Portal Server software APIs, and content providers, which you cannot extend. See the *Sun ONE Portal Server 6.2 Developer's Guide* for more information on extending the providers.

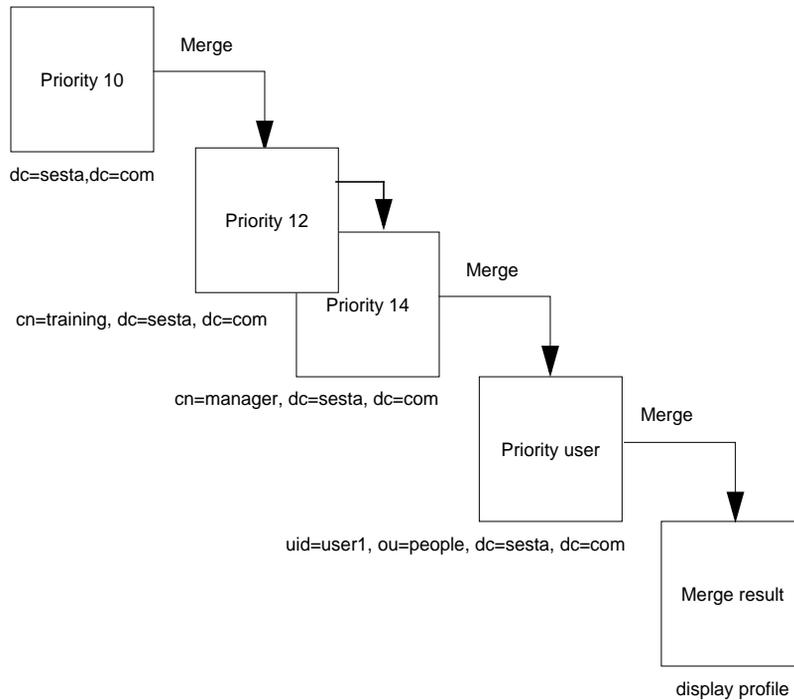
Using the display profile mechanism, you can create a new portal Desktop or modify the sample portal provided with the product. See [“What Is the Sample Portal?” on page 32](#) for more information.

Display profile documents are stored in their entirety as a single attribute in the Sun ONE Identity Server software services layer. Potentially, Sun ONE Portal Server software could store a display profile document for a user's organization or sub-organization, each role the user belongs to, and the user. That is, for the different LDAP nodes (base DN, org DN, role DNs, and uid DN), you can store a display profile document. There is also a global display profile document.

A user's display profile document set is made up from the non-empty documents stored at the user's organization, various sub-organizations, any roles, and the user LDAP nodes. This display profile document set is "merged" at runtime to form a single configuration for the user's Desktop.

For example, [Figure 1-1](#) shows a sample DIT with each level having its own display profile document. At the top of the tree is the global display profile. Next is the base DN, `dc=sesta,dc=com`. It has a two role DNs, training and manager. The tree ends at the uid DN, `user1`. Each node in the DIT has its own display profile document. The resultant display profile document is produced through a merge of all the display profile documents, based on their priorities. This merge result is presented to the user at login. Merge semantics control how display profile documents are combined. These semantics include replace, remove, and fuse. Display profile objects can also be locked. During the merge, a higher priority document can lock a display profile object to prevent a lower priority document from altering it. See the *Sun ONE Portal Server 6.2 Administrator's Guide* for more information.

Figure 1-1 Example of Merge Process



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. These are:

Provider object The provider is a programmatic entity responsible for fetching and displaying content in a channel. The XML tag for defining a provider object is `<Provider name="providerName" class="classpath">`.

The display profile document for a provider provides default property values for all properties that channels based on that provider will use. It provides the class name to use to create the provider object (see [“Provider Object” on page 26](#) for more information) at runtime.

Channel object A channel is a unit of content, usually (but not necessarily) arranged in rows and columns. You can also have channels of channels, that is, container channels. The XML tag for a channel is `<Channel name="channelName" provider="providerName">` and the XML tag for a container channel is `<Container name="containerName" provider="providerName">`.

A display profile channel definition defines the configuration for a Desktop channel. Specifically, it defines the provider Java object that is used to create a running channel instance (indirectly, by naming the display profile provider definition that names the Java class file), and defines the properties used by the channel instance. Only property values that differ from the defaults defined in the associated display profile provider definition should be defined in the display profile channel definition's properties.

Property object A property value that can be specified for a channel. Individual properties are grouped within the `<Properties>` `</Properties>` tags inside a channel definition.

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.

Because you can have multiple display profile documents defined at different LDAP nodes, at runtime, the system merges these multiple display profile documents to deliver a particular Desktop to the user. This process of merging display profile documents affects the final display profile object values.

For a complete discussion of the display profile, how the merging works, and a description of the display profile DTD syntax, see the *Sun ONE Portal Server 6.2 Administrator's Guide*.

Display Profile Document Structure

The display profile format is intended to define the Desktop's display configuration by defining provider and channel objects and their properties. Thus, a display profile is made up of some number of display profile objects. The display profile objects map directly to the XML tag that defines them. For example, the `<Channel name=>` XML tag defines a channel object.

In general, the document structure of a display profile document resembles the following:

```
<DisplayProfile>
  <Properties>...global properties...</Properties>
  <Channels>...channel definitions...</Channels>
  <Providers>...provider definitions...</Providers>
</DisplayProfile>
```

The Channels, Providers, and Properties display profile objects are used to group other display profile objects. These grouping objects are loosely referred to as “bags.” These bags add more structure to the DP XML documents.

The following sections describe the display profile objects in more detail. For a complete discussion of the display profile, see the *Sun ONE Portal Server 6.2 Administrator's Guide*.

Channel Object

A channel object represents a single display element. The objects contained by a channel object can be thought of as properties for the channel. The channel definition includes a symbolic reference to the provider. You only need to include channel-specific properties when the provider defaults are not appropriate.

Code Example 1-1 Example Channel Object XML Syntax

```
<Channel name="SampleXML" provider="XMLProvider">
  <Properties>
    <String name="refreshTime" value="600"/>
    <String name="title" value="XML Test Channel"/>
    <String name="description" value="This is a test of the XML Provider
system"/>
    <String name="url"
value="file:///etc/opt/SUNWps/desktop/default/SampleXML/getQuotes.xml"/>
    <String name="xslFileName"
value="/etc/opt/SUNWps/desktop/default/SampleXML/html_stockquote.xsl"/>
  </Properties>
</Channel>
```

Container Object

A container object is identical to a channel object, except that it primarily generates its content by aggregating the content of other (its child) channels. A container object allows for available and selected channel lists and can contain leaf channel definitions. A leaf channel is typically aggregated on a page with other channels and generates its own content. A container channel primarily generates content by aggregating the content of one or more leaf channels. Both leaf and container providers are building blocks in that they can be extended (through their public interfaces) to create new or custom providers.

The leaf building-block providers include:

- JSPPProvider

- XMLProvider
- URLScrapperProvider

The container building-block providers include:

- Tab
- Table
- Single

See the *Sun ONE Portal Server 6.2 Developer's Guide* for more information on the leaf and container building-block providers.

Code Example 1-2 Example Container Object XML Syntax

```

<Container name="JSPTableContainer" provider="JSPTableContainerProvider">
  <Properties>
    <String name="title" value="JSP Based Table Container Channel"/>
    <String name="contentPage" value="toptable.jsp"/>
    <String name="description" value="This is a test for front table
containers"/>
    <String name="Desktop-fontFacel" value="Sans-serif"/>
    <Collection name="categories">
      <String value="Personal Channels"/>
      <String value="Sample Channels"/>
      <String value="News Channels"/>
    </Collection>
    ...
  </Properties>

  <Available>
    <Reference value="UserInfo"/>
    <Reference value="MailCheck"/>
    ...
  </Available>

  <Selected>
    <Reference value="UserInfo"/>
    <Reference value="MailCheck"/>
    ...
  </Selected>

  <Channels> ...leaf definitions go here...</Channels>

</Container>

```

Provider Object

A provider object is a pointer to the display profile provider definition. The provider is a contract between ProviderContext and channel instance (provider object).

The display profile provider definition contains the information necessary for a client of the display profile to construct the provider object, namely, the Java™ class name.

The provider definition sets default property values for all channels that point to this provider. Channel-specific properties are only necessary when the provider defaults are not appropriate. The provider display profile object should contain default values for all properties that are used in the provider Java object. For example, if the provider Java code contains:

```
getStringProperty("color")
```

the provider display profile object should have a default value for color.

Code Example 1-3 Example Provider Object XML Syntax

```
<Provider name="XMLProvider"
class="com.sun.portal.providers.xml.XMLProvider">
  <Properties>
    <String name="title" value="*** XML Provider ***"/>
    <String name="description" value="*** DESCRIPTION ***"/>
    <String name="width" value="thick"/>
    <String name="refreshTime" value="0" advanced="true"/>
    <Boolean name="isEditable" value="false" advanced="true"/>
    <String name="helpURL" value="desktop/xmlchann.htm"
advanced="true"/>
    <String name="fontFace1" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <String name="url"
value="file:///etc/opt/SUNWps/desktop/default/xml/getQuotes.xml"/>
    <String name="xslFileName" value="html_stockquote.xsl"/>
    <Integer name="timeout" value="100"/>
    <String name="inputEncoding" value="iso-8859-1"/>
    <String name="urlScrapperRulesetID" value="default_ruleset"/>
    <Boolean name="cookiesToForwardAll" value="true"/>
    <Collection name="cookiesToForwardList">
      </Collection>
    </Properties>
  </Provider>
```

Modifying the Display Profile

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

- Using the Sun ONE Identity Server software administration console Channel and Container Management link.
- Using the Sun ONE Identity Server 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 information on using the `dpadmin` command, see the *Sun ONE Portal Server 6.2 Administrator's Guide*.

Where to Go for More Information

See [Chapter 2, “Display Profile Properties,”](#) [Chapter 3, “Container Provider Display Profile Definitions and Properties,”](#) [Chapter 4, “Leaf Provider Display Profile Definitions and Properties,”](#) and [Chapter 5, “Display Profile Channel Definitions”](#) for a description of the Sun ONE Portal Server software default display profile definitions for out-of-the-box providers, containers, and channels.

Using JavaServer Pages and Template Files

To generate the rendered Desktop user interface (what the industry refers to as the “presentation”), Sun ONE 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.

Sun ONE 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 Sun ONE 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`. The sampleportal 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 Sun ONE Identity Server software administration console specifies from what subdirectory to retrieve either the JSP or template files for the Desktop. For more information, see [“JSP and Template Files Lookup Scenario” on page 38](#).

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.

The Desktop and JavaServer Pages

The JSPProvider class reads in the JSP, compiles it, and runs it to produce the channel content.

The `JSPProvider` class reads at most three JSPs, one for content, one for the Edit page, and one to process the form submission from the Edit page. All other JSPs used in a JSP-based channel are referenced from one of those JSPs, either by an `include` or a `forward` statement.

A simple JSP-based channel can have just one JSP. Multiple JSPs are useful when a single part of the Desktop has to be replicated in several places. For example, consider a channel that has several display modes based on links clicked in that channel. Further, assume that the channel has a banner that must be displayed in all modes but one. You could construct a JSP to reference a `banner.jsp` file that captures common formatting that is used in multiple branches of the logic. If you didn't use this method, you would need to duplicate the content from the `banner.jsp` file, which is more difficult to maintain if that content needs to be changed. See [Appendix C, "JavaServer Pages Reference"](#) for more information.

The Desktop and Template Files

Providers that use template files hardcode their names and the template file names are not configurable in the Display Profile. For Providers that are based on JSP provider, the names of the JSP used by provider can be administratively changed as it is a property of the channel.

Both JSP and Desktop templates serve the purpose of separating business and presentation logic in the Portal Server. JSP is an accepted standard and is widely employed in many web-based applications. Desktop templates pre-date the emergence of JSP. JSP has many advantages over Desktop templates.

It is highly recommended that new Portal Server providers be based on JSP and the Portal Server `JSPProvider`. However, there may be cases where the simplicity of Desktop templates provides an advantage. Desktop templates are fully supported in Portal Server. Many pieces of the product, such as the communications channels, continue to use them.

NOTE To ensure backward compatibility with the iPlanet Portal Server 3.0 software, it is possible in Sun ONE Portal Server 6.2 software to have Desktop channels that are HTML template based. If you are migrating from iPlanet Portal Server 3.0 Service Pack 5 to Sun ONE Portal Server 6.2, you can migrate your HTML templates. See the *Sun ONE Portal Server 6.2 Migration Guide* for more information.

Changing the Desktop Type

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.

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.
```

3. Use the Sun ONE Identity Server 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, you would specify `sesta` as the value.

See the *Sun ONE Portal Server 6.2 Administrator's Guide* for more information on editing Desktop attributes.

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 *Sun ONE Portal Server 6.2 Developer's Guide* for more information on routing containers.

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 Sun ONE Portal Server software supplies. Not all Desktop channels need to be JSP-based. Channels migrated from iPlanet Portal Server 3.0 still work using HTML-based templates.

A tag library is exposed through Tag Library Descriptors (TLD) files, so tags are in their appropriate functional area. See [Appendix B, “JavaServer Pages Tag Library Reference”](#) for more information.

What Is the Sample Portal?

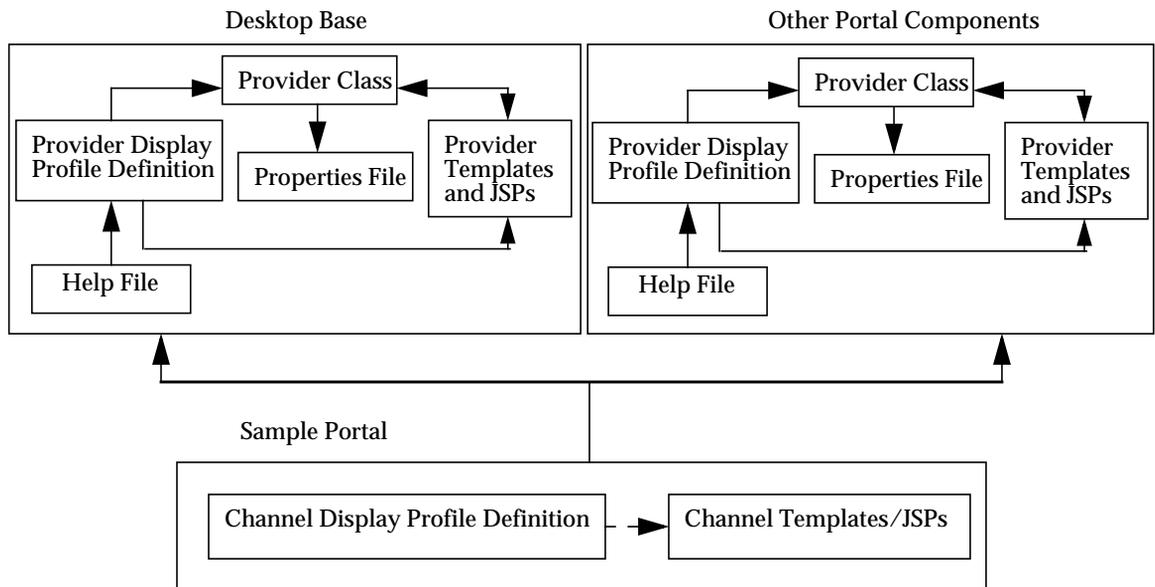
When you install Sun ONE Portal Server software, you can choose to install the sample portal. The sample portal includes five example Desktops that show the possibilities of Sun ONE Portal Server software. In this way you can quickly get a feel for the kinds of containers that are possible to design.

Conceptually, the Desktop is split into the following three well-defined components (see [Figure 1-2 on page 33](#) also):

- **Base Desktop** - This includes Provider Java classes (based on the Provider API), provider display profile definitions and resource bundles (also referred to as properties files), display profile definitions of channels referenced by base Desktop XML or base Desktop JSPs and templates, default templates and JSPs (installed in `/etc/opt/SUNWps/desktop/default` directory), and help files.
- **Sample portal** - This includes organization level display profile definitions (such as themes and channel display profile definitions), templates and JSPs for the example Desktops in the sample portal (installed in `/etc/opt/SUNWps/desktop/sampleportal` directory), and user definition, Desktop display profile definitions, and templates and JSPs for the Authless user.
- **Other components** - This includes the Provider Java classes for their component specific providers, provider display profile definitions for their component specific providers and provider resource bundles (properties files), display profile definition for channels referenced by component display profile XML or component templates and JSPs, default templates and JSPs, and help files.

The sample portal has a dependency on the base Desktop and other components and cannot be installed if the base Desktop and other components are not installed. The base Desktop and other components are installed as part of the Sun ONE Portal Server software.

Figure 1-2 Desktop Components



The sample portal can also serve as a place to start when building your own site's portal. You can customize the containers and use the building-block providers, such as XMLProvider and JSPPProvider, to add customized content. The sample portal also includes content providers, such as BookmarkProvider, that cannot be extended but that can be used to provide content.

If the existing building-block and content providers do not meet your needs, you can either extend an existing building-block provider (content providers are not public and hence not extendible), or develop custom building-block providers. If either of these methods do not suit your needs, you can develop a custom provider. See the *Sun ONE Portal Server 6.2 Developer's Guide* for more information on developing custom providers.

NOTE 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 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 lose any changes you might have made to the sample portal files, as the patch would only overwrite the initially installed sample files.

[Table 1-1 on page 34](#) shows the five Desktops, which make up the sample portal. This two column table lists the containers (Desktops) in the first column and a brief description in the second column.

Table 1-1 Sun ONE Portal Server Sample Desktop Containers

Container (Desktop) Type	Description
JSPTabContainer	Generates a JSP-based tab Desktop.
JSPTableContainer	Generates a JSP-based table Desktop.
TemplateTableContainer	Generates a template-based table Desktop.
TemplateTabContainer	Generates a template-based tab Desktop.
FrameTabContainer	Generates a frame-based Desktop.

See [Chapter 6, “Understanding the Sample Portal”](#) for a complete description of the sample Desktop containers and how they function.

Creating a New Desktop

Creating a new Desktop involves seven basic steps:

1. Developing the container
2. Editing the display profile
3. Developing JSPs or template files
4. Deploying JSP or template files
5. Loading the display profile at the appropriate LDAP nodes

6. (Optional) Creating a resource bundle
7. Testing access to the new Desktop

The following sections describe each of these steps in detail.

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.

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

NOTE You can also use the Sun ONE Identity Server 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. See [Chapter 3, “Container Provider Display Profile Definitions and Properties”](#) for more information.

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

Provider element:

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

Java class:

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

Developing 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 [Appendix A, “Desktop Template and Tag Reference”](#) and [Appendix C, “JavaServer Pages Reference”](#) 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.

Deploying JSP or Template Files

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.

JSP and Template Files Lookup Scenario

Sun ONE Portal Server software uses a specific lookup scenario to find the JSP and template files it needs for containers. The lookup scenario relies on the following parameters.

- Desktop type, for example `default` (set in Sun ONE Identity Server software administration console). Note that desktop type is now a comma separated string list and so the look up will be based on the desktop type(s) that are defined in the desktop type attribute.
- Preferred Locale, for example, `en_US` (set by users through the Sun ONE Identity Server software administration console in the “User” setting)
- Client type, for example `html` (set through the Sun ONE Identity Server software administration console Client Detection service)
- Channel name, for example, `newSingleContainer` (set in the display profile)
- Provider name, for example, `JSPSingleContainerProvider` (set in the display profile)

Based on the above parameters, the lookup scenario would search for the JSPs in the example in this order:

```

/etc/opt/SUNWps/desktop/default_en_US/newSingleContainer/html/newsinglecontent.jsp
/etc/opt/SUNWps/desktop/default_en_US/JSPSingleContainerProvider/html/newsinglecontent.jsp
/etc/opt/SUNWps/desktop/default_en_US/newSingleContainer/newsinglecontent.jsp
/etc/opt/SUNWps/desktop/default_en_US/JSPSingleContainerProvider/newsinglecontent.jsp
/etc/opt/SUNWps/desktop/default/newSingleContainer/newsinglecontent.jsp
(The system finds it here.)
/etc/opt/SUNWps/desktop/default/JSPSingleContainerProvider/newsinglecontent.jsp

```

Use this order to decide the final location of your own JSPs.

See [“Changing the Desktop Type” on page 30](#) for more information on how to modify the Desktop type.

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 Sun ONE Identity Server software administration console (as long as you are not using Netscape 4.7x) or the Upload link.

See the *Sun ONE Portal Server 6.2 Administrator's 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. See [“Creating and Extending a Provider Class” on page 40](#) for more information.

Accessing the Desktop

You can access the Desktop in one of the following ways:

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 listed in [Table 1-1 on page 34](#).

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.

Creating and Extending a Provider Class

A provider is responsible for providing the content for a channel as well as defining the various channel properties such as title and description. A provider can optionally implement edit functionality.

You create a provider by creating a new provider class. The provider class is a Java class that implements all the methods defined in the Provider interface. Sun ONE Portal Server software supplies basic implementation classes that implement the provider interface, ProviderAdapter. The ProfileProviderAdapter extends ProviderAdapter and adds some convenient methods that are indirectly defined in the ProviderContext class. Most of the time, you will extend ProfileProviderAdapter for your provider class. All building-block and content providers extend the ProfileProviderAdapter, either directly or indirectly.

See the *Sun ONE Portal Server 6.2 Developer's Guide* for details about creating and extending a provider class.

Display Profile Properties

This chapter describes the display profile global properties, and provider and channel specific properties.

This chapter contains the following sections:

- [Overview of Display Profile Properties](#)
- [Display Profile Global Properties](#)

Overview of Display Profile Properties

Display profile properties control all aspects of a channel, including:

- Content (available and selected channels)
- Position in the Desktop
- Controls

Display profile properties specify the per-channel configuration in the portal Desktop. Such properties define the visual representation of a channel in so much as the visual representation of the channel is affected by a display profile property.

Display Profile Properties

The display profile properties are contained in a properties “bag.” A bag is simply a grouping mechanism for display profile entities such as channels, providers, and properties. The property itself does not have a properties bag associated with it.

You associate properties with the following display profile objects:

- `<Properties>` definition

- `<Provider>` definition
- `<Channel>` definition
- `<Container>` definition

There are three basic categories of properties; they are:

Global Global properties are accessible to all channels. You set global properties, which are shared by all channels, in the `<Properties>` `</Properties>` definition. Themes are an example of a global property. You define the theme data globally to share it among all channels. See [“Display Profile Global Properties” on page 44](#) for more information.

NOTE Do not use global properties as defaults for all channels. Instead, use the `<Provider>` definition, as it sets the property interface used by the provider object that will use the `<Provider>` definition.

Provider Provider properties serve two purposes:

1. They define a property template or schema, defining the properties that will be used by all channels based on the provider.
2. The specific values in the provider serve as default values for channels.

If the property is not defined in channels based on this provider, the default value is used. If the default value is overridden by setting the value within the channel definition, then that value is used. By customizing a provider’s property values, you can customize all channels that the provider generates.

Channel Channel properties are available to the channel in which that properties are associated with. By customizing an individual channel’s properties, you customize that particular channel.

NOTE Properties set in the `<Provider>` definitions are defaults for channels based on that provider. Properties set in `<Channel>` definitions override the defaults in the provider definition to customize the channel. For example, `URLScaperProvider` defines a `url` property. A default does not make sense here, thus a channel would naturally override this value.

Containers are simply channels that generate the majority of their content by executing other channels (or containers). Many of the properties defined for containers pertain to how to gather and arrange content from other channels ([jtb-3](#)). For example, properties set in the `<Container>` definition can describe how to display the contained channels in the container, including: the layout of the container (thin-wide, wide-thin, or thin-wide-thin), a list of the contained channels, the position of the channel (the row and column number), and the window state of the contained channels (maximized, minimized, or detached).

Lower priority display profile documents can overwrite properties of higher priority display profile documents using merge locking. That is, the lock stops the merge on a particular property or value. See the *Sun ONE Portal Server 6.2 Administrator's Guide* for a complete discussion of the semantics of the display profile merging.

Display Profile Property Types

[Table 2-1](#) lists the property types for provider definitions. These can be used with leaf and container providers. This three column table lists the property types in the first column, a brief description in the second column, and an example in the third.

Table 2-1 Display Profile General Property Types

Property Type	Definition	Example
Boolean	An atomic object representing a Boolean value.	<code><Boolean name="removable" value="true"/></code>
Collection	An object representing either a list or hash table. A collection is a type of property, or named bag, in which to put other properties.	<code><Collection name="channelsRow"> <String name="MailCheck" value="4"/> <String name="App" value="5"/> </Collection></code>

Table 2-1 Display Profile General Property Types (Continued)

Property Type	Definition	Example
ConditionalProperty	<p>Defines the filtering criteria. The most common conditions are <code>locale</code> and <code>clientType</code>, but the API is generic in that it allows you to define and base properties on any sort of condition. <code>condition</code> and <code>value</code> are required attributes.</p> <p>In the administration console, the conditional properties are displayed as <code>condition-value</code> and can be edited like collections. The conditional properties can be nested and can be added to a channel or inside another conditional property. Use the Add Property page to add a new conditional property.</p>	<pre><ConditionalProperties condition="locale"> <String name="en_US" value="English (United States)"/> </ConditionalProperties></pre>
Integer	An atomic object representing an integer value.	<pre><Integer name="numberOfHeadlines" value="7"/></pre>
Reference	An object representing a pointer to a channel definition (that is, to a channel name in a container's selected and available channel lists.) Reference is an unnamed string useful for design tools to be able to distinguish such things from strings.	<pre><Reference value="UserInfo"/></pre>
String	An atomic object representing a string value.	<pre><String name="title" value="Table Container Channel 1"/></pre>

Display Profile Global Properties

There are no global properties defined in the base Desktop. Global properties are added via the sample portal installation. So, if you did not install the sample portal, by default, you do not have any global properties defined in the base Desktop.

Use global properties to assign properties that apply to all channels. For example, [Code Example 2-1 on page 45](#) shows (a snippet of) the global properties defined in the `dp-org.xml` display profile file that is part of the sample portal. You assign the global properties inside the `<Properties>` `</Properties>` definition by using tags such as `<Collection>` `</Collection>`, `<String>` `</String>`, `<Integer>` `</Integer>`, and so on.

Code Example 2-1 Global Properties Sample in the Display Profile dp-org.xml File

```

<DisplayProfile version="1.0" priority="10">
  <Properties>

    <Collection name="GlobalThemes" propagate="false">
      <Collection name="SunTheme">
        ...
      </Collection>
    </Collection>
    <Collection name="UserTheme">
      ...
    </Collection>
    <String name="docroot" value="/docs/" />
    <ConditionalProperties condition="locale">
      <String name="en_US" value="English (United States)" />
    </ConditionalProperties>
    <String name="helpURL" value="en/desktop/usedesk.htm" advanced="true" />
    <Collection name="userDefinedChannels" propagate="false" />
  </Properties>

```

The following is a list of all the global attributes available with the default installation of the sample portal. This two column table lists the attributes in the first (left) column and a brief description in the second (right) column.

GlobalThemes	<p>Defines the global themes for the Desktop. Themes are mainly focused on channel decoration like background color, channel border color, border width, and font face. Custom themes give the end user the ability to change the look and feel of the Desktop beyond the preset themes.</p> <p>Global themes can be added in the display profile and changed by users in their Desktops. See “Customizing the Global Themes” on page 259 for information on adding global themes.</p>
UserTheme	<p>Defines the theme that shows up in the user’s Desktop. The value must be one of the collection values defined in GlobalThemes. In dp-org.xml file, the value can be either theme1 or theme2. When users customize their Desktops, the value UserTheme will change.</p>
locales	<p>Used by UserInfoProvider to form the Language pull-down list.</p>

Display Profile Global Properties

<code>docroot</code>	Specifies the online help doc root relative to the installed <code>portal/static</code> location. See the Javadocs for more information on the <code>getHelp()</code> method in the <code>ProviderContext</code> API.
<code>helpURL</code>	Specifies a default help file that is used by all containers. If <code>helpURL</code> is specified in the container provider definition, then that one is used.
<code>userDefinedChannels</code>	Specifies the page to allow users to Create New Channel.

Container Provider Display Profile Definitions and Properties

This chapter contains information on the display profile definitions and the properties of the building-block and internally used container providers that ship with Sun™ ONE Portal Server software.

This chapter contains the following sections:

- [Display Profile Definitions for Building-Block Container Provider](#)
- [Display Profile Definitions for Internally Used Container Providers](#)
- [Container Provider Display Profile Properties](#)

Display Profile Definitions for Building-Block Container Provider

Container providers enable you to aggregate channels inside the Desktop. The container building-block providers are building blocks in a sense since you can also customize them or use them differently by changing the container properties. They include:

- [JSPTableContainerProvider](#)
- [JSPTabContainerProvider](#)
- [JSPSingleContainerProvider](#)
- [TemplateTableContainerProvider](#)
- [TemplateTabContainerProvider](#)

See the Javadocs for more information on these containers.

JSPTableContainerProvider

JSPTableContainerProvider is an extension of JSPPProvider. This JSP table provider displays the content channels in a table.

[Code Example 3-1](#) shows the display profile definition for JSPTableContainerProvider.

Code Example 3-1 Display Profile Definition for JSPTableContainerProvider

```

<Provider name="JSPTableContainerProvider"
class="com.sun.portal.providers.containers.jsp.table.JSPTableContainerProvider">
  <Properties>
    <String name="contentPage" value="table.jsp"/>
    <Integer name="timeout" value="60"/>
    <Integer name="layout" value="1"/>
    <Boolean name="showExceptions" value="false"/>
    <Boolean name="parallelChannelsInit" value="false"/>
    <String name="title" value="*** Table Container Provider ***"/>
    <String name="description" value="*** DESCRIPTION ***"/>
    <String name="refreshTime" value="" advanced="true"/>
    <String name="width" value="thin"/>
    <String name="fontFacel" value="Sans-serif"/>
    <String name="Desktop-fontFacel" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <String name="maximizedChannel" value="" />
    <String name="presetThemeChannel" value="JSPPresetThemeContainer" advanced="true"/>
    <String name="customThemeChannel" value="JSPCustomThemeContainer" advanced="true"/>
    <String name="parentTabContainer" value="FrameTabContainer" advanced="true"/>
    <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"/>
    <Collection name="categories">
      <String value="Personal Channels"/>
      <String value="Sample Channels"/>
      <String value="News Channels"/>
      <String value="Yahoo Sample Channels"/>
    </Collection>
    <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"/>
  </Properties>
</Provider>

```


Code Example 3-2 Display Profile Definition for JSPTabContainerProvider (Continued)

```

<String name="title" value="*** Tab Container Provider ***"/>
<String name="description" value="*** DESCRIPTION ***"/>
<String name="refreshTime" value="0"/>
<String name="width" value="thin" advanced="true"/>
<String name="fontFace1" value="Sans-serif"/>
<String name="productName" value="Sun ONE Portal Server"/>
<String name="presetThemeChannel" value="" advanced="true"/>
<String name="customThemeChannel" value="" advanced="true"/>
<Boolean name="isEditable" value="true" advanced="true"/>
<String name="editType" value="edit_complete" advanced="true"/>
<String name="editContainerName" value="JSPEditContainer" advanced="true"/>
<String name="startTab" value=""/>
<Integer name="maxTabs" value="6"/>
<String name="makeTabProvider" value="JSPTabCustomTableContainerProvider"
advanced="true"/>
<String name="makeTabChannel" value="JSPTabCustomTableContainer" advanced="true"/>
<Integer name="channelNumber" value="0"/>
<String name="contentChannel" value=""/>
<Collection name="TabProperties">
  <Collection name="JSPTabCustomTableContainer">
    <Boolean name="removable" value="true"/>
    <Boolean name="renamable" value="true"/>
    <Boolean name="predefined" value="false"/>
  </Collection>
</Collection>
</Properties>
</Provider>

```

On the sample portal Desktop, by default, the following list of container channels use this provider to display pages on the Desktop:

JSPTabContainer
FrameTabContainer

For more information, see [“Display Profile Definition for Container Channels”](#) in Chapter 3, [“Container Provider Display Profile Definitions and Properties.”](#)

JSPSingleContainerProvider

JSPSingleContainerProvider is an extension of the JSP container provider. The single container provider displays one channel in it.

[Code Example 3-3 on page 51](#) shows the display profile definition for JSPSingleContainerProvider.

Code Example 3-3 Display Profile Definition for JSPSingleContainerProvider

```

<Provider name="JSPSingleContainerProvider"
class="com.sun.portal.providers.containers.jsp.single.JSPSingleContainerProvider">
  <Properties>
    <String name="contentPage" value="single.jsp"/>
    <Boolean name="showExceptions" value="false"/>
    <String name="title" value="*** Single Container Provider ***"/>
    <String name="description" value="*** DESCRIPTION ***"/>
    <String name="refreshTime" value="0" advanced="true"/>
    <String name="width" value="thin" advanced="true"/>
    <String name="fontFacel" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="editType" value="" advanced="true"/>
    <String name="editContainerName" value="JSPEditContainer" advanced="true"/>
    <String name="presetThemeChannel" value="" advanced="true"/>
    <String name="customThemeChannel" value="" advanced="true"/>
  </Properties>
</Provider>

```

On the sample portal Desktop, by default, the following list of container channels use this provider to display pages on the Desktop:

JSPEditContainer
 TabJSPEditContainer
 JSPContentContainer
 JSPCreateChannelContain
 er
 JSPCustomThemeContainer
 JSPDynamicSingleContain
 er
 JSPLayoutContainer
 JSPPopupContainer
 JSPPresetThemeContainer

For more information, see [“Display Profile Definition for Container Channels” in Chapter 3, “Container Provider Display Profile Definitions and Properties.”](#)

TemplateTableContainerProvider

TemplateTableContainerProvider is the template-based table container. This provider displays the content channels in a table. [Code Example 3-4](#) shows the display profile definition for TemplateTableContainerProvider.

Code Example 3-4 Display Profile Definition for TemplateTableContainerProvider

```
<Provider name="TemplateTableContainerProvider"
class="com.sun.portal.providers.containers.template.table.TemplateTableContainerProvider">
  <Properties>
    <String name="title" value="*** Desktop Front Page ***"/>
    <String name="description" value="*** DESCRIPTION ***"/>
    <Integer name="timeout" value="60"/>
    <Integer name="layout" value="1"/>
    <String name="refreshTime" value="0"/>
    <String name="width" value="thin"/>
    <String name="fontFacel" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="editType" value="edit_complete" advanced="true"/>
    <String name="editContainerName" value="TemplateEditContainer" advanced="true"/>
    <String name="maximizedChannel" value="" />
    <String name="presetThemeChannel" value="" advanced="true"/>
    <String name="customThemeChannel" value="" advanced="true"/>
    <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"/>
    <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"/>
    <Collection name="channelsBackgroundColor"/>
    <Collection name="channelsIsMinimized" advanced="true"/>
    <Collection name="channelsIsDetached" advanced="true"/>
    <Collection name="channelsHasFrame" advanced="true"/>
    <Collection name="channelsIsMinimizable"/>
    <Collection name="channelsIsMaximizable"/>
    <Collection name="channelsRow" advanced="true"/>
    <Collection name="channelsColumn" advanced="true"/>
    <Collection name="channelsIsMovable"/>
    <Collection name="channelsIsDetachable"/>
    <Collection name="channelsIsRemovable"/>
```

Code Example 3-4 Display Profile Definition for TemplateTableContainerProvider (Continued)

```

    <Collection name="borderlessChannels"/>
  </Properties>
</Provider>

```

On the sample portal Desktop, by default, the TemplateTableContainer (container channel) use this provider to display pages on the Desktop.

TemplateTabContainerProvider

TemplateTabContainerProvider is the template-based tab container.

TemplateTabContainerProvider contains support for a number of tabs in it. [Code Example 3-5](#) shows the display profile definition for TemplateTabContainerProvider.

Code Example 3-5 Display Profile Definition for TemplateTabContainerProvider

```

<Provider name="TemplateTabContainerProvider"
class="com.sun.portal.providers.containers.template.tab.TemplateTabContainerProvider">
  <Properties>
    <String name="title" value="*** Tabbed Front Page ***"/>
    <String name="description" value="*** DESCRIPTION ***"/>
    <String name="refreshTime" value="" advanced="true"/>
    <String name="width" value="thin" advanced="true"/>
    <String name="fontFacel" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="editType" value="edit_complete" advanced="true"/>
    <String name="editContainerName" value="TemplateEditContainer" advanced="true"/>
    <String name="presetThemeChannel" value="" advanced="true"/>
    <String name="customThemeChannel" value="" advanced="true"/>
    <String name="startTab" value="" />
    <Integer name="maxTabs" value="4"/>
    <String name="makeTabProvider" value="TemplateTabCustomTableContainerProvider"
advanced="true"/>
    <String name="makeTabChannel" value="TemplateTabCustomTableContainer"
advanced="true"/>
    <Integer name="channelNumber" value="0"/>
    <Collection name="TabProperties">
      <Collection name="TemplateTabCustomTableContainer">
        <Boolean name="removable" value="true"/>
        <Boolean name="renamable" value="true"/>
        <Boolean name="predefined" value="false"/>
      </Collection>
    </Collection>
  </Properties>
</Provider>

```

Code Example 3-5 Display Profile Definition for TemplateTabContainerProvider (Continued)

```
</Collection>
</Properties>
</Provider>
```

On the sample portal Desktop, by default, the TemplateTabContainer (container channel) uses this provider to display pages on the Desktop.

Display Profile Definitions for Internally Used Container Providers

Sun ONE Portal Server software uses other container providers internally to perform such tasks as creating new tabs and edit containers. They include:

- [JSPTabCustomTableContainerProvider](#)
- [JSPFrameCustomTableContainerProvider](#)
- [TemplateEditContainerProvider](#)
- [TemplateTabCustomTableContainerProvider](#)
- [PredefinedFrontPageFramePanelContainerProvider](#)
- [PredefinedSamplesFramePanelContainerProvider](#)
- [PredefinedToolsTemplatePanelContainerProvider](#)
- [PredefinedFrontPageTemplatePanelContainerProvider](#)
- [PredefinedSamplesTemplatePanelContainerProvider](#)
- [PredefinedSamplesTabPanelContainerProvider](#)
- [PredefinedFrontPageTabPanelContainerProvider](#)

JSPTabCustomTableContainerProvider

JSPTabCustomTableContainerProvider is used when a new tab is created in the user's JSP tab-based Desktop, and is specified in the `makeTabProvider` property of JSPTabContainer. JSPTabCustomTableContainerProvider is based on the JSP table container provider.

Code Example 3-6 shows the display profile definition for JSPTabCustomTableContainerProvider.

Code Example 3-6 Display Profile Definition for JSPTabCustomTableContainerProvider

```
<Provider name="JSPTabCustomTableContainerProvider"
class="com.sun.portal.providers.containers.jsp.table.JSPTableContainerProvider">
  <Properties>
    <ConditionalProperties condition="locale" value="en">
      <ConditionalProperties condition="locale" value="US">
        <String name="title" value="Make My Own Tab"/>
        <String name="description" value="Make From Scratch Tab"/>
      </ConditionalProperties>
    </ConditionalProperties>
    <String name="contentPage" value="tabcustomtable.jsp"/>
    <String name="title" value="Make My Own Tab"/>
    <String name="description" value="Make from Scratch Tab"/>
    <String name="Desktop-fontFacel" value="Sans-serif"/>
    <Integer name="timeout" value="60"/>
    <Integer name="layout" value="1"/>
    <Boolean name="showExceptions" value="false"/>
    <Boolean name="parallelChannelsInit" value="false"/>
    <Boolean name="defaultChannelIsMinimizable" value="true"/>
    <Boolean name="defaultChannelIsMaximizable" value="true"/>
    <Boolean name="defaultChannelIsMinimized" value="false"/>
    <Boolean name="defaultChannelIsDetached" value="false"/>
    <Boolean name="defaultChannelIsDetachable" value="true"/>
    <Boolean name="defaultChannelIsRemovable" value="true"/>
    <Boolean name="defaultChannelHasFrame" value="true"/>
    <Boolean name="defaultChannelIsMovable" value="true"/>
    <Boolean name="defaultBorderlessChannel" value="false" advanced="true"/>
    <String name="defaultChannelColumn" value="1"/>
    <String name="defaultChannelRow" value="1"/>
    <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_window" value="600"/>
    <String name="refreshTime" value="" advanced="true"/>
    <String name="width" value="thin" advanced="true"/>
    <String name="fontFacel" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <String name="maximizedChannel" value="" />
    <String name="presetThemeChannel" value="JSPPresetThemeChannel" advanced="true"/>
    <String name="customThemeChannel" value="JSPCustomThemeChannel" advanced="true"/>
    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="editType" value="edit_complete" advanced="true"/>
    <String name="editContainerName" value="JSPeditContainer" advanced="true"/>
    <Collection name="categories">
      <String value="Personal Channels"/>
      <String value="Sample Channels"/>
      <String value="News Channels"/>
      <String value="Search Channels"/>
    </Collection>
  </Properties>
</Provider>
```

Code Example 3-6 Display Profile Definition for JSPTabCustomTableContainerProvider (Continued)

```

        <String value="Yahoo Sample Channels"/>
    </Collection>
    <Collection name="Personal Channels">
        <String value="UserInfo"/>
        <String value="MailCheck"/>
        <String value="Bookmark"/>
        <String value="App"/>
    </Collection>
    <Collection name="Sample Channels">
        <String value="SampleSimpleWebService"/>
        <String value="SampleJSP"/>
        <String value="SampleXML"/>
        <String value="SampleURLScrapper"/>
    </Collection>
    <Collection name="News Channels">
        <String value="SamplerSS"/>
        <String value="Notes"/>
    </Collection>
    <Collection name="Search Channels">
        <String value="Search"/>
    </Collection>
    <Collection name="Yahoo Sample Channels">
        <String value="Yahoo_channel"/>
        <String value="Yahoo_quotes"/>
        <String value="Yahoo_weather"/>
        <String value="Yahoo_comics"/>
    </Collection>
    <Collection name="channelsIsMinimized" advanced="true"/>
    <Collection name="channelsHasFrame" advanced="true"/>
    <Collection name="channelsColumn" advanced="true"/>
    <Collection name="channelsRow" advanced="true"/>
    <Collection name="channelsIsMinimizable"/>
    <Collection name="channelsIsMaximizable"/>
    <Collection name="channelsIsDetached" advanced="true"/>
    <Collection name="channelsIsDetachable"/>
    <Collection name="channelsIsRemovable"/>
    <Collection name="channelsIsMovable"/>
    <Collection name="borderlessChannels"/>
</Properties>
</Provider>

```

JSPFrameCustomTableContainerProvider

JSPFrameCustomTableContainerProvider is used when a new frame is created in the user's JSP frameset-based Desktop, and is specified in the makeTabProvider property. JSPFrameCustomTableContainerProvider is based on the table container provider.

[Code Example 3-7](#) shows the display profile definition for JSPFrameCustomTableContainerProvider.

Code Example 3-7 Display Profile Definition for JSPFrameCustomTableContainerProvider

```

<Provider name="JSPFrameCustomTableContainerProvider"
class="com.sun.portal.providers.containers.jsp.table.JSPTableContainerProvider">
  <Properties>
    <ConditionalProperties condition="locale" value="en">
      <ConditionalProperties condition="locale" value="US">
        <String name="title" value="Make My Own Page"/>
        <String name="description" value="Make From Scratch Page"/>
      </ConditionalProperties>
    </ConditionalProperties>
    <String name="contentPage" value="framecustomtable.jsp"/>
    <String name="title" value="Make My Own Page"/>
    <String name="description" value="Make from Scratch Page"/>
    <String name="Desktop-fontFacel" value="Sans-serif"/>
    <Integer name="timeout" value="60"/>
    <Integer name="layout" value="1"/>
    <Boolean name="showExceptions" value="false"/>
    <Boolean name="parallelChannelsInit" value="false"/>
    <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="defaultChannelIsMovable" value="true"/>
    <Boolean name="defaultChannelHasFrame" value="true" advanced="true"/>
    <String name="defaultChannelColumn" value="1"/>
    <String name="defaultChannelRow" value="1"/>
    <Boolean name="defaultBorderlessChannel" value="false" advanced="true"/>
    <Integer name="thin_popup_height" value="200"/>
    <Integer name="thin_popup_width" value="500"/>
    <Integer name="thick_popup_height" value="200"/>
    <Integer name="thick_popup_width" value="600"/>
    <Integer name="fullwidth_popup_height" value="500"/>
    <Integer name="fullwidth_popup_window" value="600"/>
    <String name="refreshTime" value="" advanced="true"/>
    <String name="width" value="thin" advanced="true" advanced="true"/>
    <String name="helpURL" value="en/desktop/fdesktop.htm"/>
    <String name="fontFacel" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <String name="maximizedChannel" value=""/>
    <String name="presetThemeChannel" value="JSPPresetThemeChannel" advanced="true"/>
    <String name="customThemeChannel" value="JSPCustomThemeChannel" advanced="true"/>
    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="editType" value="edit_complete" advanced="true"/>
    <String name="editContainerName" value="JSPEditContainer"/>
    <String name="parentTabContainer" value="FrameTabContainer" advanced="true"/>
    <Boolean name="refreshParentContainerOnly" value="true" advanced="true"/>
    <Collection name="categories">
      <String value="Personal Channels"/>
      <String value="Sample Channels"/>
      <String value="News Channels"/>
    </Collection>
  </Properties>
</Provider>

```

Code Example 3-7 Display Profile Definition for JSPFrameCustomTableContainerProvider (Continued)

```

        <String value="UserInfo"/>
        <String value="MailCheck"/>
        <String value="Bookmark"/>
        <String value="App"/>
    </Collection>
    <Collection name="Sample Channels">
        <String value="SampleSimpleWebService"/>
        <String value="SampleJSP"/>
        <String value="SampleXML"/>
        <String value="SampleURLScrapper"/>
    </Collection>
    <Collection name="News Channels">
        <String value="SampleRSS"/>
        <String value="Notes"/>
        <String value="Search"/>
    </Collection>
    <Collection name="channelsIsMinimized" advanced="true"/>
    <Collection name="channelsHasFrame" advanced="true"/>
    <Collection name="channelsColumn" advanced="true"/>
    <Collection name="channelsRow"/>
    <Collection name="channelsIsMinimizable"/>
    <Collection name="channelsIsMaximizable"/>
    <Collection name="channelsIsDetached" advanced="true"/>
    <Collection name="channelsIsDetachable"/>
    <Collection name="channelsIsRemovable"/>
    <Collection name="channelsIsMovable"/>
    <Collection name="borderlessChannels"/>
</Properties>
</Provider>

```

TemplateEditContainerProvider

TemplateEditContainerProvider is used by the template-based containers (TemplateTabContainer and TemplateTableContainer) as their edit provider. If a channel's editType is EDIT_SUBSET, this provider is used to draw the frame for the Edit page.

Code Example 3-8 shows the display profile definition for TemplateEditContainerProvider.

Code Example 3-8 Display Profile Definition for TemplateEditContainerProvider

```

<Provider name="TemplateEditContainerProvider"
class="com.sun.portal.providers.containers.template.TemplateEditProvider">
  <Properties>
    <String name="title" value="*** Edit Provider***"/>
    <String name="description" value="*** DESCRIPTION ***"/>
  </Properties>
</Provider>

```

Code Example 3-8 Display Profile Definition for TemplateEditContainerProvider (Continued)

```

<Integer name="timeout" value="60"/>
<Integer name="layout" value="1"/>
<String name="refreshTime" value="0" advanced="true"/>
<Boolean name="isEditable" value="true" advanced="true"/>
<String name="editType" value="edit_complete" advanced="true"/>
<String name="width" value="thin" advanced="true"/>
<String name="fontFace1" value="Sans-serif"/>
<String name="productName" value="Sun ONE Portal Server"/>
<String name="presetThemeChannel" value="" advanced="true"/>
<String name="customThemeChannel" value="" advanced="true"/>
</Properties>
</Provider>

```

TemplateTabCustomTableContainerProvider

TemplateTabCustomTableContainerProvider is used when a new tab is created in the user's template-based tab Desktop.

TemplateTabCustomTableContainerProvider is based on the template-based table container provider.

[Code Example 3-9](#) shows the display profile definition for TemplateTabCustomTableContainerProvider.

Code Example 3-9 Display Profile Definition for TemplateTabCustomTableContainerProvider

```

<Provider name="TemplateTabCustomTableContainerProvider"
class="com.sun.portal.providers.containers.template.table.TemplateTableContainerProvider">
  <Properties>
    <ConditionalProperties condition="locale" value="en">
      <ConditionalProperties condition="locale" value="US">
        <String name="title" value="Make My Own Tab"/>
        <String name="description" value="Make From Scratch Tab"/>
      </ConditionalProperties>
    </ConditionalProperties>
  </ConditionalProperties>
  <Integer name="timeout" value="60"/>
  <String name="title" value="Make My Own"/>
  <String name="refreshTime" value="" advanced="true"/>
  <Integer name="layout" value="1"/>
  <String name="description" value="Make from Scratch Tab"/>
  <String name="width" value="thin" advanced="true"/>
  <String name="fontFace1" value="Sans-serif"/>
  <String name="productName" value="Sun ONE Portal Server"/>
  <Boolean name="isEditable" value="true" advanced="true"/>
  <String name="editType" value="edit_complete" advanced="true"/>
  <String name="editContainerName" value="TemplateEditContainer" advanced="true"/>
  <Boolean name="defaultChannelIsMinimizable" value="true" advanced="true"/>
  <Boolean name="defaultChannelIsMaximizable" value="true"/>

```

Code Example 3-9 Display Profile Definition for TemplateTabCustomTableContainerProvider

```

<Boolean name="defaultChannelIsMinimized" value="false" advanced="true"/>
<Boolean name="defaultChannelIsDetached" value="false" advanced="true"/>
<Boolean name="defaultChannelIsDetachable" value="true" advanced="true"/>
<Boolean name="defaultChannelIsRemovable" value="true" advanced="true"/>
<Boolean name="defaultChannelHasFrame" value="true" advanced="true"/>
<Boolean name="defaultChannelIsMovable" value="true" advanced="true"/>
<Boolean name="defaultBorderlessChannel" value="false" advanced="true"/>
<String name="defaultChannelColumn" value="1" advanced="true"/>
<String name="defaultChannelRow" value="1" advanced="true"/>
<String name="presetThemeChannel" value="" advanced="true"/>
<String name="customThemeChannel" value="" advanced="true"/>
<String name="maximizedChannel" value="" />
<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"/>
<Collection name="channelsBackgroundColor"/>
<Collection name="channelsIsMinimized" advanced="true"/>
<Collection name="channelsHasFrame" advanced="true"/>
<Collection name="channelsColumn" advanced="true"/>
<Collection name="channelsRow" advanced="true"/>
<Collection name="channelsIsMinimizable"/>
<Collection name="channelsIsMaximizable"/>
<Collection name="channelsIsDetached" advanced="true"/>
<Collection name="channelsIsDetachable"/>
<Collection name="channelsIsRemovable"/>
<Collection name="channelsIsMovable"/>
<Collection name="borderlessChannels"/>
</Properties>
</Provider>

```

PredefinedFrontPageFramePanelContainerProvider

PredefinedFrontPageFramePanelContainerProvider is the provider for the predefined tab for MyFrontPage tab when the user creates a new page based on an existing page from the make New Page page on FrameTabContainer. [Code Example 3-10](#) shows the display profile definition for PredefinedFrontPageFramePanelContainerProvider.

Code Example 3-10 Display Profile Definition for PredefinedFrontPageFramePanelContainerProvider

```

<Provider name="PredefinedFrontPageFramePanelContainerProvider"
class="com.sun.portal.providers.containers.jsp.table.JSPTableContainerPr
vider" version="2">
  <Properties>
    <ConditionalProperties condition="locale" value="en" >
      <ConditionalProperties condition="locale" value="US" >
        <String name="title" value="My Front Page"/>
        <String name="description" value="Your Front Page"/>
      </ConditionalProperties>
    </ConditionalProperties>
    <String name="title" value="My Front Page"/>
    <String name="description" value="Predefined Front Page"/>
    <String name="contentPage" value="table.jsp"/>
    <String name="presetThemeChannel" value="JSPPresetThemeContainer" advanced="true"/>
    <String name="customThemeChannel" value="JSPCustomThemeContainer" advanced="true"/>
    <String name="parentTabContainer" value="FrameTabContainer" advanced="true"/>
    <String name="Desktop-fontFace1" value="Sans-serif"/>
    <String name="refreshTime" value="" advanced="true"/>
    <String name="width" value="thin" advanced="true"/>
    <String name="fontFace1" value="Sans-serif"/>
    <String name="productName" value="Sun ONE 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="true" 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="Personal Channels"/>
      <String value="Sample Channels"/>
      <String value="News Channels"/>
    </Collection>
  </Properties>
</Provider>

```

Code Example 3-10 Display Profile Definition for PredefinedFrontPageFramePanelContainerProvider

```

    <Collection name="Personal Channels">
      <String value="UserInfo"/>
      <String value="MailCheck"/>
      <String value="Bookmark"/>
      <String value="App"/>
    </Collection>
    <Collection name="Sample Channels">
      <String value="SampleJSP"/>
      <String value="SampleXML"/>
    </Collection>
    <Collection name="News Channels">
      <String value="SampleRSS"/>
    </Collection>
    <Collection name="channelsRow" advanced="true">
      <String name="MailCheck" value="4"/>
      <String name="App" value="5"/>
      <String name="Bookmark" value="3"/>
      <String name="SampleRSS" value="2"/>
      <String name="SampleXML" value="2"/>
    </Collection>
    <Collection name="channelsIsRemovable" >
      <Boolean name="UserInfo" value="false"/>
    </Collection>
    <Collection name="channelsIsMinimized" advanced="true"/>
    <Collection name="channelsIsDetached" advanced="true"/>
    <Collection name="channelsHasFrame" advanced="true"/>
    <Collection name="channelsIsMinimizable"/>
    <Collection name="channelsIsMaximizable"/>
    <Collection name="channelsColumn" advanced="true"/>
    <Collection name="channelsIsMovable"/>
    <Collection name="channelsIsDetachable"/>
    <Collection name="borderlessChannels"/>
  </Properties>
</Provider>

```

PredefinedSamplesFramePanelContainerProvider

PredefinedSamplesFramePanelContainerProvider is the provider for the predefined tab for Samples tab when the user creates a new page based on an existing page from the make New Page page on FrameTabContainer. [Code Example 3-11](#) shows the display profile definition for PredefinedSamplesFramePanelContainerProvider.

Code Example 3-11 Display Profile Definition for PredefinedSamplesFramePanelContainerProvider

```

<Provider name="PredefinedSamplesFramePanelContainerProvider"
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="Samples"/>
        <String name="description" value="Samples Tab"/>
      </ConditionalProperties>
    </ConditionalProperties>
    <String name="title" value="Samples"/>
    <String name="description" value="Samples Tab"/>
    <String name="contentPage" value="table.jsp"/>
    <String name="presetThemeChannel" value="JSPPresetThemeContainer" advanced="true"/>
    <String name="customThemeChannel" value="JSPCustomThemeContainer" advanced="true"/>
    <String name="parentTabContainer" value="FrameTabContainer" advanced="true"/>
    <String name="Desktop-fontFacel" value="Sans-serif"/>
    <String name="refreshTime" value="" advanced="true"/>
    <String name="width" value="thin" advanced="true"/>
    <String name="fontFacel" value="Sans-serif"/>
    <String name="productName" value="Sun ONE 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="true" 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"/>
    </Collection>
  </Properties>
</Provider>

```

Code Example 3-11 Display Profile Definition for PredefinedSamplesFramePanelContainerProvider

```

        <String value="SampleURLScrapper" />
        <String value="Notes" />
        <String value="SampleSimpleWebService" />
    </Collection>
    <Collection name="channelsColumn" advanced="true">
        <String name="SampleURLScrapper" value="2" />
        <String name="Notes" value="2" />
        <String name="SampleSimpleWebService" value="2" />
    </Collection>
    <Collection name="channelsRow" advanced="true">
        <String name="SampleURLScrapper" value="2" />
        <String name="Notes" value="3" />
        <String name="SampleSimpleWebService" value="3" />
    </Collection>
    <Collection name="channelsIsMinimized" advanced="true" />
    <Collection name="channelsIsDetached" advanced="true" />
    <Collection name="channelsHasFrame" advanced="true" />
    <Collection name="channelsIsMinimizable" />
    <Collection name="channelsIsMaximizable" />
    <Collection name="channelsIsMovable" />
    <Collection name="channelsIsRemovable" />
    <Collection name="channelsIsDetachable" />
    <Collection name="borderlessChannels" />
</Properties>
</Provider>

```

PredefinedToolsTemplatePanelContainerProvider

PredefinedToolsTemplatePanelContainerProvider is the provider for the Tools tab on TemplateTabContainer. [Code Example 3-12 on page 64](#) shows the display profile definition for PredefinedToolsTemplatePanelContainerProvider.

Code Example 3-12 Display Provider Definition for PredefinedToolsTemplatePanelContainerProvider

```

<Provider name="PredefinedToolsTemplatePanelContainerProvider"
class="com.sun.portal.providers.containers.template.table.TemplateTableContainerProvider"
version="2">
    <Properties>
        <ConditionalProperties condition="locale" value="en" >
            <ConditionalProperties condition="locale" value="US" >
                <String name="title" value="Tools" />
                <String name="description" value="Predefined Tools" />
            </ConditionalProperties>
        </ConditionalProperties>
    </ConditionalProperties>
    <String name="title" value="Tools" />
    <String name="description" value="Predefined Tools" />
    <Integer name="timeout" value="240" />
    <Integer name="layout" value="1" />

```

Code Example 3-12 Display Provider Definition for PredefinedToolsTemplatePanelContainerProvider

```

<String name="refreshTime" value="" advanced="true"/>
<String name="width" value="thin" advanced="true"/>
<String name="fontFace1" value="Sans-serif"/>
<String name="productName" value="Sun ONE Portal Server"/>
<Boolean name="isEditable" value="true" advanced="true"/>
<String name="editType" value="edit_complete" advanced="true"/>
<String name="editContainerName" value="TemplateEditContainer" advanced="true"/>
<String name="maximizedChannel" value="" />
<String name="presetThemeChannel" value="" advanced="true"/>
<String name="customThemeChannel" value="" advanced="true"/>
<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"/>
<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"/>
<Collection name="channelsColumn" advanced="true">
  <String name="Notes" value="2"/>
</Collection>
<Collection name="channelsRow" advanced="true">
  <String name="MailCheck" value="3"/>
  <String name="App" value="5"/>
  <String name="Bookmark" value="4"/>
  <String name="Notes" value="3"/>
</Collection>
<Collection name="channelsIsRemovable">
  <Boolean name="UserInfo" value="false"/>
</Collection>
<Collection name="channelsBackgroundColor"/>
<Collection name="channelsIsMinimized" advanced="true"/>
<Collection name="channelsIsDetached" advanced="true"/>
<Collection name="channelsHasFrame" advanced="true"/>
<Collection name="channelsIsMinimizable"/>
<Collection name="channelsIsMaximizable"/>
<Collection name="channelsIsMovable"/>
<Collection name="channelsIsDetachable"/>
<Collection name="borderlessChannels"/>
</Properties>
</Provider>

```

PredefinedFrontPageTemplatePanelContainerProvider

PredefinedFrontPageTemplatePanelContainerProvider is the provider for the predefined tab for MyFrontPage tab when the user creates a new tab based on an existing tab from the make New Tab page on TemplateTabContainer. [Code Example 3-13](#) shows the display profile definition for PredefinedFrontPageTemplatePanelContainerProvider.

Code Example 3-13 Display Profile Definition for PredefinedFrontPageTemplatePanelContainerProvider

```
<Provider name="PredefinedFrontPageTemplatePanelContainerProvider"
class="com.sun.portal.providers.containers.template.table.TemplateTableContainerProvider"
version="2">
  <Properties>
    <ConditionalProperties condition="locale" value="en" >
      <ConditionalProperties condition="locale" value="US" >
        <String name="title" value="My Front Page"/>
        <String name="description" value="Your Front Page"/>
      </ConditionalProperties>
    </ConditionalProperties>
    <String name="title" value="My Front Page"/>
    <String name="description" value="Predefined Front Page"/>
    <Integer name="timeout" value="240"/>
    <Integer name="layout" value="1"/>
    <String name="refreshTime" value="" advanced="true"/>
    <String name="width" value="thin" advanced="true"/>
    <String name="fontFacel" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="editType" value="edit_complete" advanced="true"/>
    <String name="editContainerName" value="TemplateEditContainer" advanced="true"/>
    <String name="maximizedChannel" value=""/>
    <String name="presetThemeChannel" value="" advanced="true"/>
    <String name="customThemeChannel" value="" advanced="true"/>
    <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"/>
    <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"/>
    <Collection name="channelsColumn" advanced="true">
      <String name="SampleJSP" value="2"/>
    </Collection>
  </Properties>
</Provider>
```

Code Example 3-13 Display Profile Definition for PredefinedFrontPageTemplatePanelContainerProvider

```

    <String name="SampleXML" value="2"/>
  </Collection>
  <Collection name="channelsRow" advanced="true">
    <String name="MailCheck" value="3"/>
    <String name="SampleRSS" value="2"/>
    <String name="SampleXML" value="2"/>
    <String name="App" value="5"/>
    <String name="Bookmark" value="4"/>
  </Collection>
  <Collection name="channelsIsRemovable">
    <Boolean name="UserInfo" value="false"/>
  </Collection>
  <Collection name="channelsBackgroundColor"/>
  <Collection name="channelsIsMinimized" advanced="true"/>
  <Collection name="channelsIsDetached" advanced="true"/>
  <Collection name="channelsHasFrame" advanced="true"/>
  <Collection name="channelsIsMinimizable"/>
  <Collection name="channelsIsMaximizable"/>
  <Collection name="channelsIsMovable"/>
  <Collection name="channelsIsDetachable"/>
  <Collection name="borderlessChannels"/>
</Properties>
</Provider>

```

PredefinedSamplesTemplatePanelContainerProvider

PredefinedSamplesTemplatePanelContainerProvider is the provider for the predefined tab for Samples tab when the user creates a new tab based on an existing tab from the make New Tab page on TemplateTabContainer. [Code Example 3-14](#) shows the display profile definition for PredefinedSamplesTemplatePanelContainerProvider.

Code Example 3-14 Display Profile Definition for PredefinedSamplesTemplatePanelContainerProvider

```

<Provider name="PredefinedSamplesTemplatePanelContainerProvider"
class="com.sun.portal.providers.containers.template.table.TemplateTableContainerProvider"
version="2">
  <Properties>
    <ConditionalProperties condition="locale" value="en" >
      <ConditionalProperties condition="locale" value="US" >
        <String name="title" value="Samples"/>
        <String name="description" value="Samples Tab"/>
      </ConditionalProperties>
    </ConditionalProperties>
  </Properties>
  <String name="title" value="Samples"/>
  <String name="description" value="Samples Tab"/>
  <Integer name="timeout" value="240"/>

```

Code Example 3-14 Display Profile Definition for PredefinedSamplesTemplatePanelContainerProvider

```

<Integer name="layout" value="1"/>
<String name="refreshTime" value="" advanced="true"/>
<String name="width" value="thin" advanced="true"/>
<String name="fontFacel" value="Sans-serif"/>
<String name="productName" value="Sun ONE Portal Server"/>
<Boolean name="isEditable" value="true" advanced="true"/>
<String name="editType" value="edit_complete" advanced="true"/>
<String name="editContainerName" value="TemplateEditContainer" advanced="true"/>
<String name="maximizedChannel" value="" />
<String name="presetThemeChannel" value="" advanced="true"/>
<String name="customThemeChannel" value="" advanced="true"/>
<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"/>
<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"/>
<Collection name="channelsColumn" advanced="true">
  <String name="SampleURLScrapper" value="2"/>
  <String name="Notes" value="2"/>
  <String name="SampleSimpleWebService" value="2"/>
</Collection>
<Collection name="channelsRow" advanced="true">
  <String name="SampleRSS" value="2"/>
  <String name="SampleURLScrapper" value="2"/>
  <String name="SampleSimpleWebService" value="4"/>
  <String name="Notes" value="3"/>
</Collection>
<Collection name="channelsBackgroundColor"/>
<Collection name="channelsIsMinimized" advanced="true"/>
<Collection name="channelsIsDetached" advanced="true"/>
<Collection name="channelsHasFrame" advanced="true"/>
<Collection name="channelsIsMinimizable"/>
<Collection name="channelsIsMaximizable"/>
<Collection name="channelsIsMovable"/>
<Collection name="channelsIsDetachable"/>
<Collection name="channelsIsRemovable"/>
<Collection name="borderlessChannels"/>
</Properties>
</Provider>

```

PredefinedSamplesTabPanelContainerProvider

PredefinedSamplesTabPanelContainerProvider is the provider for the predefined tab for Samples tab when the user creates a new tab based on an existing tab from the make New Tab page on JSPTabContainer. [Code Example 3-15](#) shows the display profile definition for PredefinedSamplesTabPanelContainerProvider.

Code Example 3-15 Display Profile Definition for PredefinedSamplesTabPanelContainerProvider

```
<Provider name="PredefinedSamplesTabPanelContainerProvider"
class="com.sun.portal.providers.containers.jsp.table.JSPTableContainerPr
ovider" version="2">
  <Properties>
    <ConditionalProperties condition="locale" value="en" >
      <ConditionalProperties condition="locale" value="US" >
        <String name="title" value="Samples"/>
        <String name="description" value="Samples Tab"/>
      </ConditionalProperties>
    </ConditionalProperties>
    <String name="title" value="Samples"/>
    <String name="description" value="Samples 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"/>
    <String name="refreshTime" value="" advanced="true"/>
    <String name="width" value="thin" advanced="true"/>
    <String name="fontFace1" value="Sans-serif"/>
    <String name="productName" value="Sun ONE 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"/>
  </Properties>
</Provider>
```

Code Example 3-15 Display Profile Definition for PredefinedSamplesTabPanelContainerProvider

```

<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"/>
  <String value="Yahoo Sample Channels"/>
</Collection>
<Collection name="Sample Channels">
  <String value="SampleRSS"/>
  <String value="SampleURLScrapper"/>
  <String value="Notes"/>
  <String value="SampleSimpleWebService"/>
</Collection>
<Collection name="channelsColumn" advanced="true">
  <String name="SampleURLScrapper" value="2"/>
  <String name="Notes" value="2"/>
  <String name="SampleSimpleWebService" value="2"/>
</Collection>
<Collection name="channelsRow" advanced="true">
  <String name="SampleURLScrapper" value="2"/>
  <String name="Notes" value="3"/>
  <String name="SampleSimpleWebService" value="4"/>
</Collection>
  <Collection name="Yahoo Sample Channels">
    <String value="Yahoo_channel"/>
    <String value="Yahoo_quotes"/>
    <String value="Yahoo_weather"/>
    <String value="Yahoo_comics"/>
  </Collection>
<Collection name="channelsIsMinimized" advanced="true"/>
<Collection name="channelsIsDetached" advanced="true"/>
<Collection name="channelsHasFrame" advanced="true"/>
<Collection name="channelsIsMinimizable"/>
<Collection name="channelsIsMaximizable"/>
<Collection name="channelsIsMovable"/>
<Collection name="channelsIsRemovable"/>
<Collection name="channelsIsDetachable"/>
<Collection name="borderlessChannels"/>
</Properties>
</Provider>

```

PredefinedFrontPageTabPanelContainerProvider

PredefinedFrontPageTabPanelContainerProvider is the provider for the predefined tab for MyFrontPage tab used when the user creates a new tab based on an existing tab from the make New Tab page on JSPTabContainer. [Code Example 3-16](#) shows the display profile definition for PredefinedFrontPageTabPanelContainerProvider.

Code Example 3-16 Display Profile Definition for PredefinedFrontPageTabPanelContainerProvider

```

<Provider name="PredefinedFrontPageTabPanelContainerProvider"
class="com.sun.portal.providers.containers.jsp.table.JSPTableContainerPr
vider" version="2">
  <Properties>
    <ConditionalProperties condition="locale" value="en" >
      <ConditionalProperties condition="locale" value="US" >
        <String name="title" value="My Front Page"/>
        <String name="description" value="Your Front Page"/>
      </ConditionalProperties>
    </ConditionalProperties>
    <String name="title" value="My Front Page"/>
    <String name="description" value="Predefined Front Page"/>
    <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"/>
    <String name="refreshTime" value="" advanced="true"/>
    <String name="width" value="thin" advanced="true"/>
    <String name="fontFace1" value="Sans-serif"/>
    <String name="productName" value="Sun ONE 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="Personal Channels"/>
      <String value="Sample Channels"/>
      <String value="News Channels"/>
    </Collection>
  </Properties>
</Provider>

```

Code Example 3-16 Display Profile Definition for PredefinedFrontPageTabPageContainerProvider

```

<Collection name="Personal Channels">
  <String value="UserInfo"/>
  <String value="MailCheck"/>
  <String value="Bookmark"/>
  <String value="App"/>
</Collection>
<Collection name="Sample Channels">
  <String value="SampleJSP"/>
  <String value="SampleXML"/>
</Collection>
<Collection name="News Channels">
  <String value="SampleRSS"/>
</Collection>
<Collection name="channelsRow" advanced="true">
  <String name="MailCheck" value="4"/>
  <String name="App" value="5"/>
  <String name="Bookmark" value="3"/>
  <String name="SampleRSS" value="2"/>
  <String name="SampleXML" value="2"/>
</Collection>
<Collection name="channelsIsRemovable" >
  <Boolean name="UserInfo" value="false"/>
</Collection>
<Collection name="channelsIsMinimized" advanced="true"/>
<Collection name="channelsIsDetached" advanced="true"/>
<Collection name="channelsHasFrame" advanced="true"/>
<Collection name="channelsIsMinimizable"/>
<Collection name="channelsIsMaximizable"/>
<Collection name="channelsColumn" advanced="true"/>
<Collection name="channelsIsMovable"/>
<Collection name="channelsIsDetachable"/>
<Collection name="borderlessChannels"/>
</Properties>
</Provider>

```

Container Provider Display Profile Properties

This section contains:

- [Available and Selected List](#)
- [Common Properties for Table Container](#)
- [Common Properties for Tab Container](#)
- [Other Container Properties](#)

Available and Selected List

All containers must define a list of available and selected channels. The presence of these is what mainly distinguishes a container from a channel.

Conceptually, the available list defines the set of channels that can be displayed in the container. The selected list defines those that are actually displayed in the container.

To take a specific example, consider the table container. Table containers use the available channel list to store channels that the user may add to their Desktop. The selected list is used to store the set of channels that are visible in their portal page. Typically, the selected channels are a subset of the available channels.

NOTE Containers are not required to make use of the available and selected channel lists in the display profile. A container may manage its contained channels in other implementation dependent ways. However, it is recommended that containers use the display profile available and selected channel lists in order to standardize how they are administrated.

The following is a list of the required container properties. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

Available

Defines a list of all available channels for this container. The `<Available>` and `</Available>` tags define the list, and the `<Reference value=>` tag defines the list items. For example:

```
<Available>
  <Reference value="UserInfo" />
  <Reference value="MailCheck" />
  <Reference value="App" />
  <Reference value="Bookmark" />
</Available>
```

Selected

Defines a list of selected channels for this container. Only selected channels are displayed on the Desktop. The `<Selected>` and `</Selected>` tags define the list, and the `<Reference value=>` tag defines the list items. For example:

```
<Selected>
    <Reference value="UserInfo" />
    <Reference value="MailCheck" />
    <Reference value="App" />
    <Reference value="Bookmark" />
</Selected>
```

To edit the available and selected list for a container from the administration console:

1. Log in to the administration console and select Services View for your organization.
2. Select Portal Desktop and the Channel and Container Management link.
3. For the container

Common Properties for Table Container

The following are the common properties for table containers. This two column table lists the property tags in the first column and a brief description in the second column.

The `<Collection name>` `</Collection>` tags define a list to contain these properties, which are set with the `<String>` tag.

Table 3-1 Other Container Properties

Property Tag	Description
<code>parentTabContainer</code>	Contained table containers have the <code>parentTabContainer</code> property whose value is the name of the tab container in which the contained table container is contained. If the contained table container has to be used in some other tab container, change this property value to the respective tab container name.
<code>refreshParentContainerOnly</code>	

Table 3-1 Other Container Properties (Continued)

Property Tag	Description
<code>thin_popup_height</code>	Defines the window height in pixels for the <code>thin</code> channel in the detached window.
<code>thin_popup_width</code>	Defines the window width in pixels for the <code>thin</code> channel in the detached window.
<code>thick_popup_height</code>	Defines the window height in pixels for the <code>thick</code> channel in the detached window.
<code>thick_popup_width</code>	Defines the window width in pixels for the <code>thick</code> channel in the detached window.
<code>fullwidth_popup_height</code>	Defines the window height in pixels for the <code>full_top</code> or <code>full_bottom</code> channel in the detached window.
<code>fullwidth_popup_width</code>	Defines the window width in pixels for the <code>full_top</code> or <code>full_bottom</code> channel in the detached window.
<code>defaultChannelIsMinimizable</code>	Defines the <code>isMinimizable</code> default value for the channels in this container. If you define a default value, then you do not have to define <code>isMinimizable</code> for all the leaf channels in the container. You can change the value for a leaf channel in the container if needed.
<code>defaultchannelsIsMaximizabl e</code>	Defines the <code>isMaximizable</code> default value for the channels in this container. If you define a default value, then you do not have to define <code>isMaximizable</code> for all the leaf channels in the container. You can change the value for a leaf channel in the container if needed.
<code>defaultChannelIsMinimized</code>	Defines the <code>isMinimized</code> default value for the channels in this container. If you define a default value, then you do not have to define <code>isMinimized</code> for all the leaf channels in the container. You can change the value for a leaf channel in the container if needed.
<code>defaultChannelIsDetached</code>	Defines the <code>isDetached</code> default value for the channels in this container. If you define a default value, then you do not have to define <code>isDetached</code> for all the leaf channels in the container. You can change the value for a leaf channel in the container if needed.
<code>defaultChannelIsDetachable</code>	Defines the <code>isDetachable</code> default value for the channels in this container. If you define a default value, then you do not have to define <code>isDetachable</code> for all the leaf channels in the container. You can change the value for a leaf channel in the container if needed.
<code>defaultChannelIsRemovable</code>	Defines the <code>isRemovable</code> default value for the channels in this container. If you define a default value, then you do not have to define <code>isRemoveable</code> for all the leaf channels in the container. You can change the value for a leaf channel in the container if needed.
<code>defaultChannelHasFrame</code>	Defines the <code>hasFrame</code> default value for the channels in this container. If you define a default value, then you do not have to define <code>hasFrame</code> for all the leaf channels in the container. You can change the value for a leaf channel in the container if needed.

Table 3-1 Other Container Properties *(Continued)*

Property Tag	Description
<code>defaultChannelIsMovable</code>	Defines the <code>isMovable</code> default value for the channels in this container. If you define a default value, then you do not have to define <code>isMovable</code> for all the leaf channels in the container. You can change the value for a leaf channel in the container if needed.
<code>defaultChannelColumn</code>	Defines the column number default value for the channels in this container. If you define a default value, then you do not have to define the column number for all the leaf channels in the container. You can change the value for a leaf channel in the container if needed.
<code>defaultChannelRow</code>	Defines the row number default value for the channels in this container. If you define a default value, then you do not have to define row number for all the leaf channels in the container. You can change the value for a leaf channel in the container if needed.
<code>channelsIsMinimized</code>	Defines a collection property to contain the <code>isMinimized</code> value for channels in this container.
<code>channelsIsDetached</code>	Defines a collection property to contain the <code>isDetached</code> value for channels in this container.
<code>channelsHasFrame</code>	Defines a collection property to contain the <code>hasFrame</code> value for channels in this container.
<code>channelsIsMinimizable</code>	Defines a collection property to contain the <code>isMinimizable</code> value for channels in this container.
<code>channelsIsMaximizable</code>	Defines a collection property to contain the <code>isMaximizable</code> value for channels in this container.
<code>channelsRow</code>	Defines a collection property to contain the row number value for channels in this container.
<code>channelsColumn</code>	Defines a collection property to contain the column number value for channels in this container.
<code>channelsIsMovable</code>	Defines a collection property to contain the <code>isMovable</code> value for channels in this container.
<code>channelsIsDetachable</code>	Defines a collection property to contain the <code>isDetachable</code> value for channels in this container.
<code>channelsIsRemovable</code>	Defines a collection property to contain the <code>isRemovable</code> value for channels in this container.
<code>borderlessChannels</code>	Defines the collection property to contain the channel name and Boolean value pair for specifying borderless channels in this container. A value of <code>true</code> means the channel does not have border.
<code>defaultBorderlessChannel</code>	Defines the default value for the borderless channels in this container. If you define a default value, then you do not have to define <code>borderlessChannels</code> for all leaf channels in the container. You can change the value for a leaf channel in the container if needed.

Common Properties for Tab Container

The following is a list of the properties common to all `TabContainerProviders`. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

<code>startTab</code>	Tab that is displayed when the user logs in.
<code>makeTabChannel</code>	Container channel name to be used when the user creates a new tab.
<code>makeTabProvider</code>	Container provider to be used as a base provider when the user creates a new tab.
<code>maxTabs</code>	The maximum number of tabs that can be selected on the user's Desktop.
<code>channelNumber</code>	Used in the naming of newly created tabs by user.
<code>contentChannel</code>	The content channel to be used as the Content page for a user created tab.
<code>TabProperties</code>	The collection property <code>TabProperties</code> creates the new tab. There needs to be a one-to-one mapping between the contents of the <code>TabProperties</code> collection and the available or selected tabs. That is, for every tab specified in the available or selected list, a new collection needs to be defined inside <code>TabProperties</code> collection.

Other Container Properties

The following is a list of properties that are common to all container providers. This two column table lists the property in the first column and a brief description in the second column.

<code>presetThemeChannel</code>	Defines the preset theme channel for the container. The JSP™ defined in the channel displays the Theme->Preset Themes page.
<code>customThemeChannel</code>	Defines the custom theme channel for the container. The JSP defined in the channel displays the Theme-> Custom Theme page.

`editContainerName` Defines the edit container channel for this container. When a leaf channel defined in this container is of the type `edit_subset`, then the edit container channel is used to display a frame for the Edit page for the leaf channel.

The table and tab containers have a Content page for users to select channels to be displayed. This page is generated from the display profile `<Collection name="categories"> </Collection>` tags. The channels in the container are grouped based on the categories that are defined in the display profile.

NOTE For the template-based Desktop, the Content page is the same as the iPlanet™ Portal Server 3.0 style, where channels are grouped based on channel width. Thus, the categories property is not required for the template-based containers.

Each container can have different categories, which is the case for the sample portal. First, you define a collection of categories in the display profile. Then, for each value in that collection, you define another collection named as the value of the categories.

For example, the following defines two channels, Personal Channels and Sample Channels, with two links each.

```
<Collection name="categories">
  <String value="Personal Channels">
  <String value="Sample Channels">
</Collection>
<Collection name="Personal Channels"> <!--category one
  <String value="UserInfo"/>
  <String value="MailCheck"/>
</Collection>
<Collection name="Sample Channels"> <!--category two
  <String value="SampleJSP">
  <String value="SampleXML">
</Collection>
```

Leaf Provider Display Profile Definitions and Properties

This chapter contains information on the display profile definitions and the properties of the building-block providers that ship with Sun™ ONE Portal Server software.

This chapter contains the following sections:

- [Overview of Display Profile Provider Definitions and Properties](#)
- [Display Profile Definitions and Properties of Leaf Building-Block Providers](#)
- [Display Profile Definition for Service Providers](#)
- [Display Profile Definitions for Content Providers](#)

Overview of Display Profile Provider Definitions and Properties

The sample portal makes use of the following display profile definitions in the *portal-server-install-root/SUNWps/samples/desktop* directory:

<code>dp-org.xml</code>	Contains the display profile definitions for channels and containers.
<code>dp-providers.xml</code>	Contains the display profile definitions for providers.
<code>dp-org-final.xml</code>	Contains the display profile definitions for channels and containers and includes NetMail and NetMail Lite definitions, and the Search definitions.

<code>dp-anon.xml</code>	Contains the display profile definitions for channels and containers for the authlessanonymous and anonymous users in the default organization.
--------------------------	---

Providers have required properties, as well as general properties. This chapter describes each provider-specific property and the general properties of the providers.

By editing the provider properties in the display profile XML files, you can create customized display profile provider definitions. Any time you modify a display profile document, use the `dpadmin` command (or the Sun™ ONE Identity Server software administration console) to store it in LDAP. For information about using the `dpadmin` command, see the *Sun ONE Portal Server 6.2 Administrator's Guide*.

Display Profile Definitions and Properties of Leaf Building-Block Providers

This section describes the display profile definitions and the properties of the leaf building-block providers.

Leaf building-block providers generate their own content. They include:

- [JSPProvider](#)
- [URLScraperProvider](#)
- [XMLProvider](#)

JSPProvider

JSPProvider uses JavaServer Pages™ (JSP™). JSPProvider obtains content from one or more JSP files. A JSP file can be a static document (HTML only) or a standard JSP file with HTML and Java code. A JSP can include other JSP files. However, only the topmost JSP can be configured through the display profile. The topmost JSP files are defined through the `contentPage`, `editPage`, and `processPage` properties. See the *Sun ONE Portal Server 6.2 Developer's Guide* for more information on how JSPProvider uses these JSPs.

If you need to make other customizations, you do so in the JSP files themselves.

[Code Example 4-1](#) shows the display profile definition for JSPProvider.

Code Example 4-1 Display Profile Definition for JSPProvider

```

<Provider name="JSPProvider" class="com.sun.portal.providers.jsp.JSPProvider">
  <Properties>
    <String name="title" value="*** JSP Provider ***"/>
    <String name="description" value="*** DESCRIPTION ***"/>
    <String name="refreshTime" value="0" advanced="true"/>
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/jspchann.htm" advanced="true"/>
    </ConditionalProperties>
    <String name="helpURL" value="en/desktop/jspchann.htm" advanced="true"/>
    <String name="fontFace1" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <String name="width" value="thick"/>
    <String name="contentPage" value="samplecontent.jsp"/>
    <String name="editPage" value="sampleedit.jsp"/>
    <String name="processPage" value="sampledoedit.jsp"/>
    <Boolean name="showExceptions" value="false"/>
    <String name="editType" value="edit_subset" advanced="true"/>
    <Boolean name="isEditable" value="false" advanced="true"/>
  </Properties>
</Provider>

```

The following is a list of the properties specific to JSPProvider. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

String name="contentPage"	Specifies the JSP that is used to generate the channel content (by using the <code>getContent</code> method).
String name="editPage"	Specifies the JSP that is used to generate the Edit page content (by using the <code>getEdit</code> method).
String name="processPage"	Specifies the JSP that is used to process the results of an Edit page (by using the <code>processEdit</code> method).
Boolean name="showExceptions"	If <code>true</code> , makes JSPProvider show exceptions generated while processing the JSP as the channel output for the <code>getContent</code> and <code>getEdit</code> methods. This can be useful for developing and troubleshooting your portal.

See [“JSP Channel” on page 172](#) for an example of a channel that uses JSPProvider.

URLScrapperProvider

URLScrapperProvider takes a URL, opens a connection to the URL, and reads the contents into a buffer. The contents are then sent to the Desktop servlet, which displays it. URLScrapperProvider uses the Rewriter to construct the URL information and an XML file to describe how to display it. See the *Sun ONE Portal Server 6.2 Administrator's Guide* for more information on the Rewriter.

[Code Example 4-2 on page 82](#) shows the display profile definition for URLScrapperProvider.

Code Example 4-2 Display Profile Definition for URLScrapperProvider

```
<Provider name="URLScrapperProvider"
class="com.sun.portal.providers.urlscrapper.URLScrapperProvider">
  <Properties>
    <String name="title" value="UrlScrapper Channel"/>
    <String name="description" value="This is a test for urlscrapper"/>
    <Boolean name="isEditable" value="false" advanced="true"/>
    <String name="url" value="/desktop/ipinfo.html"/>
    <String name="urlScrapperRulesetID" value="default_ruleset"/>
    <String name="width" value="thick" advanced="true"/>
    <String name="refreshTime" value="0" advanced="true"/>
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/urlscrpr.htm" advanced="true"/>
    </ConditionalProperties>
    <String name="helpURL" value="en/desktop/urlscrpr.htm" advanced="true"/>
    <String name="fontFacel" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <Boolean name="cookiesToForwardAll" value="true"/>
    <String name="inputEncoding" value="iso-8859-1"/>
    <Collection name="cookiesToForwardList">
    </Collection>
    <Integer name="timeout" value="100"/>
  </Properties>
</Provider>
```

The following is a list of the properties specific to URLScrapperProvider. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

String name="url"	Specifies the URL to be scraped. The default value is /desktop/ipinfo.html.
String name="urlScrapperRulesetID"	Specifies the ID of the ruleset to be used by the Rewriter for rewriting content.

Boolean name="cookiesToForwardAll"	Specifies whether to forward cookies.
String name="inputEncoding"	Specifies the input encoding to be used by <code>URLScrapperProvider</code> to encode the scraped content.
Collection name="cookiesToForwardList"	Specifies the list of cookies to be forwarded by <code>URLScrapperProvider</code> if <code>cookiesToForwardAll</code> is set to <code>false</code> .
Integer name="timeout"	Specifies the timeout for which the provider should wait to fetch content before displaying the timed out message.

The `isEditable` property for `URLScrapperProvider` cannot be turned on (set to `true`) as this channel is not editable. There are no `getEdit()` and `processEdit()` methods defined for this provider. If you want edit functionality for `URLScrapperProvider`, define another provider that extends `URLScrapperProvider`. In so doing, you would need to implement the `getEdit()` and `processEdit()` methods, and also define the `editType` property.

See the *Sun ONE Portal Server 6.2 Developer's Guide* for information on extending the `URLScrapperProvider`. See [“Display Profile Common Properties for Providers” on page 127](#) for more information on the `editType` property.

See [“Sample URL Scrapper Channel” on page 186](#) for an example of a channel that uses `URLScrapperProvider`.

XMLProvider

`XMLProvider` transforms an XML document into HTML using an XSLT (XML Style Sheet Language) file. You must create the appropriate XSLT file to match the XML document type. `XMLProvider` is an extension of `URLScrapperProvider`. This provider uses the JAXP 1.1 JAR files that come with Sun™ ONE Web Server 6.0 SP1.

NOTE This guide does not discuss XML and XSL technologies. See <http://www.w3.org/TR/xslt> for more information.

[Code Example 4-3 on page 84](#) shows the display profile definition for `XMLProvider`.

Code Example 4-3 Display Profile Definition for XMLProvider

```
<Provider name="XMLProvider" class="com.sun.portal.providers.xml.XMLProvider">
  <Properties>
    <String name="title" value="*** XML Provider ***"/>
    <String name="description" value="*** DESCRIPTION ***"/>
    <String name="width" value="thick"/>
    <String name="refreshTime" value="0" advanced="true"/>
    <Boolean name="isEditable" value="false" advanced="true"/>
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="desktop/xmlchann.htm" advanced="true"/>
    </ConditionalProperties>
    <String name="helpURL" value="desktop/xmlchann.htm" advanced="true"/>
    <String name="fontFace1" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <String name="url"
value="file:///etc/opt/SUNWps/desktop/default/XMLProvider/getQuotes.xml"/>
    <String name="xslFileName" value="html_stockquote.xsl"/>
    <Integer name="timeout" value="100"/>
    <String name="inputEncoding" value="iso-8859-1"/>
    <String name="urlScrapperRulesetID" value=""/>
    <Boolean name="cookiesToForwardAll" value="true"/>
    <Collection name="cookiesToForwardList">
    </Collection>
  </Properties>
</Provider>
```

The following is a list of the properties specific to XMLProvider. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

String name="url"	Specifies the URL that XMLProvider is to transform.
String name="xslFileName"	<p>Specifies the path to the local file to be used as the XSL style sheet.</p> <p>The provider code tries to pick up the XSL file either from the XML channel directory (that is, /etc/opt/SUNWps/desktop/default/SampleXML), or if not specified here, from the XML provider directory (/etc/opt/SUNWps/desktop/default/XMLProvider/xml).</p>
String name="urlScrapperRulesetID"	Specifies the ID of the ruleset to be used by the Rewriter for rewriting content.

Boolean name="cookiesToForwardAll"	Specifies whether to forward cookies.
String name="inputEncoding"	Specifies the input encoding to be used by XMLProvider to encode the scraped content.
Collection name="cookiesToForwardList"	Specifies the list of cookies to be forwarded by URLScrapperProvider if cookiesToForwardAll is set to false.
Integer name="timeout"	Specifies the timeout for which the provider should wait to fetch content before displaying the timed out message.

The `isEditable` property for XMLProvider cannot be turned on (set to `true`) as this channel is not editable. There are no `getEdit()` and `processEdit()` methods defined for this provider. If you want edit functionality for XMLProvider, define another provider that extends XMLProvider. In so doing, you would need to implement the `getEdit()` and `processEdit()` methods, and also define the `editType` property.

See the *Sun ONE Portal Server 6.2 Developer's Guide* for more information on extending the XMLProvider. See "Display Profile Provider Required Properties," on page 98 for more information on the `editType` property.

See "[Sample XML Channel](#)" on page 187 for an example of a channel that uses XMLProvider.

Rich Site Summary (RSS) Channel

There is no separate RSS provider. Instead, the Sample RSS channel is a channel of XMLProvider. It supports RSS version 0.91, which is XML compliant, and uses an XSLT (style sheet) file to transform the RSS XML file into HTML. By using different XSL files, the same output can be changed and presented differently for different channels.

See "[Sample RSS Channel](#)" on page 184 for more information.

Display Profile Definition for Service Providers

Service providers are providers who provide a service, such as search service. The Sun ONE Portal Server software includes the following service providers:

- [SearchProvider](#)

- [DiscussionProvider](#)
- [SubscriptionsProvider](#)

SearchProvider

SearchProvider supplies the search function using the Sun ONE Portal Server software Search Engine. SearchProvider is a JSP-based provider. The resultant channel has three interfaces:

Basic search Enables users to search within the default document database or discussion database. Document and category matches are then displayed.

Advanced search Enables users to search for documents based on author, title, URL within the default document database or discussion database, discussion, and/or comment. Users can also search on the last-modified date of a document. Advanced search is a more complex user interface, and supports customization. See [Chapter 7, “Customizing the Desktop Interface”](#) for more information.

Browse Enables users to browse the category tree and search within categories.

Search results are displayed based on the `categorySearch` and `viewHits` properties. [Code Example 4-4](#) shows the display profile definition for SearchProvider.

Code Example 4-4 Display Profile Definition for SearchProvider

```
<Provider name="SearchProvider" class="com.sun.portal.providers.jsp.JSPProvider">
  <Properties>
    <String name="title" value="*** Search Provider ***"/>
    <String name="description" value="*** DESCRIPTION ***"/>
    <String name="refreshTime" value="0" advanced="true"/>
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/search.htm" advanced="true"/>
    </ConditionalProperties>
    <String name="helpURL" value="en/desktop/search.htm" advanced="true"/>
    <String name="fontFace1" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <String name="contentPage" value="searchContent.jsp"/>
    <String name="editPage" value="searchEdit.jsp"/>
    <String name="processPage" value="searchDoEdit.jsp"/>
    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="editType" value="edit_subset" advanced="true"/>
    <String name="windowPref" value="all_new"/>
    <String name="title" value="Search and Browse Channel"/>
    <String name="width" value="thick"/>
    <String name="column" value="2"/>
    <Boolean name="showExceptions" value="false"/>
    <String name="searchServer" value="" />
  </Properties>
</Provider>
```

Code Example 4-4 Display Profile Definition for SearchProvider (Continued)

```

<Integer name="viewHits" value="8"/>
<Boolean name="basicSearchDefault" value="true"/>
<String name="defaultMode" value="basic"/>
<Boolean name="categorySearch" value="false"/>
<String name="documentDisplay" value="new"/>
</Properties>
</Provider>

```

The following is a list of the properties specific to SearchProvider. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

<code>String name="contentPage"</code>	Specifies the JSP that is used to generate the channel content (by using the <code>getContent()</code> method).
<code>String name="editPage"</code>	Specifies the JSP that is used to generate the Edit page content (by using the <code>getEdit()</code> method).
<code>String name="processPage"</code>	Specifies the JSP that is used to process the results of an Edit page (by using the <code>processEdit()</code> method).
<code>Boolean name="showExceptions"</code>	If true, makes SearchProvider show exceptions generated while processing the JSP as the channel output for the <code>getContent</code> and <code>getEdit</code> methods. This can be useful for developing and troubleshooting your portal, and for debugging the Search provider.
<code>String name="searchServer"</code>	Specifies the Search server's URL.
<code>Integer name="viewHits"</code>	Specifies the number of hits that should be displayed per page. The maximum desirable number is 25. (The Edit page specifies to choose a number between 1 and 100.) This property is user editable. Edit page displays allowable values as 5, 8, 10, 16.
<code>Boolean name="basicSearchDefault"</code>	If true, specifies that the default search mode should be basic. (Users can set this to advanced if desired.)
<code>String name="defaultMode"</code>	Specifies the default search mode. Allowable values are basic, advanced, or browse.

Boolean name="categorySearch"

Specifies the category search to be displayed by default. If set to false, category matches are not displayed.

See [“Search Channel” on page 180](#) for an example of a channel that uses SearchProvider.

DiscussionProvider

The DiscussionProvider is JSPProvider based and 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. Discussion RDs require special schema. See `schema.rdm` file in the `/var/opt/SUNWps/https-psserver/portal/config/` directory.

Discussions are stored in the discussion database specified in the `dbname` property in the display profile. Search server host (`searchServer` property) and database name (`dbname` property) are advanced properties that can be configured in the display profile.

[Code Example 4-5](#) shows the display profile definition for DiscussionProvider.

Code Example 4-5 Display Profile Definition for DiscussionProvider

```
<Provider name="DiscussionProvider" class="com.sun.portal.providers.jsp.JSPProvider">
  <Properties>
    <String name="title" value="*** Discussions Provider ***"/>
    <String name="description" value="*** DESCRIPTION ***"/>
    <String name="refreshTime" value="0" advanced="true"/>
    <String name="helpURL" value="en/desktop/discussions.htm" advanced="true"/>
    <String name="fontFace1" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <String name="contentPage" value="discussionContent.jsp"/>
    <String name="editPage" value="discussionEdit.jsp"/>
    <String name="processPage" value="discussionDoEdit.jsp"/>
    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="editType" value="edit_subset" advanced="true"/>
    <Boolean name="showExceptions" value="false"/>
    <Boolean name="showErrors" value="true"/>
    <String name="width" value="thick"/>
    <String name="column" value="2"/>
    <String name="searchServer" value=""/>
    <String name="dbname" value=""/>
    <Integer name="viewHits" value="8"/>
    <String name="defaultDiscussionDisplay" value="Threaded"/>
    <String name="defaultFilter" value="Irrelevant"/>
    <String name="defaultExpansionThreshold" value="Collapse all"/>
  </Properties>
</Provider>
```

Code Example 4-5 Display Profile Definition for DiscussionProvider (Continued)

```

    <Boolean name="viewDiscussionWindow" value="false"/>
    <String name="anonymousAuthor" value="anonymous"/>
    <Boolean name="displaySearch" value="true"/>
    <Boolean name="showDescription" value="false"/>
    <String name="ratingText" value="Irrelevant,Routine,Interesting,Important,Must
Read"/>
  </Properties>
</Provider>

```

The following is a list of properties specific to DiscussionProvider. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

searchServer	Specifies the path to the search server. By default, the value is <code>portal/search</code> .
dbname	Specifies the discussion database where discussions are stored. Any valid database can be specified.
viewHits	Specifies the number of discussions to display on the main discussion page (full view).
defaultDiscussionDisplay	Determines how the comment subtree is displayed. It can be set to flat or threaded to allow the comment subtree to be displayed as flat or threaded.
defaultFilter	Specifies the filter for searching and displaying discussions and this controls display of the subtree. It can be based on ratings such as irrelevant, routine, interesting, important, or must read. By default, its value is irrelevant; so all comments rated irrelevant and above are displayed. The Must read filter will highlight the highly rated comments.
defaultExpansionThreshold	It can be set to expand all or collapse all. By default, its value is set to collapse all. If set to expand all, it will expand all the filtered comments, show description, rating menu, and allow user to post reply via links.

<code>viewDiscussionWindow</code>	A user configurable property. If set to <code>true</code> , the discussion link gets displayed on an entire page; that is, <code>JSPDynamicSingleContainer</code> is invoked. If set to <code>false</code> , the discussion gets displayed within the channel within the tab.
<code>anonymousAuthor</code>	An anonymous user can submit comments. Default author value for an anonymous user is picked from this property. Default value is <code>anonymous</code> . For example, it can be set to <code>unknown</code> author.
<code>displaySearch</code>	Enable or disable Search in discussions.
<code>showDescription</code>	Specifies whether or not to show a description of the discussion.
<code>ratingText</code>	Specifies the type of rating that can be done on a discussion. By default, discussions can be rated as irrelevant, routine, interesting, important, or must read. This property is not used in this release.

See [“Discussion Channels” on page 168](#) for example of channels that use `DiscussionProvider`.

SubscriptionsProvider

`SubscriptionsProvider` provides subscriptions service to users. The Subscriptions service enables users to create a set of profile of interest over a source of information. The source of information supported are categories, discussions, and searchable documents. The profile is updated with the latest information every time the user accesses the Subscriptions channel. The Subscriptions channel summarizes the number of hits (relevant information) that matches each profile entry the user defined for categorized document and/or discussions.

[Code Example 4-6](#) shows the display profile definition for `SubscriptionsProvider`.

Code Example 4-6 Display Profile Definition for `SubscriptionsProvider`

```
<Provider name="SubscriptionsProvider" class="com.sun.portal.providers.jsp.JSPProvider">
  <Properties>
    <String name="title" value="Subscriptions Provider"/>
    <String name="description" value="Subscriptions to Categories and discussions"/>
    <String name="refreshTime" value="600" advanced="true"/>
    <ConditionalProperties condition="locale" value="en">
```

Code Example 4-6 Display Profile Definition for SubscriptionsProvider *(Continued)*

```

        <String name="helpURL" value="en/desktop/subscriptions.htm" advanced="true"/>
    </ConditionalProperties>
    <String name="fontFace1" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <String name="contentPage" value="subscontent.jsp"/>
    <String name="editPage" value="subsedit.jsp"/>
    <String name="processPage" value="subsdoedit.jsp"/>
    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="editType" value="edit_subset" advanced="true"/>
    <Boolean name="showExceptions" value="true"/>
    <String name="width" value="thin"/>
    <String name="column" value="1"/>
    </Properties>
</Provider>

```

See “[Subscriptions Channel](#)” on page 183 for example of a channel that uses SubscriptionsProvider.

Display Profile Definitions for Content Providers

This section provides definitions and examples for the following content providers that ship with Sun ONE Portal Server software.

- [AddressBookProvider](#)
- [AppProvider](#)
- [BookmarkProvider](#)
- [CalendarProvider](#)
- [IMProvider](#)
- [LoginProvider](#)
- [MailCheckProvider](#)
- [MailProvider](#)
- [NotesProvider](#)
- [SimpleWebServiceProvider](#)
- [SimpleWebServiceConfigurableProvider](#)
- [UserInfoProvider](#)

You cannot extend content providers as their APIs are not public.

AddressBookProvider

The address book provider works with the Sun™ One Messaging Server to provide simple personal address book functionality.

[Code Example 4-7](#) shows the display profile definition for AddressBookProvider.

Code Example 4-7 Display Profile Definition for AddressBookProvider

```
<Provider name="AddressBookProvider"
class="com.sun.portal.providers.ab.AddressBookProvider" >
  <Properties>
    <String name="title" value="Address Book" />
    <String name="description" value="Sun ONE Address Book" />
    <String name="width" value="thick" />
    <String name="refreshTime" value="0" />
    <String name="productName" value="Sun ONE Portal Server" />
    <String name="fontFacel" value="Sanes-serif" />
    <String name="editType" value="edit_subset" />
    <String name="sunPortalABSortBy" value="none" />
    <String name="sunPortalABSortOrder" value="none"/>
    <String name="ssoAdapter" value="sunOneAddressBook" />
    <Integer name="maxEntries" value="30" />
    <Integer name="numEntries" value="5" />
    <Boolean name="isEditable" value="true" />
    <Boolean name="displayEntries" value="true" />
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/addressbook.html" />
    </ConditionalProperties>
    <ConditionalProperties condition="client" value="HTML">
      <Collection name="applicationHelperEdit">
        <String value="com.sun.portal.providers.ab.LDAPABApplicationHelper" />
      </Collection>
      <String name="applicationHelperURL"
value="com.sun.portal.providers.ab.LDAPABApplicationHelper" />
    </ConditionalProperties>
    <Collection name="ssoEditAttributes">
      <String name="uid" value="string|User Name:" />
      <String name="password" value="password|User Password:" />
    </Collection>
    <Collection name="dpEditAttributes">
      <String name="displayEntries" value="check|Display Entries: " />
      <String name="numEntries" value="int|Number of Entries:" />
      <String name="sunPortalABSortBy" value="select|Sort By:" />
      <String name="sunPortalABSortOrder" value="select|Sort Order:" />
    </Collection>
    <Collection name="sunPortalABSortBySelectOptions">
      <String name="NoSort" value="None" />
      <String name="Commonname" value="Full name" />
    </Collection>
    <Collection name="sunPortalABSortOrderSelectOptions">
```

Code Example 4-7 Display Profile Definition for AddressBookProvider *(Continued)*

```

        <String name="NoSort" value="None"/>
        <String name="Up" value="Ascending"/>
        <String name="Down" value="Descending"/>
    </Collection>
</Properties>
</Provider>

```

The following is a list of the properties specific to AddressBookProvider. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

String name="sunPortalABSorderBy"	Specifies the value of the entries displayed to sort by.
String name="sunPortalABSOrder"	Specifies the sort order of the entries displayed.
String name = "ssoAdapter"	Specifies the SSOAdapter configuration used by this provider/channels.
Integer name="maxEntries"	Specifies the limit for the number of address book entries to display.
Integer name="numEntries"	Specifies the number of entries to display.
Boolean name="displayEntries"	Specifies if the entries should be shown.
Collection name="applicationHelperEdit"	Specifies the mail application helpers that you can edit settings on.
String name="applicationHelperURL"	Specifies the default mail application helper.
Collection name="ssoEditAttributes"	Specifies the attributes that will appear on the 1st edit page for the provider. These are usually server settings and have nothing to do with display profile attributes.
Collection name="dpEditAttributes"	Specifies the attributes that will appear on the edit page for the application helper. These are usually display attributes and there can be multiple attributes based on number of clients, and so on.
Collection name="sunPortalABSorderBySelectOptions"	Used to generate the drop down select boxes on the edit page. This specifies None and Full name.

Collection

```
name="sunPortalABSortOrderSelectOptions"
```

Used to generate the drop down select boxes on the edit page. This specifies Ascending, Descending, and None.

See [“Address Book Channel” on page 165](#) for an example of a channel that uses AddressBookProvider.

AppProvider

AppProvider enables a user to add or remove applications from a list of applications.

[Code Example 4-6](#) shows the display profile definition for AppProvider.

Code Example 4-8 Display Profile Definition for AppProvider

```
<Provider name="AppProvider" class="com.sun.portal.providers.app.AppProvider">
  <Properties>
    <String name="title" value="App Provider Title"/>
    <String name="description" value="App Description"/>
    <Integer name="timeout" value="100"/>
    <String name="refreshTime" value="0" advanced="true"/>
    <Boolean name="isEditable" value="true" advanced="true"/>>
    <String name="editType" value="edit_subset" advanced="true"/>
    <String name="width" value="thin" advanced="true"/>>
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/myapps.htm" advanced="true"/>
    </ConditionalProperties>
    <String name="helpURL" value="en/desktop/myapps.htm" advanced="true"/>
    <String name="fontFacel" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>>
    <String name="windowPref" value="all_new"/>
    <Collection name="targets">
      <String value="NetFile Lite|NetFileLiteUIServlet?func=initOptions"/>
    </Collection>
    <Collection name="userApps">
      <String value="NetFile Lite"/>
    </Collection>
    <String name="description" value="AppProvider Channel Sample Implementation"/>
    <String name="title" value="AppProvider"/>
  </Properties>
</Provider>
```

The following is a list of the properties specific to AppProvider. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

String name="windowPref"	Specifies how to launch a link. The possible values are: <ul style="list-style-type: none"> • <code>all_new</code> (New window is opened for every link) • <code>one_new</code> (All links open on the same new window) • <code>same</code> (Desktop window)
Collection name="targets"	Specifies the list of application links in <i>name</i> URL format, where <i>name</i> should match should match the entry in the <code>userApps</code> collection.
Collection name="userApps"	Specifies the list of applications that appear in the applications channel.

See [“Applications List Channel” on page 166](#) for an example of a channel that uses AppProvider.

BookmarkProvider

BookmarkProvider enables a user to add or remove URLs from a list of bookmarks.

[Code Example 4-9 on page 95](#) shows the display profile definition for the BookmarkProvider.

Code Example 4-9 Display Profile Definition for BookmarkProvider

```
<Provider name="BookmarkProvider"
class="com.sun.portal.providers.bookmark.BookmarkProvider">
  <Properties>
    <String name="title" value="Bookmark Provider"/>
    <String name="windowPref" value="all_new"/>
    <String name="width" value="thin" advanced="true"/>
    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="editType" value="edit_subset" advanced="true"/>
    <String name="description" value="Bookmark Channel Provider Sample
Implementation"/>
    <String name="refreshTime" value="0" advanced="true"/>
```

Code Example 4-9 Display Profile Definition for BookmarkProvider (*Continued*)

```

<ConditionalProperties condition="locale" value="en">
  <String name="helpURL" value="en/desktop/bkmark.htm" advanced="true"/>
</ConditionalProperties>
<String name="helpURL" value="en/desktop/bkmark.htm" advanced="true"/>
<String name="fontFace1" value="Sans-serif"/>
<String name="productName" value="Sun ONE Portal Server"/>
<Collection name="targets">
  <String value="Sun ONE home page|http://www.iplanet.com"/>
</Collection>
</Properties>
</Provider>

```

The following is a list of the properties specific to BookmarkProvider. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

String name="windowPref"	Specifies how to launch a link. The possible values are: <ul style="list-style-type: none"> • all_new (New window is opened for every link) • one_new (All links open on the same new window) • same (Desktop window)
Collection name="targets"	Specifies the list of bookmarks that is shown in the channel in the following format: <p><i>BookmarkName</i> URL</p>

See [“Bookmark Channel” on page 166](#) for an example of a channel that uses BookmarkProvider.

CalendarProvider

The calendar provider works with the Sun™ One Calender Server so that you can view tasks and events and launch Calendar Express without having to sign in.

[Code Example 4-10](#) shows the display profile definition for the CalendarProvider.

Code Example 4-10 Display Profile Definition for CalendarProvider

```

<Provider name="CalendarProvider"
class="com.sun.portal.providers.calendar.CalendarProvider">
  <Properties>
    <String name="title" value="Calendar" />
    <String name="description" value="Sun ONE Calendar" />
    <String name="width" value="thick" />
    <String name="refreshTime" value="0" />
    <String name="editType" value="edit_subset" />
    <String name="productName" value="Sun ONE Portal Server" />
    <String name="fontFacel" value="Sans-serif" />
    <String name="view" value="dayWeekView" />
    <String name="calendar" value="" />
    <String name="ssoAdapter" value="sunOneCalendar" />
    <Boolean name="isEditable" value="true" />
    <Boolean name="loadSubscribedCalendars" value="true" />
    <Boolean name="disableTaskEventURLs" value="false" />
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/calendar.html" />
    </ConditionalProperties>
    <ConditionalProperties condition="client" value="HTML">
      <Collection name="applicationHelperEdit">
        <String value="com.sun.portal.providers.calendar.CalendarExpressHelper" />
      </Collection>
      <String name="applicationHelperURL"
value="com.sun.portal.providers.calendar.CalendarExpressHelper" />
    </ConditionalProperties>
    <Collection name="ssoEditAttributes">
      <String name="host" value="string|Server Name:" />
      <String name="port" value="string|Server Port:" />
      <String name="uid" value="string|User Name:" />
      <String name="password" value="password|User Password:" />
    </Collection>
    <Collection name="dpEditAttributes">
      <String name="view" value="select|Calendar View:" />
      <String name="calendar" value="select|Calendar:" />
    </Collection>
    <Collection name="viewSelectOptions">
      <String name="dayView" value="Day" />
      <String name="dayWeekView" value="Day and Week" />
      <String name="weekView" value="Week" />
      <String name="monthView" value="Month" />
      <String name="none" value="None" />
    </Collection>
    <Collection name="calendarSelectOptions">
    </Collection>
  </Properties>
</Provider>

```

The following is a list of the properties specific to CalendarProvider. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

<code>String name="view"</code>	Specifies the view (day, week, or month) used.
<code>String name="calendar"</code>	Specifies the calendar to display.
<code>String name="ssoAdapter"</code>	Specifies the ssoAdapter configuration to use.
<code>Boolean name="loadSubscribedCalendars"</code>	If set to <code>true</code> , it will try to load all of the subscribed calendars and display them.
<code>Boolean name="disableTaskEventURLs"</code>	If set to <code>true</code> , it will not display links for tasks and events.
<code>Collection name="calendarSelectOptions"</code>	Specifies a list of all subscribed calendars.
<code>Collection name="applicationHelperEdit"</code>	Specifies the mail application helpers that you can edit settings on
<code>String name="applicationHelperURL"</code>	Specifies the default mail application helper
<code>Collection name="ssoEditAttributes"</code>	Specifies the attributes that will appear on the 1st edit page for the provider. These are usually server settings and have nothing to do with display attributes
<code>Collection name="dpEditAttributes"</code>	Specifies the attributes that will appear on the edit page for the application helper. These are usually display attributes and there can be multiple attributes based on number of clients, etc.
<code>Collection name="viewSelectOptions"</code>	Specifies the different Calendar views displayed in the Calendar edit page.

See [“Calendar Channel” on page 167](#) for an example of a channel that uses CalendarProvider.

IMProvider

The IMProvider includes:

- Information needed to help the user decide whether to launch the IM client.
- The ability to launch the IM client using single-sign-on.

The information is gathered by accessing the Instant Messaging server through the use of the Instant Messaging APIs.

Code Example 4-11 shows the display profile definition for the IMProvider.

Code Example 4-11 Display Profile Definition for IMProvider

```
<Provider name="IMProvider" class="com.sun.im.portal.provider.IMProvider">
  <Properties>
    <String name="title" value="Instant Messaging"/>
    <String name="description" value="Desktop access to Instant Messaging contact
status"/>
    <String name="refreshTime" value="0" advanced="true"/>
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/imchan.htm" advanced="true" />
    </ConditionalProperties>
    <String name="fontFace1" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <String name="width" value="thin"/>
    <String name="contentPage" value="IMContent.jsp"/>
    <String name="editPage" value="IMEdit.jsp"/>
    <String name="processPage" value="NOTUSED"/>
    <Boolean name="showExceptions" value="false"/>
    <String name="editType" value="edit_subset" advanced="true"/>
    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="server" value=""/>
    <String name="port" value=""/>
    <String name="mux" value=""/>
    <String name="muxport" value=""/>
    <String name="codebase" value=""/>
    <String name="netletRule" value="IM"/>
    <String name="clientRunMode" value="plugin"/>
    <String name="authMethod" value="idsvr"/>
    <String name="authUsernameAttr" value="uid"/>
    <String name="username" value=""/>
    <String name="password" value=""/>
    <String name="contactGroup" value="My Contacts"/>
  </Properties>
</Provider>
```

The following is a list of the properties specific to IMProvider. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

server	Specifies the name of the instant messaging server to use.
port	Specifies the port on which the instant messaging server listens.

<code>mux</code>	Specifies the name of instant messaging multiplexor to use (used by IM client.)
<code>muxport</code>	Specifies the port on which the instant messaging server listens.
<code>codebase</code>	Specifies where to find the instant messaging client.
<code>netletRule</code>	Specifies where to find the instant messaging client when using the netlet. By default, the value is <code>IM</code> .
<code>clientRunMode</code>	Specifies how the Instant Messaging server client must be run. The client can be run as either a <code>plugin</code> or <code>jnlp</code> . By default, the value is <code>plugin</code> .
<code>authMethod</code>	Specifies the authentication method. Clients can authenticate either via <code>idsvr</code> (for Sun ONE Identity Server) or <code>ldap</code> . By default, the value is <code>idsvr</code> .
<code>authUsernameAttr</code>	Specifies the LDAP attribute where instant messaging username is found. By default, the value is <code>uid</code> .
<code>username</code>	Specifies the username for LDAP authentication. This is not applicable if <code>authMethod</code> is set to <code>idsvr</code> .
<code>password</code>	Specifies the password for LDAP authentication. This is not applicable if <code>authMethod</code> is set to <code>idsvr</code> .
<code>contactGroup</code>	Specifies the contact group to display, or blank for all.

See [“Instant Messaging Channel” on page 171](#) for an example of a channel that uses `IMProvider`.

LoginProvider

`LoginProvider` enables the Login channel to show up in the anonymous user’s Desktop. You can configure `LoginProvider` to enable users to log in and out using the Login channel. The system administrator can select one out of the three methods to enable users to log in: LDAP, Membership, or Unix.

For the sample portal, if you type the following URL in a browser, you see the authlessanonymous user’s Desktop, which has the login channel.

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

By default, LoginProvider uses Membership authentication. No additional setup is required to use this channel. From the authlessanonymous user page, valid users can use the login channel, and new users can register using the Sign me up link in the channel. You can change the authentication module for the login channel.

Code Example 4-12 shows the display profile definition for LoginProvider.

Code Example 4-12 Display Profile Definition for LoginProvider

```
<Provider name="login" class="com.sun.portal.providers.login.LoginProvider">
  <Properties>
    <String name="title" value="Login"/>
    <String name="description" value="Login Information"/>
    <String name="refreshTime" value="0" advanced="true"/>
    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="editType" value="edit_subset" advanced="true"/>
    <String name="width" value="thin"/>
    <String name="fontFace1" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <Boolean name="persistentCookie" value="false"/>
    <Boolean name="federationEnabled" value="false"/>
    <String name="preLoginURL" value="" />
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/login.htm" advanced="true"/>
    </ConditionalProperties>
    <String name="helpURL" value="en/desktop/login.htm" advanced="true" />
  </Properties>
</Provider>
```

The following properties are specific to the LoginProvider. This two column table lists the property in the first (left) column and a brief description in the second (right) column.

Boolean name="persistentCookie"	Specifies if a persistent cookie is used to remember the user ID and password.
Boolean name="federationEnabled"	If set to true, the <code>libertyLogin.Template</code> is inserted.
String name="preLoginURL"	The value specified in the channel. This property is typically of the form: <pre>http://www.siroe.con:80/amserver/preLogin? metaAlias=www.siroe.com&goto=http://www.si ro.e.com:80/portal/dt</pre>

See [“Login Channel” on page 172](#) for an example of a channel that uses LoginProvider.

LotusNotesAddressBookProvider

The address book provider works with the Lotus Notes Server to provide simple personal address book functionality.

[Code Example 4-13](#) shows the display profile definition for LotusNotesAddressBookProvider.

Code Example 4-13 Display Profile Definition for LotusNotesAddressBookProvider

```
<Provider name="LotusNotesAddressBookProvider"
class="com.sun.portal.providers.ab.AddressBookProvider" >
  <Properties>
    <String name="title" value="Lotus Notes Address Book" />
    <String name="description" value="Lotus Notes Address Book" />
    <String name="width" value="thick" />
    <String name="refreshTime" value="0" />
    <String name="productName" value="Sun ONE Portal Server" />
    <String name="fontFacel" value="Sanes-serif" />
    <String name="editType" value="edit_subset" />
    <String name="sunPortalABSortBy" value="none" />
    <String name="sunPortalABSortOrder" value="none"/>
    <String name="ssoAdapter" value="notesAddressBook" />
    <Integer name="maxEntries" value="30" />
    <Integer name="numEntries" value="5" />
    <Boolean name="isEditable" value="true" />
    <Boolean name="displayEntries" value="true" />
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/addressbook.html" />
    </ConditionalProperties>
    <ConditionalProperties condition="client" value="HTML">
      <Collection name="applicationHelperEdit">
        <String value="com.sun.portal.providers.ab.NotesABHelper" />
      </Collection>
      <String name="applicationHelperURL"
value="com.sun.portal.providers.ab.NotesABHelper" />
    </ConditionalProperties>
    <Collection name="ssoEditAttributes">
      <String name="uid" value="string|User Name:"/>
      <String name="password" value="password|User Password:"/>
    </Collection>
    <Collection name="dpEditAttributes">
      <String name="displayEntries" value="check|Display Entries: "/>
      <String name="numEntries" value="int|Number of Entries:"/>
      <String name="sunPortalABSortBy" value="select|Sort By:"/>
      <String name="sunPortalABSortOrder" value="select|Sort Order:"/>
    </Collection>
    <Collection name="sunPortalABSortBySelectOptions">
      <String name="NoSort" value="None"/>
    </Collection>
  </Properties>
</Provider>
```

Code Example 4-13 Display Profile Definition for LotusNotesAddressBookProvider (Continued)

```

        <String name="Commonname" value="Full name"/>
    </Collection>
    <Collection name="sunPortalABSortOrderSelectOptions">
        <String name="NoSort" value="None"/>
        <String name="Up" value="Ascending"/>
        <String name="Down" value="Descending"/>
    </Collection>
</Properties>
</Provider>

```

The following properties are specific to LotusNotesAddressBookProvider. This two column table lists the property in the first (left) column and a brief description in the second (right) column.

String name="sunPortalABSortBy"	Specifies the value of the entries displayed to sort by.
String name="sunPortalABSortOrder"	Specifies the sort order of the entries displayed.
String name = "ssoAdapter"	Specifies the SSOAdapter configuration used by this provider/channels.
Integer name="maxEntries"	Specifies the limit of address book entries to display.
Integer name="numEntries"	Specified the number of entries to display.
Boolean name="displayEntries"	Specifies if the entries should be shown.
Collection name="applicationHelperEdit"	Specifies the mail application helpers that you can edit settings on.
String name="applicationHelperURL"	Specifies the default mail application helper.
Collection name="ssoEditAttributes"	Specifies the attributes that will appear on the 1st edit page for the provider. These are usually server settings and have nothing to do with display attributes.
Collection name="dpEditAttributes"	Specifies the attributes that will appear on the edit page for the application helper. These are usually display attributes and there can be multiple attributes based on number of clients, etc.

Collection
name="sunPortalABSorderBySelectOptions"

Used to generate the drop down select boxes on the edit page. This specifies None and Full name.

Collection
name="sunPortalABSorderBySelectOptions"
s"

Used to generate the drop down select boxes on the edit page. This specifies Ascending, Descending, and None.

See [“Lotus Notes Address Book Channel” on page 173](#) for an example of a channel that uses LotusNotesAddressBookProvider.

LotusNotesCalendarProvider

The calendar provider works with the Lotus Notes Server so that you can view tasks and events and launch the web application without having to sign in.

[Code Example 4-14](#) shows the display profile definition for LotusNotesCalendarProvider.

Code Example 4-14 Display Profile Definition for LotusNotesCalendarProvider

```
<Provider name="LotusNotesCalendarProvider"
class="com.sun.portal.providers.calendar.CalendarProvider">
  <Properties>
    <String name="title" value="Lotus Notes Calendar" />
    <String name="description" value="Lotus Notes Calendar" />
    <String name="width" value="thick" />
    <String name="refreshTime" value="0" />
    <String name="editType" value="edit_subset" />
    <String name="productName" value="Sun ONE Portal Server" />
    <String name="fontFace1" value="Sans-serif" />
    <String name="view" value="dayWeekView" />
    <String name="calendar" value="" />
    <String name="ssoAdapter" value="notesCalendar" />
    <Boolean name="isEditable" value="true" />
    <Boolean name="loadSubscribedCalendars" value="false" />
    <Boolean name="disableTaskEventURLs" value="false" />
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/calendar.html" />
    </ConditionalProperties>
    <ConditionalProperties condition="client" value="HTML">
      <Collection name="applicationHelperEdit">
        <String value="com.sun.portal.providers.calendar.NotesCalendarHelper" />
      </Collection>
      <String name="applicationHelperURL"
value="com.sun.portal.providers.calendar.NotesCalendarHelper" />
    </ConditionalProperties>
    <Collection name="ssoEditAttributes">
      <String name="host" value="string|Server Name:" />
    </Collection>
  </Properties>
</Provider>
```

Code Example 4-14 Display Profile Definition for LotusNotesCalendarProvider (Continued)

```

        <String name="port" value="string|Server Port:"/>
        <String name="uid" value="string|User Name:"/>
        <String name="password" value="password|User Password:"/>
    </Collection>
    <Collection name="dpEditAttributes">
        <String name="view" value="select|Calendar View:"/>
    </Collection>
    <Collection name="viewSelectOptions">
        <String name="dayView" value="Day"/>
        <String name="dayWeekView" value="Day and Week"/>
        <String name="weekView" value="Week"/>
        <String name="monthView" value="Month"/>
        <String name="none" value="None"/>
    </Collection>
    <Collection name="calendarSelectOptions">
    </Collection>
</Properties>
</Provider>

```

The following properties are specific to LotusNotesCalendarProvider. This two column table lists the property in the first (left) column and a brief description in the second (right) column.

String name="view"	Specifies the view (day, week, or month) used.
String name="calendar"	Specifies the calendar to display.
String name="ssoAdapter"	Specifies the ssoAdapter configuration to use.
Boolean name="loadSubscribedCalendars"	If set to <code>true</code> , it will try to load all of the subscribed calendars and display them.
Boolean name="disableTaskEventURLs"	If set to <code>true</code> , it will not display links for tasks and events.
Collection name="calendarSelectOptions"	Specifies a list of all subscribed calendars.
Collection name="applicationHelperEdit"	Specifies the mail application helpers that you can edit settings on.
String name="applicationHelperURL"	Specifies the default mail application helper.
Collection name="ssoEditAttributes"	Specifies the attributes that will appear on the 1st edit page for the provider. These are usually server settings and have nothing to do with display attributes.

Collection name="dpEditAttributes"	Specifies the attributes that will appear on the edit page for the application helper. These are usually display attributes and there can be multiple attributes based on number of clients, etc.
Collection name="viewSelectOptions"	Specifies the different Calendar views displayed in the Calendar edit page.

See [“Lotus Notes Calendar Channel” on page 174](#) for an example of a channel that uses LotusNotesCalendarProvider.

LotusNotesMailProvider

The mail provider works with the Lotus Notes Server to provide simple mail functionality.

[Code Example 4-15](#) shows the display profile definition for LotusNotesMailProvider.

Code Example 4-15 Display Profile Definition for LotusNotesMailProvider

```

<Provider name="LotusNotesMailProvider" class="com.sun.portal.providers.mail.MailProvider">
  <Properties>
    <String name="title" value="Lotus Notes Mail" />
    <String name="description" value="Lotus Notes Mail" />
    <String name="width" value="thick" />
    <String name="refreshTime" value="0" />
    <String name="sortOrder" value="top" />
    <String name="editType" value="edit_subset" />
    <String name="productName" value="Sun ONE Portal Server" />
    <String name="fontFacel" value="Sans-serif" />
    <String name="ssoAdapter" value="notesMail" />
    <Integer name="numberHeaders" value="10" />
    <Boolean name="displayHeaders" value="true" />
    <Boolean name="sentFolderCopy" value="false" />
    <Boolean name="isEditable" value="true" />
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/mailhelp.html" />
    </ConditionalProperties>
    <ConditionalProperties condition="client" value="HTML">
      <Collection name="applicationHelperEdit">
        <String value="com.sun.portal.providers.mail.NotesMailHelper" />
      </Collection>
      <String name="applicationHelperURL"
value="com.sun.portal.providers.mail.NotesMailHelper" />
    </ConditionalProperties>
    <Collection name="ssoEditAttributes">

```

Code Example 4-15 Display Profile Definition for LotusNotesMailProvider (Continued)

```

    <String name="host" value="string|Server Name:"/>
    <String name="port" value="string|IMAP Server Port:"/>
    <String name="uid" value="string|User Name:"/>
    <String name="password" value="password|User Password:"/>
    <String name="smtpServer" value="string|SMTP Server Name:"/>
    <String name="clientPort" value="string|Client Port:"/>
  </Collection>
  <Collection name="dpEditAttributes">
    <String name="numberHeaders" value="int|Number of Headers:"/>
    <String name="displayHeaders" value="check|Display Headers:"/>
    <String name="sortOrder" value="select|Sort Order:"/>
    <String name="sentFolderCopy" value="check|When sending a message place a copy
in Sent Folder: "/>
  </Collection>
  <Collection name="sortOrderSelectOptions">
    <String name="top" value="Most recent at top"/>
    <String name="bottom" value="Most recent at bottom"/>
  </Collection>
</Properties>
</Provider>

```

The following properties are specific to LotusNotesMailProvider. This two column table lists the property in the first (left) column and a brief description in the second (right) column.

String name = "sortOrder"	Specifies the sort order for the messages currently displayed.
String name = "ssoAdapter"	Specifies the SSOAdapter configuration used by this provider/channels.
Integer name="numberHeaders"	Specifies the limit of message headers to display. Hard limit is 30.
Boolean name="displayHeaders"	Specifies if the headers should be shown.
Boolean name="sentFolderCopy"	Specifies if sent messages should be copied to the Sent Folder (used by MA).
Collection name="applicationHelperEdit"	Specifies the mail application helpers that you can edit settings on.
String name="applicationHelperURL"	Specifies the default mail application helper.

<p>Collection name="ssoEditAttributes"</p>	<p>Specifies the attributes that will appear on the 1st edit page for the provider. These are usually server settings and have nothing to do with display attributes.</p>
<p>Collection name="dpEditAttributes"</p>	<p>Specifies the attributes that will appear on the edit page for the application helper. These are usually display attributes and there can be multiple attributes based on number of clients, etc.</p>
<p>Collection name="sortOrderSelectOptions"</p>	<p>Specifies the different mail sort order options (recent at top or bottom).</p>

See [“Lotus Notes Mail Channel” on page 174](#) for an example of a channel that uses LotusNotesMailProvider.

MailCheckProvider

MailCheckProvider gives information about a user’s mail status.

[Code Example 4-16](#) shows the display profile definition for MailCheckProvider.

Code Example 4-16 Display Profile Definition for MailCheckProvider

```

<Provider name="MailCheckProvider"
class="com.sun.portal.providers.mailcheck.MailCheckProvider">
  <Properties>
    <String name="title" value="MailCheck Provider"/>
    <Boolean name="isEditable" value="false"/>
    <String name="editType" value="edit_subset" advanced="true"/>
    <String name="width" value="thin" advanced="true"/>
    <String name="description" value="MailCheck Provider Sample Implementation"/>
    <String name="refreshTime" value="0" advanced="true"/>
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/mlchck.htm" advanced="true"/>
    </ConditionalProperties>
    <String name="helpURL" value="en/desktop/mlchck.htm" advanced="true"/>
    <String name="fontFacel" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <String name="Title" value="Mailcheck Provider"/>
    <String name="IMAPServerName" value=""/>
    <String name="IMAPUserID" value=""/>
    <String name="IMAPPassword" value="" advanced="true"/>
    <Boolean name="defaultConfigParameters" value="true"/>
    <Collection name="targets">
      <String value="NetMail Lite|NetMailServlet?nsid=newHTMLSession"/>
      <String value="NetMail|NetMailServlet?nsid=newAppletSession"/>
    </Collection>
  </Properties>
</Provider>

```

Code Example 4-16 Display Profile Definition for MailCheckProvider (Continued)

```

    </Collection>
    <Collection name="userApps">
      <String value="NetMail Lite"/>
      <String value="NetMail"/>
    </Collection>
  </Properties>
</Provider>

```

The following is a list of the properties specific to MailCheckProvider. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

String name="IMAPServerName"	Specifies the IMAP server name.
String name="IMAPUserID"	Specifies the IMAP user name.
String name="IMAPPassword"	Specifies the IMAP password.
Boolean name="defaultConfigParameters"	MailCheckProvider uses the <code>defaultConfigParameters</code> property to read a user's mail settings. If set to <code>true</code> , MailCheckProvider reads the mail settings (server name, username, and password) from the NetMail service definition. If set to <code>false</code> , MailCheckProvider reads the mail settings from the properties defined in its display profile provider properties (<code>IMAPServerName</code> , <code>IMAPUserId</code> , and <code>IMAPPassword</code>). When <code>defaultConfigParameters</code> is set to <code>true</code> , the property <code>isEditable</code> is set to <code>false</code> implying that the Edit button is not available. If you change <code>defaultConfigParameters</code> to <code>false</code> , change <code>isEditable</code> to <code>true</code> for the Edit button to appear.
Collection name="targets"	Specifies the list of application links in <code>name URL</code> format, where <code>name</code> should match should match the entry in the <code>userApps</code> collection.
Collection name="userApps"	Specifies the list of mail applications that appear in the applications channel.

See [“Mail Check Channel” on page 176](#) for an example of a channel that uses MailCheckProvider.

MailProvider

The mail provider works with the Sun™ One Messaging Server software to provide simple mail functionality and single sign-on services.

[Code Example 4-17](#) shows the display profile definition for MailProvider.

Code Example 4-17 Display Profile Definition for MailProvider

```
<Provider name="MailProvider" class="com.sun.portal.providers.mail.MailProvider">
  <Properties>
    <String name="title" value="Mail" />
    <String name="description" value="Sun ONE Mail" />
    <String name="width" value="thick" />
    <String name="refreshTime" value="0" />
    <String name="sortOrder" value="top" />
    <String name="editType" value="edit_subset" />
    <String name="productName" value="Sun ONE Portal Server" />
    <String name="fontFacel" value="Sans-serif" />
    <String name="ssoAdapter" value="sunOneMail" />
    <Integer name="numberHeaders" value="10" />
    <Boolean name="displayHeaders" value="true" />
    <Boolean name="sentFolderCopy" value="false" />
    <Boolean name="isEditable" value="true" />
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/mailhelp.html" />
    </ConditionalProperties>
    <ConditionalProperties condition="client" value="HTML">
      <Collection name="applicationHelperEdit">
        <String value="com.sun.portal.providers.mail.MessengerExpressHelper" />
      </Collection>
      <String name="applicationHelperURL"
value="com.sun.portal.providers.mail.MessengerExpressHelper" />
    </ConditionalProperties>
    <Collection name="ssoEditAttributes">
      <String name="host" value="string|Server Name:" />
      <String name="port" value="string|IMAP Server Port:" />
      <String name="uid" value="string|User Name:" />
      <String name="password" value="password|User Password:" />
      <String name="smtpServer" value="string|SMTP Server Name:" />
      <String name="clientPort" value="string|Client Port:" />
    </Collection>
    <Collection name="dpEditAttributes">
      <String name="numberHeaders" value="int|Number of Headers:" />
      <String name="displayHeaders" value="check|Display Headers:" />
      <String name="sortOrder" value="select|Sort Order:" />
      <String name="sentFolderCopy" value="check|When sending a message place a copy
in Sent Folder: "/>
    </Collection>
```

Code Example 4-17 Display Profile Definition for MailProvider *(Continued)*

```

    <Collection name="sortOrderSelectOptions">
      <String name="top" value="Most recent at top"/>
      <String name="bottom" value="Most recent at bottom"/>
    </Collection>
  </Properties>
</Provider>

```

The following is a list of the properties specific to MailProvider. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

String name = "sortOrder"	Specifies the sort order for the messages currently displayed.
String name = "ssoAdapter"	Specifies the SSOAdapter configuration used by this provider/channels.
Integer name="numberHeaders"	Specifies the limit of message headers to display. Hard limit is 30.
Boolean name="displayHeaders"	Specifies if the headers should be shown.
Boolean name="sentFolderCopy"	Specifies if sent messages should be copied to the Sent Folder (used by MA).
Collection name="applicationHelperEdit"	Specifies the mail application helpers that you can edit settings on.
String name="applicationHelperURL"	Specifies the default mail application helper.
Collection name="ssoEditAttributes"	Specifies the attributes that will appear on the first edit page for the provider. These are usually server settings and have nothing to do with display attributes.
Collection name="dpEditAttributes"	Specifies the attributes that will appear on the edit page for the application helper. These are usually display attributes and there can be multiple attributes based on number of clients, etc.
Collection name="sortOrderSelectOptions"	Specifies the different mail sort order options (recent at top or bottom).

See [“Mail Channel” on page 175](#) for an example of a channel that uses MailProvider.

MSExchangeAddressBookProvider

The address book provider works with the Microsoft Exchange Server to provide simple personal address book functionality.

[Code Example 4-18](#) shows the display profile definition for MSExchangeAddressBookProvider.

Code Example 4-18 Display Profile Definition for MSExchangeAddressBookProvider

```
<Provider name="MSExchangeAddressBookProvider"
class="com.sun.portal.providers.ab.AddressBookProvider">
  <Properties>
    <String name="title" value="Exchange Address Book"/>
    <String name="description" value="Exchange Address Book"/>
    <String name="width" value="thick"/>
    <String name="refreshTime" value="0"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <String name="fontFacel" value="Sanes-serif"/>
    <String name="editType" value="edit_subset"/>
    <String name="sunPortalABSortBy" value="none" />
    <String name="sunPortalABSortOrder" value="none"/>
    <String name="ssoAdapter" value="exchangeAddressBook"/>
    <Integer name="maxEntries" value="30"/>
    <Integer name="numEntries" value="5"/>
    <Boolean name="isEditable" value="true"/>
    <Boolean name="displayEntries" value="true"/>
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/addressbook.html"/>
    </ConditionalProperties>
    <ConditionalProperties condition="client" value="HTML">
      <Collection name="applicationHelperEdit">
        <String value="com.sun.portal.providers.ab.ExchangeABHelper"/>
      </Collection>
      <String name="applicationHelperURL"
value="com.sun.portal.providers.ab.ExchangeABHelper"/>
    </ConditionalProperties>
    <Collection name="ssoEditAttributes">
      <String name="uid" value="string|User Name:"/>
      <String name="password" value="password|User Password:"/>
      <String name="winDomain" value="string|Domain:"/>
    </Collection>
    <Collection name="dpEditAttributes">
      <String name="displayEntries" value="check|Display Entries:" />
      <String name="numEntries" value="int|Number of Entries:"/>
      <String name="sunPortalABSortBy" value="select|Sort By:"/>
      <String name="sunPortalABSortOrder" value="select|Sort Order:"/>
    </Collection>
    <Collection name="sunPortalABSortBySelectOptions">
```

Code Example 4-18 Display Profile Definition for MExchangeAddressBookProvider (Continued)

```

        <String name="NoSort" value="None"/>
        <String name="Commonname" value="Full name"/>
    </Collection>
    <Collection name="sunPortalABSortOrderSelectOptions">
        <String name="NoSort" value="None"/>
        <String name="Up" value="Ascending"/>
        <String name="Down" value="Descending"/>
    </Collection>
    </Properties>
</Provider>

```

The following properties are specific to MExchangeAddressBookProvider. This two column table lists the property in the first (left) column and a brief description in the second (right) column.

String name="sunPortalABSortBy"	Specifies the value of the entries displayed to sort by.
String name="sunPortalABSortOrder"	Specifies the sort order of the entries displayed.
String name = "ssoAdapter"	Specifies the SSOAdapter configuration used by this provider/channels.
Integer name="maxEntries"	Specifies the limit of address book entries to display.
Integer name="numEntries"	Specified the number of entries to display.
Boolean name="displayEntries"	Specifies if the entries should be shown.
Collection name="applicationHelperEdit"	Specifies the mail application helpers that you can edit settings on.
String name="applicationHelperURL"	Specifies the default mail application helper.
Collection name="ssoEditAttributes"	Specifies the attributes that will appear on the 1st edit page for the provider. These are usually server settings and have nothing to do with display attributes.
Collection name="dpEditAttributes"	Specifies the attributes that will appear on the edit page for the application helper. These are usually display attributes and there can be multiple attributes based on number of clients, etc.

<p>Collection name="sunPortalABSorderBySelectOptions"</p>	<p>Used to generate the drop down select boxes on the edit page. This specifies None and Full name.</p>
<p>Collection name="sunPortalABSorderBySelectOptions"</p>	<p>Used to generate the drop down select boxes on the edit page. This specifies Ascending, Descending, and None.</p>

See [“MS Exchange Address Book Channel” on page 177](#) for an example of a channel that uses MExchangeAddressBookProvider.

MExchangeCalendarProvider

The calendar provider works with the Microsoft Exchange Server so that you can view tasks and events and launch Exchanges web application.

[Code Example 4-19](#) shows the display profile definition for MExchangeCalendarProvider.

Code Example 4-19 Display Profile Definition for MExchangeCalendarProvider

```

<Provider name="MExchangeCalendarProvider"
class="com.sun.portal.providers.calendar.CalendarProvider">
  <Properties>
    <String name="title" value="Exchange Calendar"/>
    <String name="description" value="Exchange Calendar"/>
    <String name="width" value="thick"/>
    <String name="refreshTime" value="0"/>
    <String name="editType" value="edit_subset"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <String name="fontFace1" value="Sans-serif"/>
    <String name="view" value="dayWeekView"/>
    <String name="calendar" value=""/>
    <String name="ssoAdapter" value="exchangeCalendar"/>
    <Boolean name="isEditable" value="true"/>
    <Boolean name="loadSubscribedCalendars" value="false"/>
    <Boolean name="disableTaskEventURLs" value="false"/>
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/calendar.html"/>
    </ConditionalProperties>
    <ConditionalProperties condition="client" value="HTML">
      <Collection name="applicationHelperEdit">
        <String value="com.sun.portal.providers.calendar.ExchangeCalendarHelper"/>
      </Collection>
      <String name="applicationHelperURL"
value="com.sun.portal.providers.calendar.ExchangeCalendarHelper"/>
    </ConditionalProperties>
    <Collection name="ssoEditAttributes">
      <String name="host" value="string|Server Name:"/>

```

Code Example 4-19 Display Profile Definition for MExchangeCalendarProvider (Continued)

```

        <String name="port" value="string|Server Port:"/>
        <String name="uid" value="string|User Name:"/>
        <String name="password" value="password|User Password:"/>
        <String name="winDomain" value="string|Domain:"/>
    </Collection>
    <Collection name="dpEditAttributes">
        <String name="view" value="select|Calendar View:"/>
    </Collection>
    <Collection name="viewSelectOptions">
        <String name="dayView" value="Day"/>
        <String name="dayWeekView" value="Day and Week"/>
        <String name="weekView" value="Week"/>
        <String name="monthView" value="Month"/>
        <String name="none" value="None"/>
    </Collection>
    <Collection name="calendarSelectOptions">
    </Collection>
</Properties>
</Provider>

```

The following properties are specific to MExchangeCalendarProvider. This two column table lists the property in the first (left) column and a brief description in the second (right) column.

String name="view"	Specifies the view (day, week, or month) used.
String name="calendar"	Specifies the calendar to display.
String name="ssoAdapter"	Specifies the ssoAdapter configuration to use.
Boolean name="loadSubscribedCalendars"	If set to true, it will try to load all of the subscribed calendars and display them.
Boolean name="disableTaskEventURLs"	If set to true, it will not display links for tasks and events.
Collection name="calendarSelectOptions"	Specifies a list of all subscribed calendars.
Collection name="applicationHelperEdit"	Specifies the mail application helpers that you can edit settings on.
String name="applicationHelperURL"	Specifies the default mail application helper.

Collection name="ssoEditAttributes"	Specifies the attributes that will appear on the 1st edit page for the provider. These are usually server settings and have nothing to do with display attributes.
Collection name="dpEditAttributes"	Specifies the attributes that will appear on the edit page for the application helper. These are usually display attributes and there can be multiple attributes based on number of clients, etc.
Collection name="viewSelectOptions"	Specifies the different Calendar views displayed in the Calendar edit page.

See [“MS Exchange Calendar Channel” on page 178](#) for an example of a channel that uses MExchangeCalendarProvider.

MExchangeMailProvider

The mail provider works with the Microsoft Exchange Server to provide simple mail functionality.

[Code Example 4-20](#) shows the display profile definition for MExchangeMailProvider.

Code Example 4-20 Display Profile Definition for MExchangeMailProvider

```
<Provider name="MExchangeMailProvider" class="com.sun.portal.providers.mail.MailProvider">
  <Properties>
    <String name="title" value="Exchange Mail" />
    <String name="description" value="Exchange Mail" />
    <String name="width" value="thick" />
    <String name="refreshTime" value="0" />
    <String name="sortOrder" value="top" />
    <String name="editType" value="edit_subset" />
    <String name="productName" value="Sun ONE Portal Server" />
    <String name="fontFacel" value="Sans-serif" />
    <String name="ssoAdapter" value="exchangeMail" />
    <Integer name="numberHeaders" value="10" />
    <Boolean name="displayHeaders" value="true" />
    <Boolean name="sentFolderCopy" value="false" />
    <Boolean name="isEditable" value="true" />
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/mailhelp.html" />
    </ConditionalProperties>
    <ConditionalProperties condition="client" value="HTML">
      <Collection name="applicationHelperEdit">
        <String value="com.sun.portal.providers.mail.ExchangeMailHelper" />
      </Collection>
    </ConditionalProperties>
  </Properties>
</Provider>
```

Code Example 4-20 Display Profile Definition for MExchangeMailProvider (Continued)

```

        </Collection>
        <String name="applicationHelperURL"
value="com.sun.portal.providers.mail.ExchangeMailHelper" />
    </ConditionalProperties>
    <Collection name="ssoEditAttributes">
        <String name="host" value="string|Server Name:"/>
        <String name="port" value="string|IMAP Server Port:"/>
        <String name="uid" value="string|User Name:"/>
        <String name="password" value="password|User Password:"/>
        <String name="smtpServer" value="string|SMTP Server Name:"/>
        <String name="clientPort" value="string|Client Port:"/>
        <String name="winDomain" value="string|Domain:"/>
    </Collection>
    <Collection name="dpEditAttributes">
        <String name="numberHeaders" value="int|Number of Headers:"/>
        <String name="displayHeaders" value="check|Display Headers:"/>
        <String name="sortOrder" value="select|Sort Order:"/>
        <String name="sentFolderCopy" value="check|When sending a message place a copy
in Sent Folder: "/>
    </Collection>
    <Collection name="sortOrderSelectOptions">
        <String name="top" value="Most recent at top"/>
        <String name="bottom" value="Most recent at bottom"/>
    </Collection>
</Properties>
</Provider>

```

The following is a list of the properties specific to MExchangeMailProvider. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

String name="sortOrder"	Specifies the sort order for the messages currently displayed.
String name="ssoAdapter"	Specifies the SSOAdapter configuration used by this provider/channels.
Integer name="numberHeaders"	Specifies the limit of message headers to display. Hard limit is 30.
Boolean name="displayHeaders"	Specifies if the headers should be shown.
Boolean name="sentFolderCopy"	Specifies if sent messages should be copied to the Sent Folder (used by MA).
Collection name="applicationHelperEdit"	Specifies the mail application helpers that you can edit settings on.

String name="applicationHelperURL"	Specifies the default mail application helper.
Collection name="ssoEditAttributes"	Specifies the attributes that will appear on the 1st edit page for the provider. These are usually server settings and have nothing to do with display attributes.
Collection name="dpEditAttributes"	Specifies the attributes that will appear on the edit page for the application helper. These are usually display attributes and there can be multiple attributes based on number of clients, etc.
Collection name="sortOrderSelectOptions"	Specifies the different mail sort order options (recent at top or bottom).

See [“MS Exchange Mail Channel” on page 178](#) for an example of a channel that uses `MSEExchangeMailProvider`.

NotesProvider

`NotesProvider` enables the administrator or users the administrator has authorized to post a note to all users’ Desktops in the Notes channel.

[Code Example 4-21](#) shows the display profile definition for `NotesProvider`.

Code Example 4-21 Display Profile Definition for NotesProvider

```
<Provider name="NotesProvider" class="com.sun.portal.providers.notes.NotesProvider">
  <Properties>
    <String name="title" value="Notes Provider"/>
    <String name="width" value="thick"/>
    <String name="editType" value="edit_subset" advanced="true"/>
    <Boolean name="isEditable" value="false" advanced="true"/>
    <String name="description" value="Notes Provider Channel Sample Implementation"/>
    <String name="refreshTime" value="0" advanced="true"/>
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/notes.htm" advanced="true"/>
    </ConditionalProperties>
    <String name="helpURL" value="en/desktop/notes.htm" advanced="true"/>
    <String name="fontFace1" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <String name="location" value="/var/tmp/notes.txt"/>
    <String name="lines" value="2"/>
    <String name="maxLines" value="2"/>
    <Integer name="timeout" value="100"/>
  </Properties>
</Provider>
```

Code Example 4-21 Display Profile Definition for NotesProvider (*Continued*)

```

    <String name="timezone" value="PST"/>
  </Properties>
</Provider>

```

The following is a list of the properties specific to NotesProvider. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

String name="location"	Specifies the path to the text file, which contains the notes, in the file system.
String name="lines"	Specifies the number of lines of notes that is displayed in the channel.
String name="maxLines"	Specifies the maximum number of lines that can be displayed in the channel.
Integer name="timeout"	Specifies the time zone of the time stamp at which the notes were logged, either as an abbreviation such as PST, a full name such as America/Los_Angeles, or a custom ID such as GMT-8:00. Support of abbreviations is for JDK™ 1.1.x compatibility only and full names should be used.

Notes are stored and read in a text file in the following format:

```
userid | date | message
```

where | is the delimiter and date is the long value that denotes the time elapsed in milliseconds since January 1, 1970.

Following is a sample notes file.

```

User1|1007159465858|Message to Portal Desktop Team : Lets meet today at 2PM
User2|1007159465858|Information related to project is availble at home page

```

See [“Notes Channel” on page 179](#) for an example of a channel that uses NotesProvider.

PersonalNoteProvider

[Code Example 4-22](#) shows the display profile definition for PersonalNoteProvider.

Code Example 4-22 Display Profile Definition for PersonalNoteProvider

```
<Provider name="PersonalNoteProvider"
class="com.sun.portal.providers.personalnote.PersonalNoteProvider">
  <Properties>
    <String name="title" value="Personal Notes Provider"/>
    <String name="width" value="thick"/>
    <Boolean name="isEditable" value="true" />
    <String name="editType" value="edit_subset" advanced="true"/>
    <String name="description" value="PersonalNotes Provider Channel Sample
Implementation"/>
    <String name="refreshTime" value="0" advanced="true"/>
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/personalnotes.htm" advanced="true"/>
    </ConditionalProperties>
    <String name="helpURL" value="en/desktop/personalnotes.htm" advanced="true"/>
    <String name="fontFacel" value="Sans-serif"/>
    <Collection name="notes">
    </Collection>
  </Properties>
</Provider>
```

SimpleWebServiceProvider

SimpleWebServiceProvider, an extension of JSPProvider, makes simple web services available to an end user channel. SimpleWebServiceProvider dynamically constructs a user interface given a Web Services Description Language (WSDL) URL and a web service method name.

Using the URL, SimpleWebServiceProvider fetches the WSDL document, parses and validates it. Based on its content, SimpleWebServiceProvider generates input parameters to the method that return the information from the web service. The information is then displayed in the channel content window.

SimpleWebServiceProvider can generate channels that use the same web service, and the same method, so default parameter values can be stored using the Edit function.

SimpleWebServiceProvider supports basic data types such as String, int, and float as defined in the WSDL specification. It supports Complex Types if they are made up of only basic types (one level of nesting). There is no support for arrays.

SimpleWebServiceProvider can provide WSDL parsing for any other provider that needs it. SimpleWebServiceProvider is designed for stock quote or currency exchange rate content.

[Code Example 4-23](#) shows the display profile definition for SimpleWebServiceProvider.

Code Example 4-23 Display Profile Definition for SimpleWebServiceProvider

```
<Provider name="simpleWebService"
class="com.sun.portal.providers.simplewebservice.apache.ApacheWebServiceProvider">
  <Properties>
    <String name="title" value="***** Simple Web Service Provider *****"/>
    <String name="description" value="***** DESCRIPTION *****"/>
    <String name="width" value="thick"/>
    <String name="wsdlURL"
value="http://www.xmethods.net/sd/2001/CurrencyExchangeService.wsdl"/>
    <String name="methodName" value="getRate"/>
    <Boolean name="isDefaultShowOutput" value="false"/>
    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="editType" value="edit_subset" advanced="true"/>
    <String name="contentPage" value="webserviceContent.jsp"/>
    <String name="editPage" value="webserviceInputEdit.jsp"/>
    <Boolean name="showExceptions" value="false"/>
    <Boolean name="isDefaultAvailable" value="false"/>
    <String name="refreshTime" value="0" advanced="true"/>
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/webchann.htm" advanced="true"/>
    </ConditionalProperties>
    <String name="helpURL" value="en/desktop/webchann.htm" advanced="true"/>
    <String name="fontFacel" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <Integer name="descriptorCacheTimeout" value="600" advanced="false" lock="false"
merge="replace" propagate="true" />
    <Collection name="defaultInput">
    </Collection>
  </Properties>
</Provider>
```

The following is a list of the properties specific to SimpleWebServiceProvider. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

<code>String name="wsdlURL"</code>	Specifies the URL to web service WSDL.
<code>String name="methodName"</code>	Specifies the web service method name that is going to be executed.
<code>Boolean name="isDefaultShowOutput"</code>	Specifies the default value. If <code>true</code> , the channel uses the default input value. If <code>false</code> , the channel uses the user input value.
<code>String name="contentPage"</code>	Specifies the JSP that is used to generate the channel content (by using the <code>getContent</code> method).
<code>String name="editPage"</code>	Specifies the JSP that is used to generate the Edit page content (by using the <code>getEdit</code> method).
<code>Boolean name="showExceptions"</code>	If <code>true</code> , makes <code>SimpleWebServiceProvider</code> show exceptions generated while processing the JSP as the channel output for the <code>getContent</code> and <code>getEdit</code> methods. This can be useful for developing and troubleshooting your portal.
<code>Boolean name="isDefaultAvailable"</code>	If <code>true</code> , the default value is available from the profile database.
<code>Collection name="defaultInput"</code>	Specifies the default input value.

See [“Sample Simple Web Service Channel” on page 182](#) for an example of a channel that uses `SimpleWebServiceProvider`.

SimpleWebServiceConfigurableProvider

`SimpleWebServiceConfigurableProvider` is similar to `SimpleWebServiceProvider`, except that it permits users to use the Edit function to change URLs and methods, hence, it is configurable. See [“SimpleWebServiceProvider” on page 120](#) for details on how the provider works.

[Code Example 4-24](#) shows the display profile definition for `SimpleWebServiceConfigurableProvider`.

Code Example 4-24 Display Profile Definition for SimpleWebServiceConfigurableProvider

```

<Provider name="simpleWebServiceConfigurable"
class="com.sun.portal.providers.simplewebservice.apache.ApacheWebServiceProvider">
  <Properties>
    <String name="title" value="Configurable Simple Web Service Provider"/>
    <String name="description" value="***** DESCRIPTION *****"/>
    <String name="width" value="thick"/>
    <String name="wsdlURL"
value="http://www.xmethods.net/sd/2001/CurrencyExchangeService.wsdl"/>
    <String name="methodName" value="getRate"/>
    <Boolean name="isDefaultShowOutput" value="false"/>
    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="editType" value="edit_subset" advanced="true"/>
    <String name="contentPage" value="webserviceContent.jsp"/>
    <String name="editPage" value="webserviceWsdlEdit.jsp"/>
    <Boolean name="showExceptions" value="false"/>
    <Boolean name="isDefaultAvailable" value="false"/>
    <String name="refreshTime" value="0" advanced="true"/>
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/webchann.htm" advanced="true"/>
    </ConditionalProperties>
    <String name="helpURL" value="en/desktop/webchann.htm" advanced="true"/>
    <String name="fontFace1" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <Integer name="descriptorCacheTimeout" value="600" advanced="false" lock="false"
merge="replace" propagate="true" />
    <Collection name="defaultInput">
    </Collection>
  </Properties>
</Provider>

```

The following is a list of the properties specific to SimpleWebServiceConfigurableProvider. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

String name="wsdlURL"	Specifies the URL to web service WSDL.
String name="methodName"	Specifies the web service method name that is going to be executed.
Boolean name="isDefaultShowOutput"	Specifies the default value. If true, the channel uses the default input value. If false, the channel uses the user input value.

String name="contentPage"	Specifies the JSP that is used to generate the channel content (by using the <code>getContent()</code> method).
String name="editPage"	Specifies the JSP that is used to generate the Edit page content (by using the <code>getEdit()</code> method).
Boolean name="showExceptions"	If <code>true</code> , makes <code>SimpleWebServiceConfigurableProvider</code> show exceptions generated while processing the JSP as the channel output for the <code>getContent()</code> and <code>getEdit()</code> methods. This can be useful for developing and troubleshooting your portal.
Boolean name="isDefaultAvailable"	If <code>true</code> , the default value is available from the profile database.
Collection name="defaultInput"	Specifies the default input value.

See [“Sample Simple Web Service Configurable Channel” on page 182](#) for an example of a channel that uses `SimpleWebServiceConfigurableProvider`.

UserInfoProvider

`UserInfoProvider` collects information from the display profile and Sun ONE Identity Server. It displays a greeting, the user’s name, time zone, and locale, and has access to the user’s IMAP and SMTP data.

[Code Example 4-25](#) shows the display profile definition for `UserInfoProvider`.

Code Example 4-25 Display Profile Definition for `UserInfoProvider`

```
<Provider name="UserInfoProvider"
class="com.sun.portal.providers.userinfo.UserInfoProvider">
  <Properties>
    <String name="title" value="*** Userinfo Provider ***"/>
    <String name="description" value="*** DESCRIPTION ***"/>
    <String name="refreshTime" value="0" advanced="true"/>
    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="editType" value="edit_subset" advanced="true"/>
    <String name="width" value="thin" advanced="true"/>
    <ConditionalProperties condition="locale" value="en">
      <String name="helpURL" value="en/desktop/userinfo.htm" advanced="true"/>
    </ConditionalProperties>
    <String name="helpURL" value="desktop/userinfo.htm" advanced="true"/>
    <String name="fontFace1" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
```

Code Example 4-25 Display Profile Definition for UserInfoProvider (Continued)

```

    <String name="greeting" value="Welcome!" />
    <Collection name="tags" advanced="true">
      <String name="greeting" value="dp"/>
      <String name="fontFace1" value="dp"/>
      <String name="cn" value="attribute"/>
      <String name="givenname" value="attribute"/>
      <String name="sn" value="attribute"/>
      <String name="uid" value="attribute"/>
      <String name="sunPortalNetmailIMAPServerName" value="attribute"/>
      <String name="sunPortalNetmailSMTPServerName" value="attribute"/>
      <String name="sunPortalNetmailIMAPUserid" value="attribute"/>
      <String name="sunPortalNetmailIMAPPASSWORD" value="imappw"/>
      <String name="currentDate" value="datetime"/>
      <String name="timeLeft" value="datetime"/>
      <String name="maxIdle" value="datetime"/>
      <String name="preferredtimezone" value="timezone"/>
      <String name="timezoneList" value="timezone"/>
      <String name="locale" value="localelist"/>
      <String name="localeList" value="localelist"/>
      <String name="preferredlocale" value="localelist"/>
      <String name="membershipNewPassword" value="membershippw"/>
      <String name="membershipConfirmPassword" value="membershippw"/>
      <String name="membershipOriginalPassword" value="membershippw"/>
    </Collection>
    <Collection name="tagModules" advanced="true">
      <String name="dp" value="com.sun.portal.providers.userinfo.tag.DPtag"/>
      <String name="attribute"
value="com.sun.portal.providers.userinfo.tag.AttributeTag"/>
      <String name="datetime" value="com.sun.portal.providers.userinfo.DateTimeTag"/>
      <String name="timezone" value="com.sun.portal.providers.userinfo.TimezoneTag"/>
      <String name="localelist"
value="com.sun.portal.providers.userinfo.LocaleListTag"/>
      <String name="membershippw"
value="com.sun.portal.providers.userinfo.MembershipPWtag"/>
    </Collection>
    <Collection name="authTypes" advanced="true">
      <String value="Membership"/>
    </Collection>
    <String name="netmailServiceName" value="SunPortalNetMailService"/>
  </Properties>
</Provider>

```

The following is a list of the properties specific to UserInfoProvider. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

Collection name="tags"

Specifies the collection tags used in the user info template.

<code>String name="greeting"</code>	Specifies the tag name for greeting.
<code>String name="cn"</code>	Specifies the tag name for common name.
<code>String name="givenname"</code>	Specifies the given name.
<code>String name="sn"</code>	Specifies the surname.
<code>String name="uid"</code>	Specifies the user ID.
<code>String name="sunPortalNetmailIMAPServerName"</code>	Specifies the IMAP server name.
<code>String name="sunPortalNetmailSMTPServerName"</code>	Specifies the SMTP server name.
<code>String name="sunPortalNetmailIMAPUserid"</code>	Specifies the IMAP user ID.
<code>String name="sunPortalNetmailIMAPPassword"</code>	Specifies the IMAP password.
<code>String name="currentDate"</code>	Specifies the tag name for the current date.
<code>String name="timeLeft"</code>	Specifies the tag name for the time remaining.
<code>String name="maxIdle"</code>	Specifies the tag name for the maximum idle time.
<code>String name="preferredtimezone"</code>	Specifies the user's preferred time zone.
<code>String name="timezoneList"</code>	Specifies the tag name for the time zone.
<code>String name="locale"</code>	Specifies the user's locale.
<code>String name="localeList"</code>	Specifies the tag name for language.
<code>String name="preferredlocale"</code>	Specifies the user's preferred locale.
<code>String name="membershipNewPassword"</code>	Specifies the tag name for membership new password.
<code>String name="membershipConfirmPassword"</code>	Specifies the tag name for membership confirm password.
<code>String name="membershipOriginalPassword"</code>	Specifies the tag name for the original membership password.
<code>Collection name="tagModules"</code>	Specifies the collection of tag modules.
<code>String name="dp"</code>	Specifies the tag module to get and set user info channel properties.
<code>String name="attribute"</code>	Specifies the tag module to get and set user info channel attributes.

<code>String name="imapppw"</code>	Specifies the tag module to get and set IMAP password.
<code>String name="datetime"</code>	Specifies the tag module to get current time.
<code>String name="timezone"</code>	Specifies the tag module to get and set time zone list.
<code>String name="localelist"</code>	Specifies the tag module to get and set local list.
<code>String name="membershippw"</code>	Specifies the tag module to get and set membership password.
<code>Collection name="authTypes"</code>	Specifies the collection of authentication types.
<code>String value="Membership"</code>	Specifies the value of authentication type.
<code>String name="netmailServiceName"</code>	Specifies the NetMail service name.

See [“User Information Channel” on page 185](#) for an example of a channel that uses `UserInfoProvider`.

Display Profile Common Properties for Providers

The Sun™ ONE Portal Server software sample portal display profile XML fragments define the default values for all expected properties, so that channels that use the supplied providers do not have to define all the properties, only the ones that need to be different.

NOTE Depending on the merge priorities and locking assigned to the display profile documents that make up a user’s display profile, the ultimate property values that are returned to a user’s Desktop can change. See the *Sun ONE Portal Server 6.2 Administrator’s Guide* for more information on merging.

The following is a list of common properties for a `<Provider>` definition. In addition to the following properties, there are properties that are used in specific providers. For example, the `lines` and `maxLines` properties are required by the notes provider code.

The following two column table lists the general properties in the first (left) column and a brief description in the second (right) column.

<code>title</code>	Specifies the title that appears in the channel title bar in the Desktop.
<code>description</code>	<p>The description is displayed on the content page to give the user a little more information about what the channel is more. For example:</p> <pre>title: Bookmarks description: manage your portal-specific bookmarks</pre>
<code>refreshTime</code>	The <code>refreshTime</code> property controls how often a channel's content is reloaded. See “Channel Refresh Times and Container Caching” on page 213 for more information.
<code>editType</code>	Specifies the edit type, either <code>edit_complete</code> or <code>edit_subset</code> . If <code>edit_complete</code> , the provider's <code>getEdit()</code> method is responsible for generating the complete Edit page content. If <code>edit_subset</code> , a generic edit provider container is used to put a frame around the Edit page. The provider's <code>getEdit()</code> page is then called, and displays the content of the Edit page.
<code>isEditable</code>	Determines if the provider has an edit view. If true, the Edit button is generated in the channel title bar. If <code>isEditable</code> is false, no edit view is provided and no Edit button is generate in the title bar. See “isEditable” on page 130 for more information.
<code>width</code>	<p>A channel's width setting is a suggestion for containing channels as to how much screen real estate the channel may require. This value is only a suggestion; a container is not required to utilize this value for its contained channels. Possible values are <code>thin</code>, <code>thick</code>, <code>full_top</code>, or <code>full_bottom</code>. In general, these values only make sense for an HTML-based Desktop.</p>

helpURL

Specifies the online help URL, which can be either a fully qualified URL value or a relative path to the doc root location. For example, the online help URL for the bookmark channel is:

```
http://hostname:port/portal/docs/en/desktop/bkmark.html
```

This URL could also be defined as `desktop/bmark.html`. In this case, the provider context code figures out the doc root and the user locale, and locates the online help URL.

A return value of null signifies that this provider does not have a help page.

To have the provider code not generate a Help icon on the title bar for the channel, use a value of "".

fontFace1

Specifies the default font face for the channel, for example, `Sans-serif`.

productName

Specifies the name of the product, for example, Sun ONE Portal Server. This is not required¹ by all the providers.

Required properties and their values for the `<Provider>` definition are based on the provider interface. The required properties are necessary for the provider code, but the channel can set its own properties that override these values.

1. Required properties and their values for the provider definition are based on the Provider interface. The required properties are necessary for the provider code if the provider class extends `ProviderAdapter` or the `ProfileProviderAdapter` class. Note that the channel can set its own properties that override these values.

You can also define other properties in the `<Provider>` definition. For example:

```
<Provider name="MyProvider" class="com.sesta.providers.MyProvider">
  <Properties>
    ... required properties ...
    <String name="authorInfo" value="John Doe (john.doe@sesta.com)"/>
  </Properties>
</Provider>
```

isEditable

The `isEditable` property determines if the provider has an edit view (jtb-15). When `isEditable` is set to `true`, the Edit button is generated in the channel title bar. By clicking the Edit button, users can display an Edit page and customize channel settings such as whether the channel is minimized or detached. The Edit page is generated by the `Provider.getEdit()` method.

When `isEditable` is set to `false`, the Edit button is not generated in the channel title bar and users cannot change the settings for the channel. To implement an editable provider, the default no-operation implementations of `ProviderAdapter.getEdit()` and `ProviderAdapter.processEdit()` must be overridden (jtb-17). If the provider has the `getEdit()` and `processEdit()` methods defined, you can change the value of `isEditable` from `false` to `true` to cause the Edit button to appear in the channel title bar.

The following providers must have the `isEditable` property tag set to `false`, as, by default, there is no code for handling the Edit page:

- [URLScrapperProvider](#)
- [XMLProvider](#)
- [Rich Site Summary \(RSS\) Channel](#) (a channel of [XMLProvider](#))

You would need to extend these providers to enable them to have an edit view. See the *Sun ONE Portal Server 6.2 Developer's Guide* for more information on how to extend a provider.

The following providers can have the `isEditable` property tag set to `true` or `false`, but can only be used by an authorized user, such as the administrator:

- [NotesProvider](#)
- [MailCheckProvider](#)

The following providers have the `isEditable` property tag set to `true` by default, though you can change `isEditable` to `false` if you do not want users to be able to customize the channels:

- [AddressBookProvider](#)
- [AppProvider](#)
- [BookmarkProvider](#)
- [CalendarProvider](#)
- [DiscussionProvider](#)

- [IMProvider](#)
- [JSPPProvider](#)
- [LoginProvider](#)
- [MailProvider](#)
- [SearchProvider](#)
- [SimpleWebServiceProvider](#)
- [SubscriptionsProvider](#)
- [UserInfoProvider](#)

authlessState

The `authlessState` property determines how client specific state is managed when the Desktop is operating in authless mode. Client specific state is accessed via the `ProviderContext.getClientProperty()` methods. The `authlessState` client type property can take on three values: `client`, `server`, and `none`. When set to:

- `client`, authless state is stored on the client.
- `server`, authless state is stored on the server.
- `none`, no authless state is recorded and the `ProviderContext.getClientProperty()` methods have no effect.

By default, the `authlessState` client type property is not present, and defaults to `client` for HTML devices, and `none` for non-HTML devices. To modify the default value for a specific client type, add the `authlessState` client type property and set its value to either `client`, `server`, or `none`.

encoderClassName

The `encoderClassName` client type property maps an encoding algorithm (class) to a specific client type. This information is used by the `ProviderContext.escape()` method to escape strings in a client type specific manner.

ConditionalProperties

This provides a generic operation for retrieving conditional properties. The most common conditions are `locale` and `clientType`, but the API is generic in that it allows you to define and base properties on any sort of condition.

In the administration console, the conditional properties are displayed as condition-value and can be edited like collections. The conditional properties can be nested and can be added to a channel or inside another conditional property. Use the Add Property page to add a new conditional property.

The `<ConditionalProperties>` tag must be used to define the filtering criteria. The tag contains the following required attributes:

<code>condition</code>	Specifies name of the filter
<code>value</code>	Specifies the value to be used in the filter

In the display profile, the `<ConditionalProperties>` tag can be defined as outlined in [Code Example 4-26](#).

Code Example 4-26 `<ConditionalProperties>` Tag Usage Sample

```
<Properties>
  <String name="foo" value="bar">
    <ConditionalProperties condition="locale" value="de">
      <String name="foo" value="german bar">
        <String name="baz" value="a german baz value">
        </ConditionalProperties>
    <ConditionalProperties condition="client" value="nokia">
      <ConditionalProperties condition="locale" value="de">
        <String name="foo" value="nokia german bar">
        </ConditionalProperties>
      </ConditionalProperties>
    </ConditionalProperties>
  </Properties>
```

Display Profile Channel Definitions

This chapter contains information on the channel display profile definitions that ship with Sun™ ONE Portal Server software.

This chapter contains the following sections:

- [Overview of Channel Display Profile Definitions](#)
- [Display Profile Definition for Container Channels](#)
- [Display Profile Definition for Leaf Channels](#)

Overview of Channel Display Profile Definitions

The provider definition is the template that decides the properties for a channel. However, the display profile channel definition ultimately decides the values for the channel attributes. The display profile channel definition can define properties that overwrite the properties defined by the provider definition.

To facilitate understanding of what display profile channel definitions look like and how they work, this chapter contains the default display profile channel definitions for both the container and leaf channels shipped with the product.

Display Profile Definition for Container Channels

Container channels are channels that primarily generate its content by aggregating the content of other (its child) channels. A container channel allows for available and selected channel lists and can contain leaf channel definitions.

This section includes the default display profile XML fragments for the following container channels:

- [JSPTableContainer](#), [JSPTabContainer](#), and [JSPTabCustomTableContainer](#)
- [MyFrontPageFramePanelContainer](#), [MyFrontPageTabPanelContainer](#), and [MyFrontPageTemplatePanelContainer](#)
- [SamplesTabPanelContainer](#), [SamplesFramePanelContainer](#), and [SamplesTemplatePanelContainer](#)
- [SearchTabPanelContainer](#) and [ToolsTemplatePanelContainer](#)
- [JSPCreateChannelContainer](#), [JSPPopupContainer](#), and [JSPDynamicSingleContainer](#)
- [JSPPresetThemeContainer](#) and [JSPCustomThemeContainer](#)
- [TemplateTableContainer](#), [TemplateTabContainer](#), and [TemplateTabCustomTableContainer](#)
- [FrameTabContainer](#) and [FrameCustomTableContainer](#)
- [PredefinedFrontPageTabPanelContainer](#), [PredefinedSamplesTabPanelContainer](#), [PredefinedFrontPageTemplatePanelContainer](#), [PredefinedSamplesTemplatePanelContainer](#), [PredefinedToolsTemplatePanelContainer](#), [PredefinedFrontPageFramePanelContainer](#), and [PredefinedSamplesFramePanelContainer](#)

JSPTableContainer

The JSPTableContainer is used as a top level container channel for displaying the JSP based table desktop.

[Code Example 5-1](#) shows the display profile definition for JSPTableContainer channel.

Code Example 5-1 Display Profile Definition for JSPTableContainer

```
<Container name="JSPTableContainer" provider="JSPTableContainerProvider">
  <Properties>
    <String name="title" value="JSP Based Table Container Channel"/>
    <String name="contentPage" value="toptable.jsp"/>
    <String name="description" value="This is a test for front table containers"/>
    <String name="Desktop-fontFacel" value="Sans-serif"/>
    <String name="presetThemeChannel" value="JSPPresetThemeContainer" advanced="true"/>
    <String name="customThemeChannel" value="JSPCustomThemeContainer" advanced="true"/>
    <String name="parentTabContainer" value="FrameTabContainer" advanced="true"/>
    <Collection name="categories">
```

Code Example 5-1 Display Profile Definition for JSPTableContainer (Continued)

```

        <String value="Personal Channels" />
        <String value="Sample Channels" />
        <String value="News Channels" />
    </Collection>
    <Collection name="Personal Channels">
        <String value="UserInfo" />
        <String value="MailCheck" />
        <String value="Bookmark" />
        <String value="PersonalNotes" />
        <String value="App" />
    </Collection>
    <Collection name="Sample Channels">
        <String value="SampleSimpleWebService" />
        <String value="SampleJSP" />
        <String value="SampleXML" />
    </Collection>
    <Collection name="News Channels">
        <String value="SamplerRSS" />
        <String value="Notes" />
    </Collection>
    <Collection name="channelsRow" advanced="true">
        <String name="MailCheck" value="4" />
        <String name="App" value="5" />
        <String name="SampleSimpleWebService" value="6" />
        <String name="Bookmark" value="3" />
        <String name="SamplerRSS" value="2" />
        <String name="SampleXML" value="2" />
        <String name="Notes" value="3" />
        <String name="PersonalNotes" value="4" />
    </Collection>
    <Collection name="channelsIsRemovable">
        <Boolean name="UserInfo" value="false" />
    </Collection>
</Properties>
<Available>
    <Reference value="UserInfo" />
    <Reference value="MailCheck" />
    <Reference value="App" />
    <Reference value="SampleSimpleWebService" />
    <Reference value="Bookmark" />
    <Reference value="SampleJSP" />
    <Reference value="SamplerRSS" />
    <Reference value="SampleXML" />
    <Reference value="Notes" />
    <Reference value="PersonalNotes" />
</Available>
<Selected>
    <Reference value="UserInfo" />
    <Reference value="MailCheck" />
    <Reference value="App" />
    <Reference value="SampleSimpleWebService" />
    <Reference value="Bookmark" />
    <Reference value="SampleJSP" />
    <Reference value="SamplerRSS" />

```

Code Example 5-1 Display Profile Definition for JSPTableContainer *(Continued)*

```

    <Reference value="SampleXML" />
    <Reference value="Notes" />
    <Reference value="PersonalNotes" />
  </Selected>
</Channels>
</Container>

```

Properties can be configured from the administration console. For table containers, the following is a list of user editable properties. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

channelsRow	Defines a collection property to contain the row number value for channels in this container.
channelsColumn	Defines a collection property to contain the column number value for channels in this container.
channelsIsMinimized	Defines a collection property to contain the <code>isMinimized</code> value for channels in this container.
channelsIsDetached	Defines a collection property to contain the <code>isDetached</code> value for channels in this container.
Selected	Defines a list of selected channels for display on this container Desktop.
title	Specifies the title that appears in the channel title bar in the Desktop.

For more information on all the JSPTableContainer channel properties, see [“JSPTableContainerProvider.”](#)

MyFrontPageFramePanelContainer

MyFrontPageFramePanelContainer generates the MyFrontPage page on the FrameTabContainer (framed Desktop).

[Code Example 5-2 on page 137](#) shows the display profile definition for MyFrontPageFramePanelContainer channel.

Code Example 5-2 Display Profile Definition for MyFrontPageFramePanelContainer

```

<Container name="MyFrontPageFramePanelContainer"
provider="PredefinedFrontPageFramePanelContainerProvider">
  <Properties>
    <ConditionalProperties condition="locale" value="en" >
      <ConditionalProperties condition="locale" value="US" >
        <String name="title" value="My Front Page"/>
        <String name="description" value="Your Front Page"/>
      </ConditionalProperties>
    </ConditionalProperties>
    <String name="title" value="My Front Page"/>
    <String name="description" value="Your Front Page"/>
  </Properties>
  <Available>
    <Reference value="UserInfo"/>
    <Reference value="MailCheck"/>
    <Reference value="App"/>
    <Reference value="Bookmark"/>
    <Reference value="SampleJSP"/>
    <Reference value="SampleRSS"/>
    <Reference value="SampleXML"/>
  </Available>
  <Selected>
    <Reference value="UserInfo"/>
    <Reference value="MailCheck"/>
    <Reference value="App"/>
    <Reference value="Bookmark"/>
    <Reference value="SampleJSP"/>
    <Reference value="SampleRSS"/>
    <Reference value="SampleXML"/>
  </Selected>
  <Channels>
  </Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the MyFrontPageFramePanelContainer channel properties, see [“PredefinedFrontPageFramePanelContainerProvider.”](#)

SamplesTabPanelContainer

SamplesTabPanelContainer provides the Samples tab on JSPTabContainer (JSP based Tabbed Desktop).

[Code Example 5-3 on page 138](#) shows the display profile definition for SamplesTabPanelContainer channel.

Code Example 5-3 Display Profile Definition for SamplesTabPanelContainer

```

<Container name="SamplesTabPanelContainer"
provider="PredefinedSamplesTabPanelContainerProvider">
  <Properties>
    <ConditionalProperties condition="locale" value="en" >
      <ConditionalProperties condition="locale" value="US" >
        <String name="title" value="Samples"/>
        <String name="description" value="Samples Tab"/>
      </ConditionalProperties>
    </ConditionalProperties>
    <String name="title" value="Samples"/>
    <String name="description" value="Samples Tab"/>
  </Properties>
  <Available>
    <Reference value="SampleRSS"/>
    <Reference value="SampleURLScrapper"/>
    <Reference value="Notes"/>
    <Reference value="SampleSimpleWebService"/>
    <Reference value="Yahoo_channel"/>
    <Reference value="Yahoo_quotes"/>
    <Reference value="Yahoo_weather"/>
    <Reference value="Yahoo_comics"/>
  </Available>
  <Selected>
    <Reference value="SampleRSS"/>
    <Reference value="SampleURLScrapper"/>
    <Reference value="Notes"/>
    <Reference value="SampleSimpleWebService"/>
    <Reference value="Yahoo_channel"/>
    <Reference value="Yahoo_quotes"/>
    <Reference value="Yahoo_weather"/>
    <Reference value="Yahoo_comics"/>
  </Selected>
  <Channels>
  </Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the SamplesTabPanelContainer channel properties, see [“PredefinedSamplesTabPanelContainerProvider.”](#)

SearchTabPanelContainer

SearchTabPanelContainer provides the Search tab on JSPTabContainer (JSP based tabbed Desktop.)

[Code Example 5-4 on page 139](#) shows the display profile definition for SearchTabPanelContainer channel.

Code Example 5-4 Display Profile Definition for SearchTabPanelContainer

```

<Container name="SearchTabPanelContainer" provider="JSPTableContainerProvider">
  <Properties>
    <String name="title" value="Search"/>
    <String name="contentPage" value="tabtable.jsp"/>
    <String name="description" value="Search tab"/>
    <String name="presetThemeChannel" value="JSPPresetThemeContainer" advanced="true"/>
    <String name="customThemeChannel" value="JSPCustomThemeContainer" advanced="true"/>
    <String name="parentTabContainer" value="FrameTabContainer" advanced="true"/>
    <Collection name="categories">
      <String value="Search Channels"/>
    </Collection>
    <Collection name="Search Channels">
      <String value="Search"/>
    </Collection>
  </Properties>
  <Available>
    <Reference value="Search"/>
  </Available>
  <Selected>
    <Reference value="Search"/>
  </Selected>
  <Channels>
  </Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the SearchTabPanelContainer channel properties, see [“JSPTableContainerProvider.”](#)

MyFrontPageTabPanelContainer

MyFrontPageTabPanelContainer provides the MyFrontPage tab on JSPTabContainer (JSP based tabbed Desktop).

[Code Example 5-5](#) shows the display profile definition for MyFrontPageTabPanelContainer channel.

Code Example 5-5 Display Profile Definition for MyFrontPageTabPanelContainer

```

<Container name="MyFrontPageTabPanelContainer"
provider="PredefinedFrontPageTabPanelContainerProvider">
  <Properties>
    <ConditionalProperties condition="locale" value="en" >
      <ConditionalProperties condition="locale" value="US" >

```

Code Example 5-5 Display Profile Definition for MyFrontPageTabPanelContainer (Continued)

```

        <String name="title" value="My Front Page"/>
        <String name="description" value="Your Front Page"/>
    </ConditionalProperties>
</ConditionalProperties>
<String name="title" value="My Front Page"/>
<String name="description" value="Your Front Page"/>
</Properties>
<Available>
    <Reference value="UserInfo"/>
    <Reference value="MailCheck"/>
    <Reference value="App"/>
    <Reference value="Bookmark"/>
    <Reference value="MyFrontPageTabPanelContainer/Bookmark2"/>
    <Reference value="SampleJSP"/>
    <Reference value="SampleRSS"/>
    <Reference value="SampleXML"/>
</Available>
<Selected>
    <Reference value="UserInfo"/>
    <Reference value="MailCheck"/>
    <Reference value="App"/>
    <Reference value="Bookmark"/>
    <Reference value="MyFrontPageTabPanelContainer/Bookmark2"/>
    <Reference value="SampleJSP"/>
    <Reference value="SampleRSS"/>
    <Reference value="SampleXML"/>
</Selected>
<Channels>
    <Channel name="Bookmark2" provider="BookmarkProvider">
        <Properties>
            <String name="title" value="My Bookmarks #2"/>
            <String name="refreshTime" value="600"/>
            <Collection name="targets">
                <String value="Sun home page|http://www.sun.com"/>
                <String value="Everything you want to know about Sun ONE
... |http://www.sun.com/software/products/portal_srvr/home_portal.html"/>
                <String value="Sun ONE home page|http://www.sun.com/software"/>
            </Collection>
        </Properties>
    </Channel>
</Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the MyFrontPageTabPanelContainer channel properties, see [“PredefinedFrontPageTabPanelContainerProvider.”](#)

SamplesFramePanelContainer

SamplesFramePanelContainer provides the Samples page on FrameTabContainer.

[Code Example 5-6](#) shows the display profile definition for SamplesFramePanelContainer channel.

Code Example 5-6 Display Profile Definition for SamplesFramePanelContainer

```
<Container name="SamplesFramePanelContainer"
provider="PredefinedSamplesFramePanelContainerProvider">
  <Properties>
    <ConditionalProperties condition="locale" value="en" >
      <ConditionalProperties condition="locale" value="US" >
        <String name="title" value="Samples"/>
        <String name="description" value="Samples Tab"/>
      </ConditionalProperties>
    </ConditionalProperties>
    <String name="title" value="Samples"/>
    <String name="description" value="Samples Tab"/>
  </Properties>
  <Available>
    <Reference value="SampleRSS"/>
    <Reference value="SampleURLScrapper"/>
    <Reference value="Notes"/>
    <Reference value="SampleSimpleWebService"/>
  </Available>
  <Selected>
    <Reference value="SampleRSS"/>
    <Reference value="SampleURLScrapper"/>
    <Reference value="Notes"/>
    <Reference value="SampleSimpleWebService"/>
  </Selected>
  <Channels>
  </Channels>
</Container>
```

Properties can be configured from the administration console. For more information on all the SamplesFramePanelContainer channel properties, see [“PredefinedSamplesFramePanelContainerProvider.”](#)

JSPTabContainer

JSPTabContainer is used as the top level container for JSP based tabbed Desktop.

[Code Example 5-7 on page 142](#) shows the display profile definition for JSPTabContainer channel.

Code Example 5-7 Display Profile Definition for JSPTabContainer

```

<Container name="JSPTabContainer" provider="JSPTabContainerProvider">
  <Properties>
    <String name="title" value="JSP Tab Container Channel"/>
    <String name="description" value="This is a test for tab containers"/>
    <String name="contentPage" value="tab.jsp"/>
    <String name="editPage" value="tabedit.jsp"/>
    <String name="startTab" value="MyFrontPageTabPanelContainer"/>
    <Integer name="maxTabs" value="6"/>
    <String name="makeTabProvider" value="JSPTabCustomTableContainerProvider"
advanced="true"/>
    <String name="makeTabChannel" value="JSPTabCustomTableContainer" advanced="true"/>
    <Integer name="channelNumber" value="0"/>
    <String name="contentChannel" value="JSPContentContainer"/>
    <String name="presetThemeChannel" value="JSPPresetThemeContainer" advanced="true"/>
    <String name="customThemeChannel" value="JSPCustomThemeContainer" advanced="true"/>
    <Collection name="TabProperties">
      <Collection name="MyFrontPageTabPanelContainer">
        <Boolean name="removable" value="false"/>
        <Boolean name="renamable" value="true"/>
        <Boolean name="predefined" value="false"/>
      </Collection>
      <Collection name="SamplesTabPanelContainer">
        <Boolean name="removable" value="true"/>
        <Boolean name="renamable" value="true"/>
        <Boolean name="predefined" value="false"/>
      </Collection>
      <Collection name="JSPTabCustomTableContainer">
        <Boolean name="removable" value="true"/>
        <Boolean name="renamable" value="true"/>
        <Boolean name="predefined" value="false"/>
      </Collection>
      <Collection name="SearchTabPanelContainer">
        <Boolean name="removable" value="true"/>
        <Boolean name="renamable" value="true"/>
        <Boolean name="predefined" value="false"/>
      </Collection>
      <Collection name="PredefinedFrontPageTabPanelContainer">
        <Boolean name="removable" value="true"/>
        <Boolean name="renamable" value="true"/>
        <Boolean name="predefined" value="true"/>
      </Collection>
      <Collection name="PredefinedSamplesTabPanelContainer">
        <Boolean name="removable" value="true"/>
        <Boolean name="renamable" value="true"/>
        <Boolean name="predefined" value="true"/>
      </Collection>
    </Collection>
  </Properties>
  <Available>
    <Reference value="MyFrontPageTabPanelContainer"/>
    <Reference value="SamplesTabPanelContainer"/>
    <Reference value="SearchTabPanelContainer"/>
    <Reference value="PredefinedFrontPageTabPanelContainer"/>
  </Available>
</Container>

```

Code Example 5-7 Display Profile Definition for JSPTabContainer (Continued)

```

    <Reference value="PredefinedSamplesTabPanelContainer" />
  </Available>
  <Selected>
    <Reference value="MyFrontPageTabPanelContainer" />
    <Reference value="SamplesTabPanelContainer" />
    <Reference value="SearchTabPanelContainer" />
  </Selected>
  <Channels>
  </Channels>
</Container>

```

Properties can be configured from the administration console. For tab containers, the following is a list of user editable properties. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

startTab	Tab that is displayed when the user logs in.
channelNumber	Used in the naming of newly created tabs by user.
selected	Defines a list of selected channels for display on this container Desktop.

For more information on all the JSPTabContainer channel properties, see [“JSPTabContainerProvider.”](#)

JSPCreateChannelContainer

[Code Example 5-8](#) shows the display profile definition for JSPCreateChannelContainer channel.

Code Example 5-8 Display Profile Definition for JSPCreateChannelContainer

```

<Container name="JSPCreateChannelContainer" provider="JSPSingleContainerProvider"
advanced="true">
  <Properties>
    <String name="title" value="JSP Create Channel Channel"/>
    <String name="description" value="This is the JSP Create Channel Channel"/>
    <String name="contentPage" value="" propagate="false"/>
    <String name="editPage" value="createchanneledit.jsp"/>
    <String name="processPage" value="createchanneldoedit.jsp"/>
  </Properties>
</Container>

```

Code Example 5-8 Display Profile Definition for JSPCreateChannelContainer (Continued)

```

    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="editType" value="edit_complete" advanced="true"/>
    <Collection name="availableProviders">
      <String name="AddressBookProvider" value="Sun ONE AddressBook Channel"/>
      <String name="CalendarProvider" value="Sun ONE Calendar Channel"/>
      <String name="MailProvider" value="Sun ONE Mail Channel"/>
      <String name="MSExchangeAddressBookProvider" value="Exchange AddressBook
Channel"/>
      <String name="MSExchangeCalendarProvider" value="Exchange Calendar Channel"/>
      <String name="MSExchangeMailProvider" value="Exchange Mail Channel"/>
      <String name="LotusNotesAddressBookProvider" value="Lotus Notes AddressBook
Channel"/>
      <String name="LotusNotesCalendarProvider" value="Lotus Notes Calendar
Channel"/>
      <String name="LotusNotesMailProvider" value="Lotus Notes Mail Channel"/>
    </Collection>
  </Properties>
  <Available>
  </Available>
  <Selected>
  </Selected>
  <Channels>
  </Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the JSPCreateChannelContainer channel properties, see [“JSPSingleContainerProvider.”](#)

JSPPresetThemeContainer

JSPPresetThemeContainer displays the preset theme page, which allows users to select from a set of preset themes.

[Code Example 5-9](#) shows the display profile definition for JSPPresetThemeContainer channel.

Code Example 5-9 Display Profile Definition for JSPPresetThemeContainer

```

<Container name="JSPPresetThemeContainer" provider="JSPSingleContainerProvider"
advanced="true">
  <Properties>
    <ConditionalProperties condition="locale" value="en" >
      <String name="helpURL" value="en/desktop/theme.htm" advanced="true"/>
    </ConditionalProperties>
  </Properties>
</Container>

```

Code Example 5-9 Display Profile Definition for JSPPresetThemeContainer (Continued)

```

    <String name="helpURL" value="en/desktop/theme.htm" advanced="true" />
    <String name="title" value="JSP Preset Theme Channel"/>
    <String name="contentPage" value="" merge="replace" lock="false"
propagate="false"/>
    <String name="description" value="This is the JSP Preset Theme Channel"/>
    <String name="editPage" value="themededit.jsp"/>
    <String name="processPage" value="themedoedit.jsp"/>
    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="editType" value="edit_complete" advanced="true"/>
    <String name="presetThemeChannel" value="JSPPresetThemeContainer" advanced="true"/>
    <String name="customThemeChannel" value="JSPCustomThemeContainer" advanced="true"/>
  </Properties>
  <Available>
  </Available>
  <Selected>
  </Selected>
  <Channels>
  </Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the JSPPresetThemeContainer channel properties, see [“JSPSingleContainerProvider.”](#)

JSPCustomThemeContainer

JSPCustomThemeContainer displays the custom theme page, which allows users to customize the theme attributes.

[Code Example 5-10](#) shows the display profile definition for JSPCustomThemeContainer channel.

Code Example 5-10 Display Profile Definition for JSPCustomThemeContainer

```

<Container name="JSPCustomThemeContainer" provider="JSPSingleContainerProvider"
advanced="true">
  <Properties>
    <ConditionalProperties condition="locale" value="en" >
      <String name="helpURL" value="en/desktop/theme.htm" advanced="true"/>
    </ConditionalProperties>
    <String name="helpURL" value="en/desktop/theme.htm" advanced="true"/>
    <String name="title" value="JSP Custom Theme Channel"/>
    <String name="description" value="This is the JSP Custom Theme Channel" />
    <String name="contentPage" value="" merge="replace" lock="false"
propagate="false"/>

```

Code Example 5-10 Display Profile Definition for JSPCustomThemeContainer (Continued)

```

    <String name="editPage" value="customthemededit.jsp"/>
    <String name="processPage" value="customthemedoedit.jsp"/>
    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="editType" value="edit_complete" advanced="true"/>
  </Properties>
  <Available>
</Available>
  <Selected>
</Selected>
  <Channels>
</Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the JSPCustomThemeContainer channel properties, see [“JSPSingleContainerProvider.”](#)

JSPPopupContainer

JSPPopupContainer is no longer used in Sample Portal. It is included for backward compatibility. The detached windows on the desktop are now drawn by the TableContainer.

[Code Example 5-11](#) shows the display profile definition for JSPPopupContainer channel.

Code Example 5-11 Display Profile Definition for JSPPopupContainer

```

<Container name="JSPPopupContainer" provider="JSPSingleContainerProvider" advanced="true">
  <Properties>
    <String name="title" value="Popup Container"/>
    <String name="description" value="Popup container" />
    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="editType" value="edit_subset" advanced="true"/>
    <String name="contentPage" value="popup.jsp"/>
  </Properties>
  <Available>
</Available>
  <Selected>
</Selected>
  <Channels>
</Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the JSPPopupContainer channel properties, see [“JSPSingleContainerProvider.”](#)

TemplateTableContainer

TemplateTableContainer is used as the top level container for displaying a template based table Desktop.

[Code Example 5-12](#) shows the display profile definition for TemplateTableContainer channel.

Code Example 5-12 Display Profile Definition for TemplateTableContainer

```
<Container name="TemplateTableContainer" provider="TemplateTableContainerProvider">
  <Properties>
    <String name="title" value="Template Based Table Container"/>
    <String name="description" value="This is the channel for the front provider" />
    <String name="presetThemeChannel" value="JSPPresetThemeContainer" advanced="true"/>
    <String name="customThemeChannel" value="JSPCustomThemeContainer" advanced="true"/>
    <Collection name="channelsColumn" advanced="true">
      <String name="SampleJSP" value="2"/>
      <String name="SampleXML" value="2"/>
      <String name="Notes" value="2"/>
    </Collection>
    <Collection name="channelsRow" advanced="true">
      <String name="MailCheck" value="3"/>
      <String name="SampleRSS" value="2"/>
      <String name="SampleXML" value="2"/>
      <String name="App" value="5"/>
      <String name="SampleSimpleWebService" value="6"/>
      <String name="Bookmark" value="4"/>
      <String name="Notes" value="3"/>
    </Collection>
    <Collection name="channelsIsRemovable">
      <Boolean name="UserInfo" value="false"/>
    </Collection>
  </Properties>
  <Available>
    <Reference value="UserInfo"/>
    <Reference value="MailCheck"/>
    <Reference value="SampleRSS"/>
    <Reference value="SampleJSP"/>
    <Reference value="SampleXML"/>
    <Reference value="App"/>
    <Reference value="SampleSimpleWebService"/>
    <Reference value="Bookmark"/>
    <Reference value="Notes"/>
    <Reference value="PersonalNotes"/>
  </Available>
  <Selected>
    <Reference value="UserInfo"/>
  </Selected>
</Container>
```

Code Example 5-12 Display Profile Definition for TemplateTableContainer (Continued)

```

    <Reference value="MailCheck" />
    <Reference value="SampleRSS" />
    <Reference value="SampleJSP" />
    <Reference value="SampleXML" />
    <Reference value="App" />
    <Reference value="SampleSimpleWebService" />
    <Reference value="Bookmark" />
    <Reference value="Notes" />
    <Reference value="PersonalNotes" />
  </Selected>
</Channels>
</Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the TemplateTableContainer channel properties, see [“TemplateTableContainerProvider.”](#)

TemplateTabContainer

TemplateTabContainer is used as the top level container for displaying a template based tabbed desktop.

[Code Example 5-13](#) shows the display profile definition for TemplateTabContainer channel.

Code Example 5-13 Display Profile Definition for TemplateTabContainer

```

<Container name="TemplateTabContainer" provider="TemplateTabContainerProvider">
  <Properties>
    <String name="title" value="Template Based Tab Container Channel"/>
    <String name="description" value="This is a test for tab containers" />
    <String name="presetThemeChannel" value="JSPPresetThemeContainer" advanced="true"/>
    <String name="customThemeChannel" value="JSPCustomThemeContainer" advanced="true"/>
    <Boolean name="rtl" value="false"/>
    <String name="startTab" value="MyFrontPageTemplatePanelContainer"/>
    <String name="makeTabProvider" value="TemplateTabCustomTableContainerProvider"
advanced="true"/>
    <String name="makeTabChannel" value="TemplateTabCustomTableContainer"
advanced="true"/>
    <Integer name="channelNumber" value="0"/>
    <Integer name="maxTabs" value="4"/>
    <Collection name="TabProperties">
      <Collection name="MyFrontPageTemplatePanelContainer">
        <Boolean name="removable" value="true"/>
        <Boolean name="renamable" value="true"/>
      </Collection>
    </Collection>
  </Properties>
</Container>

```

Code Example 5-13 Display Profile Definition for TemplateTabContainer (Continued)

```

        <Boolean name="predefined" value="false" />
    </Collection>
    <Collection name="SamplesTemplatePanelContainer">
        <Boolean name="removable" value="false" />
        <Boolean name="renamable" value="true" />
        <Boolean name="predefined" value="false" />
    </Collection>
    <Collection name="ToolsTemplatePanelContainer">
        <Boolean name="removable" value="true" />
        <Boolean name="renamable" value="true" />
        <Boolean name="predefined" value="false" />
    </Collection>
    <Collection name="PredefinedFrontPageTemplatePanelContainer">
        <Boolean name="removable" value="true" />
        <Boolean name="renamable" value="true" />
        <Boolean name="predefined" value="true" />
    </Collection>
    <Collection name="PredefinedSamplesTemplatePanelContainer">
        <Boolean name="removable" value="true" />
        <Boolean name="renamable" value="true" />
        <Boolean name="predefined" value="true" />
    </Collection>
    <Collection name="PredefinedToolsTemplatePanelContainer">
        <Boolean name="removable" value="true" />
        <Boolean name="renamable" value="true" />
        <Boolean name="predefined" value="true" />
    </Collection>
    <Collection name="TemplateTabCustomTableContainer">
        <Boolean name="removable" value="true" />
        <Boolean name="renamable" value="true" />
        <Boolean name="predefined" value="false" />
    </Collection>
</Collection>
</Properties>
<Available>
    <Reference value="MyFrontPageTemplatePanelContainer" />
    <Reference value="SamplesTemplatePanelContainer" />
    <Reference value="ToolsTemplatePanelContainer" />
    <Reference value="PredefinedFrontPageTemplatePanelContainer" />
    <Reference value="PredefinedSamplesTemplatePanelContainer" />
    <Reference value="PredefinedToolsTemplatePanelContainer" />
</Available>
<Selected>
    <Reference value="MyFrontPageTemplatePanelContainer" />
    <Reference value="SamplesTemplatePanelContainer" />
</Selected>
<Channels>
</Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the TemplateTabContainer channel properties, see [“JSPTabContainerProvider.”](#)

MyFrontPageTemplatePanelContainer

MyFrontPageTemplatePanelContainer provides the MyFrontPage tab on TemplateTabContainer (template based tabbed Desktop).

Code Example 5-14 shows the display profile definition for MyFrontPageTemplatePanelContainer channel.

Code Example 5-14 Display Profile Definition for MyFrontPageTemplatePanelContainer

```
<Container name="MyFrontPageTemplatePanelContainer"
provider="PredefinedFrontPageTemplatePanelContainerProvider">
  <Properties>
    <ConditionalProperties condition="locale" value="en" >
      <ConditionalProperties condition="locale" value="US" >
        <String name="title" value="My Front Page"/>
        <String name="description" value="Your Front Page"/>
      </ConditionalProperties>
    </ConditionalProperties>
    <String name="title" value="My Front Page"/>
    <String name="description" value="Your Front Page" />
  </Properties>
  <Available>
    <Reference value="UserInfo" />
    <Reference value="MailCheck" />
    <Reference value="SampleRSS" />
    <Reference value="SampleJSP" />
    <Reference value="SampleXML" />
    <Reference value="App" />
    <Reference value="Bookmark" />
  </Available>
  <Selected>
    <Reference value="UserInfo" />
    <Reference value="MailCheck" />
    <Reference value="SampleRSS" />
    <Reference value="SampleJSP" />
    <Reference value="SampleXML" />
    <Reference value="App" />
    <Reference value="Bookmark" />
  </Selected>
  <Channels>
  </Channels>
</Container>
```

Properties can be configured from the administration console. For more information on all the MyFrontPageTemplatePanelContainer channel properties, see [“PredefinedFrontPageTemplatePanelContainerProvider.”](#)

SamplesTemplatePanelContainer

SamplesTemplatePanelContainer provides the Samples tab on TemplateTabContainer (template based tabbed Desktop).

[Code Example 5-15](#) shows the display profile definition for the SamplesTemplatePanelContainer channel.

Code Example 5-15 Display Profile Definition for SamplesTemplatePanelContainer

```
<Container name="SamplesTemplatePanelContainer"
provider="PredefinedSamplesTemplatePanelContainerProvider">
  <Properties>
    <ConditionalProperties condition="locale" value="en" >
      <ConditionalProperties condition="locale" value="US" >
        <String name="title" value="Samples"/>
        <String name="description" value="Samples Tab"/>
      </ConditionalProperties>
    </ConditionalProperties>
    <String name="title" value="Samples"/>
    <String name="description" value="Samples Tab"/>
  </Properties>
  <Available>
    <Reference value="SampleRSS"/>
    <Reference value="SampleURLScrapper"/>
    <Reference value="SampleSimpleWebService"/>
    <Reference value="Notes"/>
  </Available>
  <Selected>
    <Reference value="SampleRSS"/>
    <Reference value="SampleURLScrapper"/>
    <Reference value="SampleSimpleWebService"/>
    <Reference value="Notes"/>
  </Selected>
  <Channels>
  </Channels>
</Container>
```

Properties can be configured from the administration console. For more information on all the SamplesTemplatePanelContainer channel properties, see [“PredefinedSamplesTemplatePanelContainerProvider.”](#)

ToolsTemplatePanelContainer

ToolsTemplatePanelContainer provides the Tools tab on TemplateTabContainer (template based tabbed Desktop).

[Code Example 5-16](#) shows the display profile definition for the ToolsTemplatePanelContainer channel.

Code Example 5-16 Display Profile Definition for ToolsTemplatePanelContainer

```
<Container name="ToolsTemplatePanelContainer"
provider="PredefinedToolsTemplatePanelContainerProvider">
  <Properties>
    <String name="title" value="Tools"/>
    <String name="description" value="Tools Tab"/>
  </Properties>
  <Available>
    <Reference value="UserInfo"/>
    <Reference value="MailCheck"/>
    <Reference value="App"/>
    <Reference value="Bookmark"/>
    <Reference value="Notes"/>
  </Available>
  <Selected>
    <Reference value="UserInfo"/>
    <Reference value="MailCheck"/>
    <Reference value="App"/>
    <Reference value="Bookmark"/>
    <Reference value="Notes"/>
  </Selected>
  <Channels>
  </Channels>
</Container>
```

Properties can be configured from the administration console. For more information on all the ToolsTemplatePanelContainer channel properties, see [“PredefinedToolsTemplatePanelContainerProvider.”](#)

FrameTabContainer

FrameTabContainer is used as a top level container for displaying the Frame based desktop.

[Code Example 5-17 on page 153](#) shows the display profile definition for the FrameTabContainer channel.

Code Example 5-17 Display Profile Definition for FrameTabContainer

```

<Container name="FrameTabContainer" provider="JSPTabContainerProvider">
  <Properties>
    <ConditionalProperties condition="locale" value="en" >
      <String name="helpURL" value="en/desktop/fdesktop.htm" advanced="true"/>
    </ConditionalProperties>
    <String name="helpURL" value="en/desktop/fdesktop.htm" advanced="true"/>
    <String name="title" value="Framed Tab Container Channel"/>
    <String name="description" value="This is a test for frame containers" />
    <String name="contentPage" value="frametab.jsp"/>
    <String name="editPage" value="frametabedit.jsp"/>
    <String name="startTab" value="MyFrontPageFramePanelContainer"/>
    <Integer name="maxTabs" value="4"/>
    <String name="makeTabProvider" value="JSPFrameCustomTableContainerProvider"
advanced="true"/>
    <String name="makeTabChannel" value="FrameCustomTableContainer" advanced="true"/>
    <Integer name="channelNumber" value="0"/>
    <String name="contentChannel" value="JSPContentContainer"/>
    <String name="presetThemeChannel" value="JSPPresetThemeContainer" advanced="true"/>
    <String name="customThemeChannel" value="JSPCustomThemeContainer" advanced="true"/>
    <Collection name="TabProperties">
      <Collection name="MyFrontPageFramePanelContainer">
        <Boolean name="removable" value="false"/>
        <Boolean name="renamable" value="true"/>
        <Boolean name="predefined" value="false"/>
      </Collection>
      <Collection name="SamplesFramePanelContainer">
        <Boolean name="removable" value="true"/>
        <Boolean name="renamable" value="true"/>
        <Boolean name="predefined" value="false"/>
      </Collection>
      <Collection name="PredefinedFrontPageFramePanelContainer">
        <Boolean name="removable" value="true"/>
        <Boolean name="renamable" value="true"/>
        <Boolean name="predefined" value="true"/>
      </Collection>
      <Collection name="PredefinedSamplesFramePanelContainer">
        <Boolean name="removable" value="true"/>
        <Boolean name="renamable" value="true"/>
        <Boolean name="predefined" value="true"/>
      </Collection>
      <Collection name="FrameCustomTableContainer">
        <Boolean name="removable" value="true"/>
        <Boolean name="renamable" value="true"/>
        <Boolean name="predefined" value="false"/>
      </Collection>
    </Collection>
  </Properties>
  <Available>
    <Reference value="MyFrontPageFramePanelContainer"/>
    <Reference value="SamplesFramePanelContainer"/>
    <Reference value="PredefinedFrontPageFramePanelContainer"/>
    <Reference value="PredefinedSamplesFramePanelContainer"/>
  </Available>

```

Code Example 5-17 Display Profile Definition for FrameTabContainer (Continued)

```

<Selected>
  <Reference value="MyFrontPageFramePanelContainer" />
  <Reference value="SamplesFramePanelContainer" />
</Selected>
<Channels>
</Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the FrameTabContainer channel properties, see [“JSPTabContainerProvider.”](#)

JSPDynamicSingleContainer

JSPDynamicSingleContainer demonstrates the new method, `getSelectedChannel(HttpServletRequest req)`, in the `JSPSingleContainerProvider`. Since the display profile allows for only one hard-coded channel name for the `JSPSingleContainer`, the `JSPDynamicSingleContainer` enhances the single container’s capability by taking a channel name in the request parameter and displaying the channel on the Desktop.

That is, this method gets the selected channel name by querying the request for parameter `ContainerName.selectedChannel` and stores the `selectedChannel` name in session for future requests. For example, for a URL of the type:

```
http://hostname.domain/portal/dt?action=content&provider=ContainerName&ContainerName.selectedChannel=ChannelName
```

`ChannelName` is taken from the request parameter (`ContainerName.selectedChannel`) instead of from the display profile. If `selectedChannel` cannot be obtained either from the request or from the session, then `selectedChannel` from a call to `getSelectedChannel()` is returned.

[Code Example 5-18](#) shows the display profile definition for the `JSPDynamicSingleContainer` channel.

Code Example 5-18 Display Profile Definition for JSPDynamicSingleContainer

```

<Container name="JSPDynamicSingleContainer" provider="JSPSingleContainerProvider"
advanced="true">
  <Properties>
    <String name="title" value="Dynamic Single Container"/>
  </Properties>
</Container>

```

Code Example 5-18 Display Profile Definition for JSPDynamicSingleContainer *(Continued)*

```

    <String name="description" value="Dynamic Single Container" />
    <String name="contentPage" value="dynamicSingle.jsp" />
    <Boolean name="isEditable" value="false" advanced="true" />
  </Properties>
  <Available>
    <Reference value="SampleJSP" />
  </Available>
  <Selected>
    <Reference value="SampleJSP" />
  </Selected>
  <Channels>
  </Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the JSPDynamicSingleContainer channel properties, see [“JSPSingleContainerProvider.”](#)

PredefinedFrontPageTabPanelContainer

PredefinedFrontPageTabPanelContainer is used as the predefined tab for MyFrontPage tab used when the user creates a new tab based on an existing tab from the make New Tab page on JSPTabContainer.

[Code Example 5-19](#) shows the display profile definition for the PredefinedFrontPageTabPanelContainer channel.

Code Example 5-19 Display Profile Definition for PredefinedFrontPageTabPanelContainer

```

<Container name="PredefinedFrontPageTabPanelContainer"
provider="PredefinedFrontPageTabPanelContainerProvider">
  <Properties/>
  <Available>
    <Reference value="MailCheck" />
    <Reference value="MailCheck" />
    <Reference value="App" />
    <Reference value="Bookmark" />
    <Reference value="PredefinedFrontPageTabPanelContainer/Bookmark2" />
    <Reference value="SampleJSP" />
    <Reference value="SampleRSS" />
    <Reference value="SampleXML" />
  </Available>
  <Selected>
    <Reference value="MailCheck" />
    <Reference value="MailCheck" />

```

Code Example 5-19 Display Profile Definition for PredefinedFrontPageTabPanelContainer (Continued)

```

    <Reference value="App"/>
    <Reference value="Bookmark"/>
    <Reference value="PredefinedFrontPageTabPanelContainer/Bookmark2"/>
    <Reference value="SampleJSP"/>
    <Reference value="SampleRSS"/>
    <Reference value="SampleXML"/>
  </Selected>
</Channels>
  <Channel name="Bookmark2" provider="BookmarkProvider">
    <Properties>
      <String name="title" value="My Bookmarks #2"/>
      <String name="refreshTime" value="600"/>
      <Collection name="targets">
        <String value="Sun home page|http://www.sun.com"/>
        ...|http://www.sun.com/software/products/portal_srvr/home_portal.html"/>
        <String value="Everything you want to know about Sun ONE
        <String value="Sun ONE home page|http://www.sun.com/software"/>
      </Collection>
    </Properties>
  </Channel>
</Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the PredefinedFrontPageTabPanelContainer channel properties, see [“PredefinedSamplesTabPanelContainerProvider.”](#)

PredefinedSamplesTabPanelContainer

PredefinedSamplesTabPanelContainer is used as the predefined tab for Samples tab when the user creates a new tab based on an existing tab from the make New Tab page on JSPTabContainer.

[Code Example 5-20](#) shows the display profile definition for the PredefinedSamplesTabPanelContainer channel.

Code Example 5-20 Display Profile Definition for PredefinedSamplesTabPanelContainer

```

<Container name="PredefinedSamplesTabPanelContainer"
provider="PredefinedSamplesTabPanelContainerProvider">
  <Properties/>
  <Available>
    <Reference value="SampleRSS"/>
    <Reference value="SampleURLScraper"/>
    <Reference value="Notes"/>
  </Available>
</Container>

```

Code Example 5-20 Display Profile Definition for PredefinedSamplesTabPanelContainer (Continued)

```

    <Reference value="SampleSimpleWebService" />
    <Reference value="Yahoo_channel" />
    <Reference value="Yahoo_quotes" />
    <Reference value="Yahoo_weather" />
    <Reference value="Yahoo_comics" />
  </Available>
  <Selected>
    <Reference value="SampleRSS" />
    <Reference value="SampleURLScrapper" />
    <Reference value="Notes" />
    <Reference value="SampleSimpleWebService" />
    <Reference value="Yahoo_channel" />
    <Reference value="Yahoo_quotes" />
    <Reference value="Yahoo_weather" />
    <Reference value="Yahoo_comics" />
  </Selected>
  <Channels>
  </Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the PredefinedSamplesTabPanelContainer channel properties, see ["PredefinedSamplesTabPanelContainerProvider."](#)

JSPTabCustomTableContainer

JSPTabCustomTableContainer is used when the user creates a new tab from scratch from the Make New Tab page in JSPTabContainer.

[Code Example 5-21](#) shows the display profile definition for the JSPTabCustomTableContainer channel.

Code Example 5-21 Display Profile Definition for JSPTabCustomTableContainer

```

<Container name="JSPTabCustomTableContainer" provider="JSPTabCustomTableContainerProvider">
  <Properties/>
  <Available>
    <Reference value="UserInfo" />
    <Reference value="MailCheck" />
    <Reference value="App" />
    <Reference value="Bookmark" />
    <Reference value="SampleJSP" />
    <Reference value="SampleRSS" />
    <Reference value="SampleXML" />
    <Reference value="SampleURLScrapper" />
    <Reference value="Notes" />
  </Available>
</Container>

```

Code Example 5-21 Display Profile Definition for JSPTabCustomTableContainer (Continued)

```

    <Reference value="PersonalNotes" />
    <Reference value="Search" />
    <Reference value="SampleSimpleWebService" />
  </Available>
  <Selected>
</Selected>
  <Channels>
  </Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the JSPTabCustomTableContainer channel properties, see [“JSPTabCustomTableContainerProvider.”](#)

PredefinedFrontPageTemplatePanelContainer

PredefinedFrontPageTemplatePanelContainer is used as the predefined tab for MyFrontPage tab when the user creates a new tab based on an existing tab from the make New Tab page on TemplateTabContainer.

[Code Example 5-22](#) shows the display profile definition for the PredefinedFrontPageTemplatePanelContainer channel.

Code Example 5-22 Display Profile Definition for PredefinedFrontPageTemplatePanelContainer

```

<Container name="PredefinedFrontPageTemplatePanelContainer"
provider="PredefinedFrontPageTemplatePanelContainerProvider">
  <Properties/>
  <Available>
    <Reference value="UserInfo" />
    <Reference value="MailCheck" />
    <Reference value="App" />
    <Reference value="Bookmark" />
    <Reference value="SampleJSP" />
    <Reference value="SampleRSS" />
    <Reference value="SampleXML" />
  </Available>
  <Selected>
    <Reference value="UserInfo" />
    <Reference value="MailCheck" />
    <Reference value="App" />
    <Reference value="Bookmark" />
    <Reference value="SampleJSP" />
    <Reference value="SampleRSS" />
    <Reference value="SampleXML" />
  </Selected>
</Container>

```

Code Example 5-22 Display Profile Definition for PredefinedFrontPageTemplatePanelContainer

```

</Selected>
<Channels>
</Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the PredefinedFrontPageTemplatePanelContainer channel properties, see “[PredefinedFrontPageTemplatePanelContainerProvider.](#)”

PredefinedSamplesTemplatePanelContainer

PredefinedSamplesTemplatePanelContainer is used as the predefined tab for Samples tab when the user creates a new tab based on an existing tab from the make New Tab page on TemplateTabContainer.

[Code Example 5-23](#) shows the display profile definition for the PredefinedSamplesTemplatePanelContainer channel.

Code Example 5-23 Display Profile Definition for PredefinedSamplesTemplatePanelContainer

```

<Container name="PredefinedSamplesTemplatePanelContainer"
provider="PredefinedSamplesTemplatePanelContainerProvider">
  <Properties/>
  <Available>
    <Reference value="SampleRSS"/>
    <Reference value="SampleURLScrapper"/>
    <Reference value="SampleSimpleWebService"/>
    <Reference value="Notes"/>
  </Available>
  <Selected>
    <Reference value="SampleRSS"/>
    <Reference value="SampleURLScrapper"/>
    <Reference value="SampleSimpleWebService"/>
    <Reference value="Notes"/>
  </Selected>
  <Channels>
  </Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the PredefinedSamplesTemplatePanelContainer channel properties, see “[PredefinedSamplesTemplatePanelContainerProvider.](#)”

PredefinedToolsTemplatePanelContainer

[Code Example 5-24](#) shows the display profile definition for the PredefinedToolsTemplatePanelContainer channel.

Code Example 5-24 Display Profile Definition for PredefinedToolsTemplatePanelContainer

```
<Container name="PredefinedToolsTemplatePanelContainer"
provider="PredefinedToolsTemplatePanelContainerProvider">
  <Properties/>
  <Available>
    <Reference value="UserInfo" />
    <Reference value="MailCheck" />
    <Reference value="App" />
    <Reference value="Bookmark" />
    <Reference value="Notes" />
  </Available>
  <Selected>
    <Reference value="UserInfo" />
    <Reference value="MailCheck" />
    <Reference value="App" />
    <Reference value="Bookmark" />
    <Reference value="Notes" />
  </Selected>
  <Channels>
  </Channels>
</Container>
```

Properties can be configured from the administration console. For more information on all the PredefinedToolsTemplatePanelContainer channel properties, see [“PredefinedToolsTemplatePanelContainerProvider.”](#)

TemplateTabCustomTableContainer

TemplateTabCustomTableContainer is used when the user creates a new tab from scratch from the Make New Tab page in TemplateTabContainer.

[Code Example 5-25 on page 160](#) shows the display profile definition for the TemplateTabCustomTableContainer channel.

Code Example 5-25 Display Profile Definition for TemplateTabCustomTableContainer

```
<Container name="TemplateTabCustomTableContainer"
provider="TemplateTabCustomTableContainerProvider">
  <Properties/>
  <Available>
```

Code Example 5-25 Display Profile Definition for TemplateTabCustomTableContainer (Continued)

```

    <Reference value="UserInfo" />
    <Reference value="MailCheck" />
    <Reference value="App" />
    <Reference value="Bookmark" />
    <Reference value="SampleJSP" />
    <Reference value="SampleRSS" />
    <Reference value="SampleXML" />
    <Reference value="SampleURLScrapper" />
    <Reference value="Notes" />
    <Reference value="PersonalNotes" />
    <Reference value="Search" />
    <Reference value="SampleSimpleWebService" />
  </Available>
  <Selected>
</Selected>
  <Channels>
  </Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the TemplateTabCustomTableContainer channel properties, see [“TemplateTabCustomTableContainerProvider.”](#)

PredefinedFrontPageFramePanelContainer

PredefinedFrontPageFramePanelContainer is used as the predefined tab for MyFrontPage tab when the user creates a new page based on an existing page from the make New Page page on FrameTabContainer.

[Code Example 5-26](#) shows the display profile definition for the PredefinedFrontPageFramePanelContainer channel.

Code Example 5-26 Display Profile Definition for PredefinedFrontPageFramePanelContainer

```

<Container name="PredefinedFrontPageFramePanelContainer"
provider="PredefinedFrontPageFramePanelContainerProvider">
  <Properties/>
  <Available>
    <Reference value="UserInfo" />
    <Reference value="MailCheck" />
    <Reference value="App" />
    <Reference value="Bookmark" />
    <Reference value="SampleJSP" />
    <Reference value="SampleRSS" />
    <Reference value="SampleXML" />
  </Available>
</Container>

```

Code Example 5-26 Display Profile Definition for PredefinedFrontPageFramePanelContainer (Continued)

```

</Available>
<Selected>
  <Reference value="UserInfo" />
  <Reference value="MailCheck" />
  <Reference value="App" />
  <Reference value="Bookmark" />
  <Reference value="SampleJSP" />
  <Reference value="SampleRSS" />
  <Reference value="SampleXML" />
</Selected>
<Channels>
</Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the JSPTableContainer channel properties, see [“PredefinedFrontPageFramePanelContainerProvider.”](#)

PredefinedSamplesFramePanelContainer

PredefinedSamplesFramePanelContainer is used as the predefined tab for Samples tab when the user creates a new page based on an existing page from the make New Page page on FrameTabContainer.

[Code Example 5-28](#) shows the display profile definition for the PredefinedSamplesFramePanelContainer channel.

Code Example 5-27 Display Profile Definition for PredefinedSamplesFramePanelContainer

```

<Container name="PredefinedSamplesFramePanelContainer"
provider="PredefinedSamplesFramePanelContainerProvider">
  <Properties>
  </Properties>
  <Available>
    <Reference value="SampleRSS" />
    <Reference value="SampleURLScrapper" />
    <Reference value="Notes" />
    <Reference value="SampleSimpleWebService" />
  </Available>
  <Selected>
    <Reference value="SampleRSS" />
    <Reference value="SampleURLScrapper" />
    <Reference value="Notes" />
    <Reference value="SampleSimpleWebService" />
  </Selected>

```

Code Example 5-27 Display Profile Definition for PredefinedSamplesFramePanelContainer (Continued)

```

    <Channels>
  </Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the PredefinedSamplesFramePanelContainer channel properties, see [“PredefinedSamplesFramePanelContainerProvider.”](#)

FrameCustomTableContainer

FrameCustomTableContainer is used when the user creates a new tab from scratch from the Make New Tab page in FrameTabContainer.

[Code Example 5-28](#) shows the display profile definition for the FrameCustomTableContainer channel.

Code Example 5-28 Display Profile Definition for FrameCustomTableContainer

```

<Container name="FrameCustomTableContainer"
provider="JSPFrameCustomTableContainerProvider">
  <Properties/>
  <Available>
    <Reference value="UserInfo" />
    <Reference value="MailCheck" />
    <Reference value="App" />
    <Reference value="Bookmark" />
    <Reference value="SampleJSP" />
    <Reference value="SampleRSS" />
    <Reference value="SampleXML" />
    <Reference value="SampleURLScrapper" />
    <Reference value="Notes" />
    <Reference value="PersonalNotes" />
    <Reference value="Search" />
    <Reference value="SampleSimpleWebService" />
  </Available>
  <Selected>
</Selected>
  <Channels>
</Channels>
</Container>

```

Properties can be configured from the administration console. For more information on all the `FrameCustomTableContainer` channel properties, see [“JSPFrameCustomTableContainerProvider.”](#)

Display Profile Definition for Leaf Channels

Leaf channels are channels typically aggregated on a page with other channels and generates its own content. This section includes the display profile XML fragments for the following channels:

- [Address Book Channel](#)
- [Applications List Channel](#)
- [Bookmark Channel](#)
- [Calendar Channel](#)
- [Discussion Channels](#)
- [Instant Messaging Channel](#)
- [JSP Channel](#)
- [Login Channel](#)
- [Lotus Notes Address Book Channel](#)
- [Lotus Notes Calendar Channel](#)
- [Lotus Notes Mail Channel](#)
- [Mail Channel](#)
- [Mail Check Channel](#)
- [MS Exchange Address Book Channel](#)
- [MS Exchange Calendar Channel](#)
- [MS Exchange Mail Channel](#)
- [Notes Channel](#)
- [Search Channel](#)
- [Sample Simple Web Service Channel](#)
- [Sample Simple Web Service Configurable Channel](#)

- [Subscriptions Channel](#)
- [Sample RSS Channel](#)
- [User Information Channel](#)
- [Sample URL Scraper Channel](#)
- [Sample XML Channel](#)

NOTE The channel-defined properties overwrite the corresponding properties in the channel provider. Properties not defined in the channel are picked up from the channel Provider.

Address Book Channel

The address book channel displays entries from the users Sun™ One Messaging Servers address book.

[Code Example 5-29](#) shows the display profile channel definition for the AddressBook channel.

Code Example 5-29 Display Profile Definition for the Address Book Channel

```
<Channel name="AddressBook" provider="AddressBookProvider">
  <Properties/>
</Channel>
```

Properties can be configured from the administration console. The following AddressBook channel properties can be configured by the user. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

displayEntries	Specifies if the entries should be shown.
numEntries	Specifies the number of entries to display.
sunPortalABSorderBy	Specifies the value of the entries displayed to sort by.
sunPortalABSorderBy	Specifies the sort order of the entries displayed.

For a detailed list of properties specific to the `AddressBookProvider`, see [“AddressBookProvider.”](#)

Applications List Channel

The applications list channel enables a user to add or remove applications from a list of applications.

[Code Example 5-30](#) shows the display profile channel definition for the applications list channel.

Code Example 5-30 Display Profile Definition for the Applications List Channel

```
<Channel name="App" provider="AppProvider">
  <Properties>
    <String name="refreshTime" value="600"/>
    <Collection name="targets">
  </Collection>
    <Collection name="userApps">
  </Collection>
    <String name="title" value="My Applications"/>
  </Properties>
</Channel>
```

Properties can be configured from the administration console. The following property can be configured by the user. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

<code>userApps</code>	Specifies the list of applications that appear in the applications channel.
-----------------------	---

For a detailed list of properties specific to the `AppProvider`, see [“AppProvider.”](#)

Bookmark Channel

The bookmark channel enables users to store URLs.

[Code Example 5-31 on page 167](#) shows the display profile definition for the bookmark channel.

Code Example 5-31 Display Profile Definition for the Bookmark Channel

```

<Channel name="Bookmark" provider="BookmarkProvider">
  <Properties>
    <String name="title" value="My Bookmarks"/>
    <String name="refreshTime" value="600"/>
    <Collection name="targets">
      <String value="Sun home page|http://www.sun.com"/>
      <String value="Everything you want to know about Sun ONE ...|
http://www.sun.com/software/products/portal_srvr/home_portal.html"/>
      <String value="Sun ONE home page|http://www.sun.com/software"/>
    </Collection>
  </Properties>
</Channel>

```

Properties can be configured from the administration console. The following is a user editable bookmark channel property. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

targets	Specifies the target URL. For example, three URLs: <pre> http://www.sun.com http://www.sun.com/software/products/portal_srvr/home_portal.html http://www.sun.com/software. </pre>
---------	--

For more information on the other bookmark channel properties, see [“BookmarkProvider.”](#)

Calendar Channel

The calendar channel displays events and task from the Sun™ One Calendar Server.

[Code Example 5-32 on page 168](#) shows the display profile definition for the Calendar channel.

Code Example 5-32 Display Profile Definition for Calendar Channel

```
<Channel name="Calendar" provider="CalendarProvider">
  <Properties/>
</Channel>
```

Properties can be configured from the administration console. The following is a list of user editable properties in the calendar channel. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

calendar	Specifies the calendar to display.
view	Specifies the view (day, week, or month) used.

For more information on the other Calendar channel properties, see [“CalendarProvider.”](#)

Discussion Channels

The DiscussionLite channel and the Discussions channel are based on the DiscussionProvider. Similar to the search channel JSPs, they have a query portion, a display portion, and use Desktop themes. Discussions are tied to topics and specific documents.

The discussion channels include discussion threads, starting discussions based on documents or new topics, searching discussions, subscribing to discussions, and rating discussions. By default, the channels are available on the sample portal for anonymous users. However, an anonymous user cannot subscribe to a discussion or edit the channels.

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, start a new discussion which target the Discussions Channel. By default, all links are brought up in a JSPDynamicSingleContainer.

[Code Example 5-33](#) shows the display profile definition for the DiscussionLite channel.

Code Example 5-33 DiscussionLite Channel Display Profile

```
<Channel name="DiscussionLite" provider="DiscussionProvider">
  <Properties>
    <String name="title" value="Recent Discussions"/>
    <String name="description" value="This is a DiscussionLite provider example"/>
    <String name="contentPage" value="discussionLiteContent.jsp"/>
    <String name="editPage" value="" />
    <String name="processPage" value="" />
    <String name="width" value="thin"/>
    <String name="searchServer" value="" />
    <String name="db" value="discussion"/>
    <Integer name="viewHits" value="20"/>
  </Properties>
</Channel>
```

Properties can be configured from the administration console. By default, there are no user editable properties for this channel. For more information on all the DiscussionLite channel properties, see [“DiscussionProvider.”](#)

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.

[Code Example 5-34](#) shows the display profile definition for the DiscussionLite channel.

Code Example 5-34 Discussions Channel Display Profile

```

<Channel name="Discussions" provider="DiscussionProvider">
  <Properties>
    <String name="title" value="Discussions"/>
    <String name="description" value="This is a Discussion provider example" />
    <String name="searchServer" value="" />
    <String name="dbname" value="discussion"/>
    <Integer name="viewHits" value="8"/>
  </Properties>
</Channel>

```

By default, Discussions channel inherits most property values from the provider definition. Properties can be configured from the administration console. The following properties can be configured by the user. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

viewHits	Specifies the number of discussions to display on the main discussion page.
defaultFilter	Specifies the filter for searching and displaying discussions and this controls display of the subtree.
defaultExpansionThreshold	It can be set to expand all or collapse all.
viewDiscussionWindow	If set to <code>true</code> , the discussion link gets displayed on an entire page; that is, <code>JSPDynamicSingleContainer</code> is invoked. If set to <code>false</code> , the discussion gets displayed within the channel within the tab.
showDescription	Specifies whether or not to show a description of the discussion.

For more information on all the Discussions channel properties, see [“DiscussionProvider.”](#)

Instant Messaging Channel

The Instant Messaging channel displays non-offline users from a contact list in the channel and allows instant messaging conference to be started by clicking on entry.

[Code Example 5-35](#) shows the display profile definition for the Instant Messaging channel.

Code Example 5-35 Display Profile Definition for IMChannel

```
<Channel name="IMChannel" provider="IMProvider">
  <Properties/>
</Channel>
```

Properties can be configured from the administration console. The following properties can be configured by the user. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

clientRunMode	Specifies how the Instant Messaging server client must be run.
server	Specifies the name of the instant messaging server to use.
port	Specifies the port on which the instant messaging server listens.
mux	Specifies the name of instant messaging multiplexor to use (used by Instant Messaging client.)
muxport	Specifies the multiplexor port on which the instant messaging server listens.

For more information on all the Instant Messaging channel properties, see [“IMProvider.”](#)

JSP Channel

A JSP channel uses JSP pages rather than Desktop templates to define the channel. You can therefore make changes to the channel and the provider without having to restart the server.

[Code Example 5-36](#) shows the display profile definition for the JSP channel.

Code Example 5-36 Display Profile Definition for the JSP Channel

```
<Channel name="SampleJSP" provider="JSPProvider">
  <Properties>
    <String name="title" value="Sample JSP Provider"/>
    <String name="description" value="This is the sample for the Jsp Provider"/>
    <ConditionalProperties condition="locale" value="en">
      <ConditionalProperties condition="locale" value="US">
        <String name="title" value="Sample JSP Channel"/>
        <String name="description" value="This is the sample for the Jsp Provider"
      />
    </ConditionalProperties>
  </ConditionalProperties>
  </Properties>
  <Boolean name="isEditable" value="true"/>
</Channel>
```

Properties can be configured from the administration console. By default, there are no user editable properties for this provider. For more information on all the JSP channel properties, see [“JSPProvider.”](#)

Login Channel

The login channel shows up in the anonymous user's Desktop, enabling users to log in. You can select one out of the three methods to enable users to log in: LDAP, Membership, or Unix.

[Code Example 5-37](#) shows the display profile definition for the login channel.

Code Example 5-37 Display Profile Definition for the Login Channel

```
<Channel name="Login" provider="LoginProvider">
  <Properties>
    <String name="refreshTime" value="600" advanced="true"/>
    <String name="title" value="Login "/>
  </Properties>
</Channel>
```

Code Example 5-37 Display Profile Definition for the Login Channel (*Continued*)

```

    <String name="description" value="Login Information"/>
    <String name="fontFace1" value="Arial"/>
    <String name="width" value="thin" advanced="true"/>
  </Properties>
</Channel>

```

Properties can be configured from the administration console. By default, there are no user editable properties for this channel. For more information on all the login channel properties, see [“LoginProvider.”](#)

Lotus Notes Address Book Channel

The Lotus Notes Address Book channel displays entries from the users Lotus Notes Server address book functionality.

[Code Example 5-38](#) shows the display profile definition for the Lotus Notes Address Book channel.

Code Example 5-38 Display Profile Definition for LotusNotesAddressBook Channel

```

<Channel name="LotusNotesAddressBook" provider="LotusNotesAddressBookProvider">
  <Properties/>
</Channel>

```

Properties can be configured from the administration console. The following LotusNotesAddressBook channel properties can be configured by the user. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

displayEntries	Specifies if the entries should be shown.
numEntries	Specifies the number of entries to display.
sunPortalABSorderBy	Specifies the value of the entries displayed to sort by.
sunPortalABSorderByOrder	Specifies the sort order of the entries displayed.

For more information on all the Lotus Notes Address Book channel properties, see [“LotusNotesAddressBookProvider.”](#)

Lotus Notes Calendar Channel

The Lotus Notes Calendar channel displays events and task from the Lotus Notes Server calendar functionality.

[Code Example 5-39](#) shows the display profile definition for the Lotus Notes Calendar channel.

Code Example 5-39 Display Profile Definition for LotusNotesCalendar Channel

```
<Channel name="LotusNotesCalendar" provider="LotusNotesCalendarProvider">  
  <Properties/>  
</Channel>
```

Properties can be configured from the administration console. The following is a list of user editable properties in the LotusNotesCalendar channel. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

calendar	Specifies the calendar to display.
view	Specifies the view (day, week, or month) used.

For more information on all the Lotus Notes Calendar channel properties, see [“LotusNotesCalendarProvider.”](#)

Lotus Notes Mail Channel

The Lotus Notes Mail channel displays messages from the Lotus Notes Server mail functionality.

[Code Example 5-40 on page 175](#) shows the display profile definition for the Lotus Notes Mail channel.

Code Example 5-40 Display Profile Definition for LotusNotesMail Channel

```
<Channel name="LotusNotesMail" provider="LotusNotesMailProvider">
  <Properties/>
</Channel>
```

Properties can be configured from the administration console. The following is a list of user editable properties in the LotusNotesMail channel. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

numberHeaders	Specifies the limit of message headers to display. Hard limit is 30.
displayHeaders	Specifies if the headers should be shown.
sentFolderCopy	Specifies if sent messages should be copied to the Sent Folder.
sortOrder	Specifies the sort order for the messages currently displayed.

For more information on all the Lotus Notes Mail channel properties, see [“LotusNotesMailProvider.”](#)

Mail Channel

The Mail channel displays messages from the Sun One Messaging Server.

[Code Example 5-41](#) shows the display profile definition for the Mail channel.

Code Example 5-41 Display Profile Definition for Mail Channel

```
<Channel name="Mail" provider="MailProvider">
  <Properties/>
</Channel>
```

Properties can be configured from the administration console. The following is a list of user editable properties in the Mail channel. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

numberHeaders	Specifies the limit of message headers to display. Hard limit is 30.
displayHeaders	Specifies if the headers should be shown.
sentFolderCopy	Specifies if sent messages should be copied to the Sent Folder.
sortOrder	Specifies the sort order for the messages currently displayed.

For more information on all the Mail channel properties, see [“MailProvider.”](#)

Mail Check Channel

The mail check channel gives information about a user’s mail status.

[Code Example 5-42](#) shows the display profile definition for the mail check channel.

Code Example 5-42 Display Profile Definition for the Mail Check Channel

```
<Channel name="MailCheck" provider="MailCheckProvider">
  <Properties>
    <String name="refreshTime" value="120"/>
    <String name="Title" value="My Unread Mail Count"/>
    <String name="IMAPServerName" value=""/>
    <String name="IMAPUserId" value=""/>
    <String name="IMAPPassword" value=""/>
  </Properties>
</Channel>
```

The following MailCheck channel properties can be configured by the user. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

IMAPServerName	Specifies the IMAP server name.
IMAPUserId	Specifies the IMAP user name.
IMAPPassword	Specifies the IMAP password.

These user specified properties overwrite the corresponding properties in MailCheckProvider. For more information on the other MailCheck channel properties, see “[MailCheckProvider](#).”

MS Exchange Address Book Channel

The MS Exchange Address Book channel displays entries from the users MS Exchange server address book functionality.

[Code Example 5-43](#) shows the display profile definition for the MS Exchange Address Book channel.

Code Example 5-43 Display Profile Definition for MExchangeAddressBook Channel

```
<Channel name="MExchangeAddressBook" provider="MExchangeAddressBookProvider">
  <Properties/>
</Channel>
```

Properties can be configured from the administration console. The following MExchangeAddressBook channel properties can be configured by the user. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

displayEntries	Specifies if the entries should be shown.
numEntries	Specifies the number of entries to display.
sunPortalABSortBy	Specifies the value of the entries displayed to sort by.
sunPortalABSortOrder	Specifies the sort order of the entries displayed.

For more information on all the MS Exchange Address Book channel properties, see [“MSEExchangeAddressBookProvider.”](#)

MS Exchange Calendar Channel

The MS Exchange Calendar channel displays events and task from the MS Exchange server calendar functionality.

[Code Example 5-44](#) shows the display profile definition for the MS Exchange Calendar channel.

Code Example 5-44 Display Profile Definition for MSEExchangeCalendar Channel

```
<Channel name="MSEExchangeCalendar" provider="MSEExchangeCalendarProvider">
  <Properties/>
</Channel>
```

Properties can be configured from the administration console. The following is a list of user editable properties in the MSEExchangeCalendar channel. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

calendar	Specifies the calendar to display.
view	Specifies the view (day, week, or month) used.

For more information on all the MS Exchange Calendar channel properties, see [“MSEExchangeCalendarProvider.”](#)

MS Exchange Mail Channel

The MS Exchange Mail channel displays messages from the MS Exchange server mail functionality.

[Code Example 5-45 on page 179](#) shows the display profile definition for the MS Exchange Mail channel.

Code Example 5-45 Display Profile Definition for MExchangeMail Channel

```
<Channel name="MExchangeMail" provider="MExchangeMailProvider">
  <Properties/>
</Channel>
```

Properties can be configured from the administration console. The following is a list of user editable properties in the MExchangeMail channel. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

numberHeaders	Specifies the limit of message headers to display. Hard limit is 30.
displayHeaders	Specifies if the headers should be shown.
sentFolderCopy	Specifies if sent messages should be copied to the Sent Folder.
sortOrder	Specifies the sort order for the messages currently displayed.

For more information on all the MS Exchange Mail channel properties, see [“MExchangeMailProvider.”](#)

Notes Channel

The notes channel enables users to post messages to other users' notes channel on their Desktop.

[Code Example 5-46](#) shows the display profile definition for the notes channel.

Code Example 5-46 Display Profile Definition for the Notes Channel

```
<Channel name="Notes" provider="NotesProvider">
  <Properties>
    <String name="refreshTime" value="600"/>
    <String name="title" value="Notes"/>
    <String name="description" value="Notes Channel Description"/>
  </Properties>
</Channel>
```

Code Example 5-46 Display Profile Definition for the Notes Channel (*Continued*)

```

    <String name="lines" value="4"/>
    <String name="maxLines" value="4"/>
  </Properties>
</Channel>

```

The following is a list of the properties specific to the notes channel. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

lines	4, which starts the channel with four lines of notes.
maxLines	4, which provides for a maximum of four lines of notes.

These channel-defined properties overwrite the corresponding properties in `NotesProvider`. For more information on the other Notes channel properties, see [“NotesProvider.”](#)

The notes themselves are stored and read in a text file in the following format:

```
userid | date | message
```

where | is the delimiter and date is long value which denotes the time elapsed in milliseconds since Jan. 1, 1970. Here is a sample of what may be included in the `notes.txt` file:

```
User1|1007159465858|Message to Portal Desktop Team : Lets meet today at 2PM
User2|1007159465858|Information related to the project is availble at home
page
```

Search Channel

The search channel can provide three interfaces:

- **Basic search** - Enables users to type keywords in a search box. Document and category matches are then displayed.

- **Advanced search** - Enables users to search for documents based on author, title, and URL. Users can also search on the last-modified date of a document. Advanced search is a more complex user interface, and supports customization.
- **Browse** - Enables users to browse the category tree and search within categories.

[Code Example 5-47](#) shows the display profile definition for the search channel.

Code Example 5-47 Display Profile Definition for the Search Channel

```
<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>
```

Properties can be configured from the administration console. The following is a list of user editable properties in the Search channel. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

windowPref	Specifies how to launch a link.
viewHits	Specifies the number of hits that should be displayed per page.
categorySearch	Specifies whether or not to display the category search by default.
defaultMode	Specifies the default search mode.

For more information on all the Search channel properties, see [“SearchProvider.”](#)

Sample Simple Web Service Channel

The Simple Web Service channel makes simple web services available to an end user channel. This pre-configured channel is always bound to a single web service and associated method by using channel initialization properties. This channel type enables end users to cache (or store) default values for the web service input parameters by using the Edit button functionality on the channel.

Code Example 5-48 shows the display profile definition for a sample simple web service channel.

Code Example 5-48 Display Profile Definition for SampleSimpleWebService Channel

```
<Channel name="SampleSimpleWebService" provider="SimpleWebServiceProvider">
  <Properties>
    <String name="title" value="Pre-Configured Web Service Channel"/>
    <String name="description" value="This is a sample pre-configured Simple Web
Service Channel"/>
    <String name="wsdlURL"
value="file:///etc/opt/SUNWps/desktop/default/SampleSimpleWebService/CurrencyExchangeService.wsdl"/>
    <String name="editPage" value="webserviceInputEdit.jsp"/>
    <String name="methodName" value="getRate"/>
    <Boolean name="isDefaultShowOutput" value="false"/>
    <Boolean name="isDefaultAvailable" value="true"/>
    <Collection name="defaultInput">
      <String name="country1" value="usa"/>
      <String name="country2" value="japan"/>
    </Collection>
  </Properties>
</Channel>
```

Properties can be configured from the administration console. By default, there are no user editable properties for this channel. For more information on all the SampleSimpleWebService channel properties, see [“SimpleWebServiceProvider.”](#)

Sample Simple Web Service Configurable Channel

The Simple Web Service Configurable channel is similar to the Simple Web Service channel, except that it permits users to use the Edit function to change URLs and methods, hence, it is configurable. This channel does not enable end users to cache (or store) default values for the web service input parameters.

Code Example 5-49 shows the display profile definition for a sample simple web service configurable channel.

Code Example 5-49 Display Profile Definition for SampleSimpleWebServiceConfigurable Channel

```
<Channel name="SampleSimpleWebServiceConfigurable"
provider="SimpleWebServiceConfigurableProvider">
  <Properties>
    <String name="title" value="Sample Configurable Simple Web Service Channel"/>
    <String name="description" value="This is a sample configurable Simple Web Service
Channel"/>
    <String name="wsdlURL"
value="file:///etc/opt/SUNWps/desktop/default/SampleSimpleWebServiceConfigurable/CurrencyEx
changeService.wsdl"/>
    <String name="methodName" value="getRate"/>
  </Properties>
</Channel>
```

Properties can be configured from the administration console. The following is a list of user editable properties in the SimpleWebServiceConfigurable channel. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

String name="wsdlURL"	Specifies the URL to web service WSDL.
String name="methodName"	Specifies the web service method name that is going to be executed.

For more information on all the SampleSimpleWebServiceConfigurable channel properties, see “[SimpleWebServiceConfigurableProvider](#).”

Subscriptions Channel

The subscriptions channel shows subscriptions by types which can be category subscriptions, discussion subscriptions, and saved searches. For each type of subscription, the following is displayed:

- The subscription label.

- A link to the subscription detail representing the number of hits for that particular subscription.

Code Example 5-50 shows the display profile definition for the subscriptions channel.

Code Example 5-50 Subscriptions Channel Display Profile

```
<Channel name="Subscriptions" provider="SubscriptionsProvider">
  <Properties>
    <String name="title" value="Subscriptions"/>
    <String name="description" value="This is a Subscriptions provider example" />
  </Properties>
</Channel>
```

Sample RSS Channel

The RSS channel uses an XSLT (style sheet) file to transform the RSS XML file into HTML. By using different XSL files the same output can be changed and presented differently for different channels.

Code Example 5-51 shows the display profile definition for the RSS channel.

Code Example 5-51 Display Profile Definition for SampleRSS Channel

```
<Channel name="SampleRSS" provider="XMLProvider">
  <Properties>
    <String name="title" value="Sun ONE Information"/>
    <String name="description" value="Company News"/>
    <Integer name="timeout" value="100"/>
    <String name="width" value="thin" advanced="true"/>
    <Boolean name="isEditable" value="false" advanced="true"/>
    <ConditionalProperties condition="locale" value="en" >
      <String name="helpURL" value="en/desktop/rsschann.htm" advanced="true"/>
    </ConditionalProperties>
    <String name="helpURL" value="en/desktop/rsschann.htm" advanced="true"/>
    <String name="fontFace1" value="Sans-serif"/>
    <String name="productName" value="Sun ONE Portal Server"/>
    <String name="refreshTime" value="600" advanced="true"/>
    <String name="url"
value="file:///etc/opt/SUNWps/desktop/default/SampleRSS/SampleRSS.rss"/>
    <String name="xslFileName" value="rss.xsl"/>
  </Properties>
</Channel>
```

The RSS channel sets many more properties than most channels, due to the fact that there is no RSS provider and the channel uses XMLProvider. Because of this, the RSS channel needs to override most of the properties set by XMLProvider. See [“XMLProvider”](#) for more information.

You cannot set `isEditable` to true for the RSS channel, as the channel is not editable and there are no `getEdit()` and `processEdit()` methods defined.

User Information Channel

The user information channel displays a greeting, the user’s name, time zone, locale and has access to the user’s IMAP and SMTP data. The user information channel enables users to view and edit user information.

[Code Example 5-52](#) shows the display profile definition for the user information channel.

Code Example 5-52 Display Profile Definition for UserInfo Channel

```
<Channel name="UserInfo" provider="UserInfoProvider">
  <Properties>
    <String name="refreshTime" value="60" advanced="true"/>
    <String name="title" value="User Information"/>
    <String name="description" value="View/Edit User Information"/>
  </Properties>
</Channel>
```

In this example, only `refreshTime`, `title`, and `description` properties are defined. Properties can be configured from the administration console. The following is a list of default user editable properties in the UserInfo channel. This two column table lists the properties in the first (left) column and a brief description in the second (right) column.

<code>cn</code>	Specifies the tag name for greeting.
<code>greeting</code>	Specifies the tag name for common name.
<code>timezone</code>	Specifies the tag module to get and set time zone list.
<code>preferredlocale</code>	Specifies the user’s preferred locale.
<code>sunPortalNetmailIMAPServerName</code>	Specifies the IMAP server name.

sunPortalNetmailSMTPServerName	Specifies the SMTP server name.
sunPortalNetmailIMAPUserid	Specifies the IMAP user ID.
sunPortalNetmailIMAPPASSWORD	Specifies the IMAP password.
membershipOriginalPassword	Specifies the tag name for the original membership password.
membershipNewPassword	Specifies the tag name for membership new password.
membershipConfirmPassword	Specifies the tag name for membership confirm password.

For more information on all the UserInfo channel properties, see [“UserInfoProvider.”](#)

Sample URL Scraper Channel

The URL scraper channel takes a URL, opens a connection to the URL, and reads the contents into a buffer. The contents are then sent to the Desktop Servlet, which displays it. URLScrapperProvider uses the Rewriter to construct the URL information as long as the MIME type of the file is one that the Rewriter supports. (Rewriter rules are expressed using an XML document but the URL scraper channel does not use an XML file to describe how to display it.)

Code Example 5-53 shows the display profile definition for a sample URL scraper channel.

Code Example 5-53 Display Profile Definition for SampleURLScrapper Channel

```
<Channel name="SampleURLScrapper" provider="URLScrapperProvider">
  <Properties>
    <String name="refreshTime" value="600" advanced="true"/>
    <String name="url"
value="file:///etc/opt/SUNWps/desktop/default/SampleURLScrapper/ipinfo.html"/>
    <String name="title" value="URL Scraper Channel"/>
  </Properties>
</Channel>
```

Properties can be configured from the administration console. By default, there are no user editable properties for this channel. For more information on all the URLScrapper channel properties, see “[URLScrapperProvider](#).”

Creating a URL scraper channel is a quick and simple way to add content to the Desktop, as no coding is required. However, the following scenarios are inappropriate for the URL scraper channel:

- If the site being scraped requires HTTP basic authentication.
- If an edit function is required so that users can customize the channel.
- If the data comes from a non-HTML, non-web server source, for example, a database or mail server.
- If the data needs to be reformatted in some way for the channel.
- If a more efficient solution is required. The URL scraper channel does a request and look up for every Desktop display and user.

For situations similar to these, use the PAPI. See the *Sun ONE Portal Server 6.2 Developer’s Guide* for more information.

Sample XML Channel

The XML channel provides a flexible way to include XML content in the Desktop, enabling an XML document to be transformed into HTML using an XSL (style sheet) file.

[Code Example 5-54](#) shows the display profile definition for the sample XML channel.

Code Example 5-54 Display Profile Definition for SampleXML Channel

```
<Channel name="SampleXML" provider="XMLProvider">
  <Properties>
    <String name="refreshTime" value="600" advanced="true"/>
    <String name="title" value="XML Test Channel"/>
    <String name="description" value="This is a test of the XML Provider system"/>
    <String name="url"
value="file:///etc/opt/SUNWps/desktop/sampleportal/SampleXML/getQuotes.xml"/>
    <String name="xslFileName" value="stockquote.xsl"/>
  </Properties>
</Channel>
```

Properties can be configured from the administration console. By default, there are no user editable properties for this channel. For more information on all the SampleXML channel properties, see [“XMLProvider.”](#)

Understanding the Sample Portal

This chapter describes the sample portal that you can choose to install on your system during the Sun™ ONE Portal Server installation. See the *Sun ONE Portal Server 6.2 Installation Guide* for more information on installing the sample portal.

This chapter assumes that the sample portal is installed. It contains the following sections:

- [Overview of the Sample Portal](#)
- [JSP-based Tab Container \(JSPTabContainer\)](#)
- [JSP-based Table Container \(JSPTableContainer\)](#)
- [Frame-based Container \(FrameTabContainer\)](#)
- [Template-based Containers \(TemplateTabContainer and TemplateTableContainer\)](#)

Overview of the Sample Portal

The sample portal consist of the following elements:

- Sample channels and containers (discussed in this chapter)
- Sample Portal templates and JSPs (see [Appendix A, “Desktop Template and Tag Reference”](#) and [Appendix C, “JavaServer Pages Reference”](#))
- Authless user definition and display profile XML fragment
- Authless templates and JSPs (see [Appendix A, “Desktop Template and Tag Reference”](#) and [Appendix C, “JavaServer Pages Reference”](#))

Sample Containers for the Sample Desktop

This chapter discusses the sample portal sample containers, each of which produces a different looking Desktop. A container is a channel that is able to hold and organize content from other channels. The sample portal is created from the following containers:

JSPTabContainer Creates a Desktop that contains multiple containers selected using different tabs. Normally, each tab is constructed by using the corresponding `PredefinedTabPanelContainer`. This container is JSP-based. Its provider is `JSPTabContainerProvider`.

JSPTableContainer Creates a Desktop that arranges a maximum of five sub-containers into the channel arrangement. This container is JSP-based. Its provider is `JSPTableContainerProvider`.

FrameTabContainer Creates a Desktop using frames. The left-hand frame enables you to navigate, and the right-hand frame displays the channels. This container is JSP-based. Its provider is `JSPTabContainerProvider`.

TemplateTabContainer Creates a Desktop that contains multiple containers selected using different tabs. Normally, each tab is constructed by using the corresponding `PredefinedTemplatePanelContainer`. This container is template-based. Its provider is `TemplateTabContainerProvider`.

TemplateTableContainer Creates a Desktop that arranges a maximum of five sub-containers into the channel arrangement. This container is template-based. Its provider is `TemplateTableContainerProvider`.

When you install the sample portal, 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.

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

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.

For more information on each container, see the appropriate section in this chapter.

Guidelines for Using the Sample Portal

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 lose any changes you might have made to the sample portal files, as the patch would only overwrite the initially installed sample files.

You should also create a custom Desktop type for your users. For example, if the Desktop type were `sesta`, then you would create a `/etc/opt/SUNWps/desktop/sesta` directory to contain the template files that you have customized. 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. Also, in the administration console, specify `sesta,sampleportal` for the `sunPortalDesktopType` attribute.

To re-load the default (original) display profile for the sample portal providers, the `dp-providers.xml` and `dp-org.xml` files (in `portal-server-install-root/SUNWps/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:

```
portal-server-install-root/SUNWps/bin/dpadmin modify -u
"uid=amAdmin,ou=People,dc=sesta,dc=com" -w password -g
portal-server-install-root/SUNWps/samples/desktop/dp-providers.xml

portal-server-install-root/SUNWps/bin/dpadmin modify -u
"uid=amAdmin,ou=People,dc=sesta,dc=com" -w password -d "dc=sesta,dc=com"
portal-server-install-root/SUNWps/samples/desktop/dp-org.xml
```

Sample Portal Installation Directories

If you choose to install the sample portal, the installer locates the appropriate files in the following directories:

portal-server-install-root/SUNWps/samples/desktop

This directory contains the following display profile documents:

<code>dp-org.xml</code>	Contains the display profile definitions for channels and containers.
-------------------------	---

`dp-anon.xml`

Contains the display profile definitions for channels and containers for the authlessanonymous and anonymous users in the default organization.

`/etc/opt/SUNWps/desktop/sampleportal`

Contains the JSP, template, and other support files for the Sun ONE Portal Server software sample portal.

`/etc/opt/SUNWps/desktop/anonymous`

Contains the JSP, template, and other support files for the Sun ONE Portal Server software Desktop anonymous user.

Desktop Theme Link

For the template-based containers, when you choose the Theme link, the look and feel of the tool bar changes slightly. (This is not the case for the JSP-based containers.) This happens because the template-based containers use `JSPSingleContainer` (which is a JSP-based container), and not a template-based container. Thus, the look and feel of the Theme page is “JSP,” not “template.”

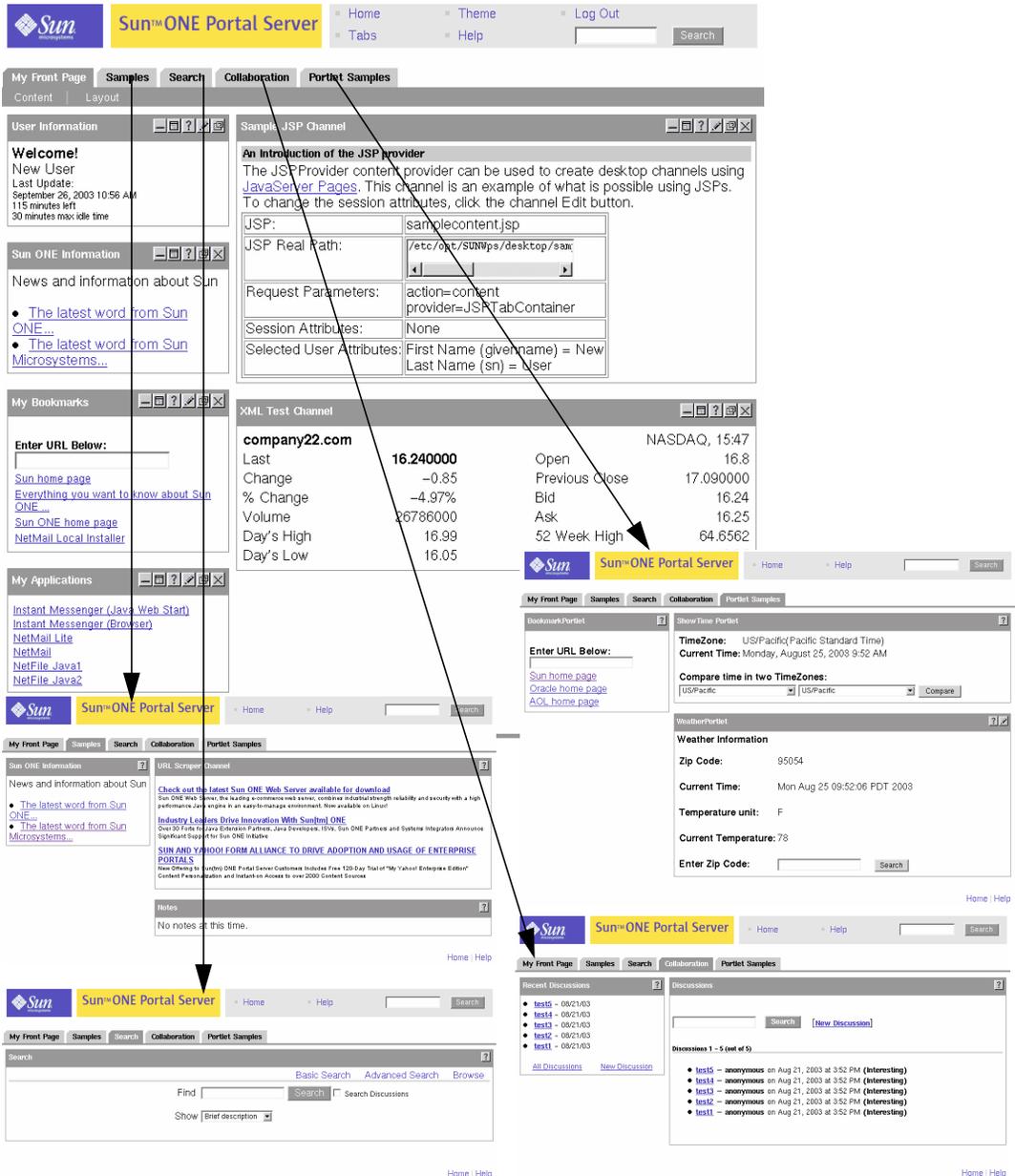
Also, not all of the theme attribute values are applied to the template-based Desktops. For example, the header of the selected theme is not used in the template-based Desktops. The template-based desktops only use those attributes that are supported in Sun ONE Portal Server 6.0 software.

JSP-based Tab Container (JSPTabContainer)

[Figure 6-1 on page 193](#) shows the JSP-based tab Desktop, `JSPTabContainer`, which provides a JSP-based tab container consisting of five tabs, My Front Page, Samples, Search, Collaboration, and Sample Portlet which includes the following channels:

- In the My Front Page tab: Login or User Information, Sun ONE Information, Bookmark, Sample JSP, and XML Test channels
- In the Samples tab: Sun ONE Information, URL Scraper, and Notes channels
- In the Search tab: Search channel
- In the Collaboration tab: Discussions Lite and Discussions channels
- In the Portlet Samples tab: Bookmark, Showtime, and Weather portlet channels

Figure 6-1 JSP Tab-Based Desktop (JSPTabContainer)



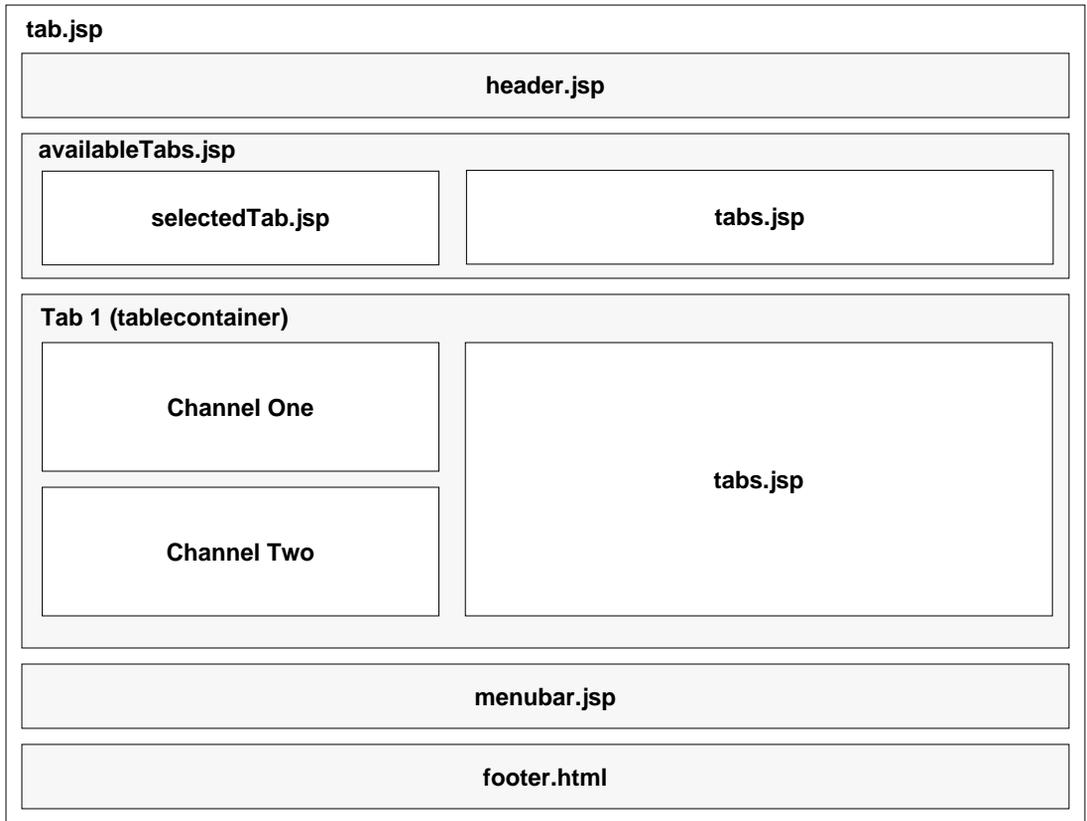
JSPTabContainer Architecture

Figure 6-2 shows the JSPTabContainer architecture. In this figure, `tab.jsp` is the top-level JSP file. The `tab.jsp` file makes include calls to the `header.jsp`, `availableTabs.jsp`, `menubar.jsp`, and `footer.html` files. The `availableTabs.jsp` file makes an include call to the `selectedTab.jsp` and `tabs.jsp` files.

See [Appendix C, “JavaServer Pages Reference”](#) more information on these JSP files.

JSPTabContainer, by default, is made up of five sub-containers—[MyFrontPageTabPanelContainer](#), [SamplesTabPanelContainer](#), [SearchTabContainer](#), [CollaborationTabPanelContainer](#), and [JSPTabContainer/PortletSamplesTabPanelContainer](#)—as represented by Tab1 (tablecontainer) in the figure.

Figure 6-2 JSPTabContainer Architecture



JSPTabContainer Actions

This section describes the various actions that you can perform within JSPTabContainer.

Banner Links

- **Home** - Returns you to the Desktop Home page from customization pages and updates content.
- **Tabs** - Lets you remove, rename, or select the start tab, and also create a new tab. The link for this page is:

`action=edit&provider=JSPTabContainer`

- **Theme** - Lets you edit the Desktop theme. The theme enables you to set the color scheme and font type for the Desktop. You can choose from preset themes or make your own colors for a custom theme. The link for this page is:

```
action=edit&provider=JSPPresetThemeContainer
```

The custom theme channel is defined as a `JSPSingleContainerProvider` channel, and has an entry in the display profile. See [“JSPSingleContainerProvider” on page 50](#) for more information.

- **Help** - Displays the Desktop sample online help. The link for this page is:

```
../docs/locale/desktop/helppage.htm
```

- **Log Out** - Logs you out from the Desktop. The link for this page is:

```
action=logout
```

- **Search** - A short cut to allow you to search. You can also select the Search tab to do a search.

Links Specific to the Contained Container

The channels in each tab depend on the contained container of the `JSPTabContainer`. In the Sample Portal, these contained containers are `JSPTableContainer` and the channels are dependant on this container; but this does not have to be the case, they can be any container. The Content and Layout links provide the ability to customize the current selected contained container.

Content and Layout Links

The top-most JSP in the table container defines the Content and Layout links. `JSPContentContainer` is the container that displays the Content page, and `JSPLayoutContainer` is the container that displays the Layout page. The following describes the actions for these pages:

- **Content** - Lets you edit the content for this particular tab on the Content page. The link for this page, when the My Front Page tab is selected, is:

```
action=edit&provider=JSPContentContainer&container=MyFrontPageTabPanelContainer&selected=My+Front+Page
```

- **Layout** - Lets you edit the layout of the channels for this particular tab on the Layout page. The link for this page, when the My Front Page tab is selected, is:

```
action=edit&provider=JSPLayoutContainer&container=MyFrontPageTabPanelContainer&selected=My+Front+Page
```

JSPTabContainer Edit Pages

When you click the Tabs link in the Desktop, the Current Tabs Settings Edit page, where you can make changes, is displayed.

- Start Tab - Lets you set the starting tab.
- Tab Name - Specifies the name of the tabs in the container.
- Action - Lets you rename or delete a tab from the Desktop. (JavaScript handles the action.)

When you click Make a New Tab, the corresponding Edit page is shown. You decide what to name the tab and what the tab topics are. Content Page is displayed only when making a new tab from scratch. When other Tab Topics are selected, a new tab which looks similar to the TabTopic selected, is created and displayed.

Container Display Profile

JSPTabContainerProvider Display Profile

The provider responsible for generating the JSPTabContainer channel is JSPTabContainerProvider. See [“JSPTabContainerProvider” on page 49](#) for the display profile XML for this provider.

JSPTabContainer Channel Display Profile

The provider profile is the template which decides the properties for a container channel, but the container channel profile will ultimately decide the values for the container channel attributes. See [“JSPTabContainer” on page 141](#) for the display profile for JSPTabContainer channel.

Container’s Display Profile Properties

The default properties that make up JSPTabContainer work as follows:

- `contentPage` is set to `tab.jsp`. This draws the Content Page for the tab container.
- `editPage` is set to `tabedit.jsp`. This displays the Edit page for the tab container where new tabs can be added, and existing tabs removed or renamed.
- `startTab` sets the tab that opens first on the Desktop as `MyFrontPageTabPanelContainer`.

- `maxTabs` allows six tabs to be created. As there are currently five tabs, one more can be added.
- `makeTabProvider` is used when creating a new tab from scratch.
- `channelNumber` specifies that a number is appended to a newly created tab as the channel name. This number is increased each time a new tab is created, so that the new tab will have unique name. For example, to create a new tab based on `MyFrontPageTabPanelContainer` in `JSPTabContainer`, the new tab channel name would be `JSPTabContainer/MyFrontPageTabPanelContainer1`. (The new tab name is actually the `channelName` property in the display profile plus the the value of `channelNumber`. `channelNumber` is incremented by one each time a new tab is created.)
- `contentChannel` specifies `JSPContentContainer` as the content channel that provides the Content page displaying channels to add to a user-created tab.
- `presetThemeChannel` specifies `JSPPresetThemeContainer` as the channel that is displayed in the Theme - Preset Theme page.
- `customThemeChannel` specifies `JSPCustomThemeContainer` is the channel that displayed in the Theme - Custom Theme page.
- The collection `TabProperties` has `<Collection name=>` entries for each of the available tab defined in `JSPTabContainer`.
- The `Available` list describes all available channels for this container. The available channels are displayed in the Content Preference page for users to select from.
- The `Selected` list describes selected channels for this container. Only selected channels shows up in the Desktop.

Contained Containers' Display Profile

The JSP tab container available channel list has [MyFrontPageTabPanelContainer](#), [SamplesTabPanelContainer](#), [SearchTabPanelContainer](#), [PredefinedSamplesTabPanelContainer](#), and [PredefinedFrontPageTabPanelContainer](#) as its contained channels.

JSP Files Used by JSPTabContainer

Sun ONE Portal Server software uses JSP files for a channel's presentation layer. `JSPTabContainer` references two main JSPs, `tab.jsp` and `tabedit.jsp`, through the `contentPage` and `editPage` properties.

Content template is responsible for the front page of the container channel and the file name for the tab container channel is `tab.jsp`. The `tab.jsp` file extensively uses the Desktop taglibs. For more information on taglibs, see [Appendix B, “JavaServer Pages Tag Library Reference.”](#)

The Edit page is where you can add, remove, and rename tabs. The `tabedit.jsp` is used to display this page. For a description of all JSP files used by JSPTabContainer, see [Appendix C, “JavaServer Pages Reference.”](#)

JSP-based Table Container (JSPTableContainer)

[Figure 6-3](#) shows the JSP-based table Desktop, JSPTableContainer, which provides a JSP-based table container consisting of the following channels:

- Thin channels: User Information, Sun ONE Information, My Bookmarks, Mailcheck Provider, and My Applications
- Wide channels: Sample JSP, XML Test, Notes, Personal Notes, and Preconfigured Web Service

Figure 6-3 JSP Table-Based Desktop (JSPTableContainer)

The screenshot displays the Sun ONE Portal Server interface with the following components:

- Header:** Sun Microsystems logo, Sun ONE Portal Server title, Home and Help navigation links, and a search box.
- Login Panel:**
 - Member Login:** Local Login section with fields for user ID and password, and a Login button.
 - Links for "New User? Sign me up" and "Trouble signing in? Get Help".
- Sample JSP Channel Panel:**
 - An Introduction of the JSP provider:** Text explaining the JSPProvider content provider and its use for creating desktop channels.
 - Configuration Table:**

JSP:	samplecontent.jsp
JSP Real Path:	/etc/opt/SUNWps/desktop/sam
Request Parameters:	provider=JSPTableContainer
Session Attributes:	None
Selected User Attributes:	First Name (givenname) = null Last Name (sn) = default
- Sun ONE Information Panel:**
 - News and information about Sun.
 - Links to "The latest word from Sun ONE..." and "The latest word from Sun Microsystems..."
- XML Test Channel Panel:**
 - Market data for **company22.com** (NASDAQ, 15:47).

Last	16.240000	Open	16.8
Change	-0.85	Previous Close	17.090000
% Change	-4.97%	Bid	16.24
Volume	26786000	Ask	16.25
Day's High	16.99	52 Week High	64.6562
Day's Low	16.05	52 Week Low	12.85
- Bookmark Provider Panel:**
 - Enter URL Below: [input field]
 - Links: Sun home page, Everything you want to know about Sun ONE ..., Sun ONE home page
- Notes Panel:**
 - No notes at this time.

JSPTableContainer Architecture

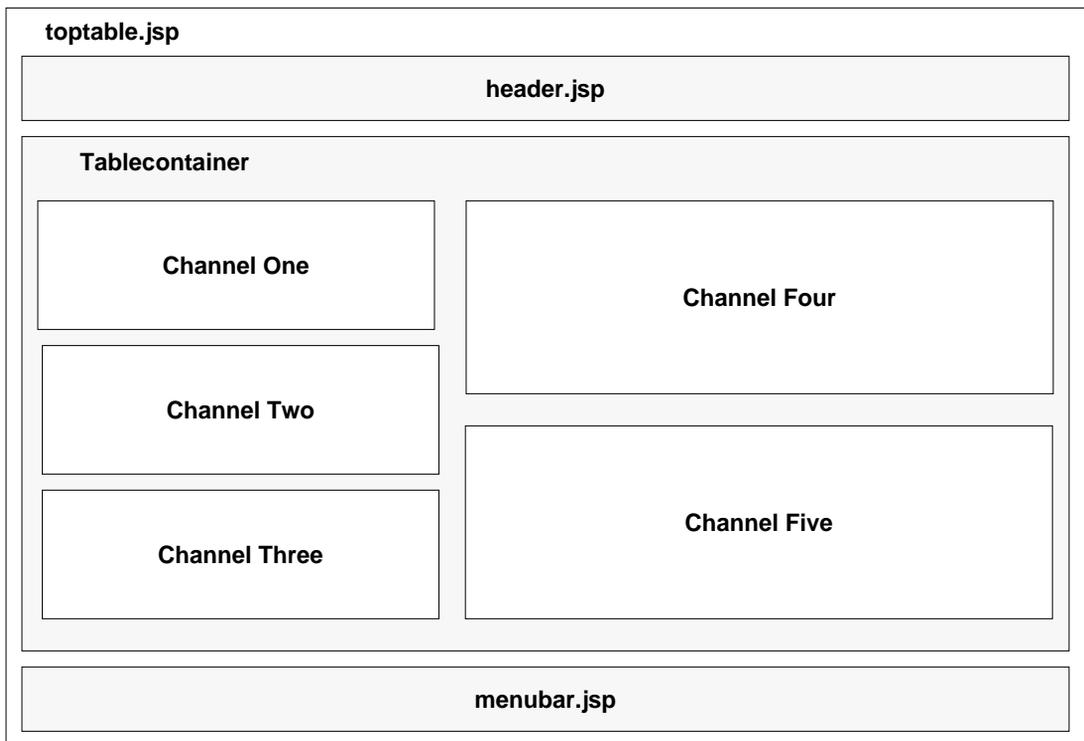
Figure 6-4 on page 201 shows the JSPTableContainer architecture. In this figure, `toptable.jsp` is the top-level JSP file. The `toptable.jsp` file makes include calls to the `header.jsp`, `launchPopup.jsp`, `leafWrapper.jsp`, and `menubar.jsp` files.

See Appendix C, "JavaServer Pages Reference" more information on these JSP files.

JSP-based table Desktop contains leaf channels that can be arranged in rows and columns. There are three kinds of column layouts: thin-wide, wide-thin, or thin-wide-thin.

The default layout for JSPTableContainer is set to 1, which gives the thin-wide columns for the table. In [Figure 6-4](#), Channels One, Two, and Three are thin channels, and Channels Four and Five are wide channels.

Figure 6-4 JSPTableContainer Architecture



JSPTableContainer Actions

This section describes the various actions that you can perform within JSPTableContainer.

Banner Links

- Home - Returns you to the Desktop Home page from customization pages and updates content.
- Theme - Lets you edit the Desktop theme. The theme enables you to set the color scheme and font type for the Desktop. You can choose from preset themes or make your own colors for a custom theme. The link for this page is:

```
action=edit&provider=JSPPresetThemeContainer
```

The custom theme channel is defined as a `JSPSingleContainerProvider` channel, and has an entry in the display profile. See “`JSPSingleContainerProvider`,” on page 52 for more information.

- Log Out - Logs you out from the Desktop. The link for this page is:

```
action=logout
```
- Help - Displays the Desktop sample online help. The link for this page is:

```
../docs/locale/desktop/helppage.htm
```
- Search - A short cut to allow you to search. You can also select the Search tab to do a search.

Links Specific to the Contained Container

`JSPTableContainer` does not contain any container containers, it only has leaf channels. This container uses `JSPContentContainer` and `JSPLayoutContainer` to edit the content and layout, respectively.

Content and Layout Links

The `toptable.jsp` file defines the Content and Layout links. `JSPContentContainer` is the container that displays the Content page, and `JSPLayoutContainer` is the container that displays the Layout page. The following describes the actions for these pages:

- Content - Lets you edit the content on the Content page. The link for this page is:

```
action=edit&provider=JSPContentContainer&container=JSPTableContainer
```

- Layout - Lets you edit the layout of the channels on the Layout page. The link for this page is:

```
action=edit&provider=JSPLayoutContainer&container=JSPTableContainer
```

Container's Display Profile

JSPTableContainerProvider Display Profile

The provider responsible for generating the JSPTableContainer channel is JSPTableContainerProvider. See [“JSPTableContainerProvider” on page 48](#) for the display profile XML for this provider.

JSPTableContainer Channel Display Profile

The provider profile is the template which decides the properties for a container channel, but the container channel profile will ultimately decide the values for the container channel attributes. See [“JSPTableContainer” on page 134](#) for the display profile for JSPTableContainer channel.

Container's Display Profile Properties

The default properties that make up JSPTableContainer work as follows:

- `contentPage` is set to `toptable.jsp`. This draws the Content Page for the table container.
- The collection `categories` defines the categories under which the available channels in the JSPTabContainer will be grouped in the Content page. Here there are three categories: `Personal Channels`, `Sample Channels`, and `News Channels`.
- The collection `channelsRow`, and values that appear in this collection, contain the row number value for channels in this container. For example, the mail check channel is defined as row 4.
- The collection `channelsIsRemovable` defines a collection to contain the `isRemovable` value for channels in this container. Only one channel, user information, is defined, with a value of `false`, so that it cannot be removed.
- The `Available` list describes all available channels for this container. The available channels are displayed in the content preference page for users to select from.
- The `Selected` list describes selected channels for this container. Only selected channels shows up in the Desktop.

JSP Files Used by JSPTableContainer

Sun ONE Portal Server uses JSP files for a channel's presentation layer. JSPTableContainer references one main JSPs, `toptable.jsp`, through the `contentPage` property.

Content template is responsible for the front page of the container channel and the file name for the tab container channel is `toptable.jsp`. The `toptable.jsp` file extensively uses the Desktop taglibs. For more information on taglibs, see [Appendix B, "JavaServer Pages Tag Library Reference."](#) For a description of all JSP files used by JSPTableContainer, see [Appendix C, "JavaServer Pages Reference."](#)

Frame-based Container (FrameTabContainer)

[Figure 6-5](#) shows the frame-based table Desktop, FrameTabContainer, which provides a frame-based table container consisting of two frames, My Front Page and Samples, with the following channels:

- In the My Front Page frame: User Information, Sun ONE Information, My Bookmarks, Mailcheck Provider, My Applications, Sample JSP, and XML Test
- In the Samples frame: Sun ONE Information, URL Scraper, Notes, and Preconfigured Web Service

Figure 6-5 Frame-Based Desktop (FrameTabContainer)

The screenshot displays the Sun ONE Portal Server interface. On the left is a navigation menu with links: My Front Page, Samples, Bookmark Provider, UrlScraper Channel, *** JSP Provider ***, and *** XML Provider ***. The main content area is divided into several frames:

- Login**: A form for Member Login with fields for user ID and password, and a Login button. It also includes links for New User? Sign me up and Trouble signing in? Get Help.
- Sample JSP Channel**: Contains an introduction to the JSP provider and a table of channel attributes.

JSP:	samplecontent.jsp
JSP Real Path:	/etc/opt/SUNWps/desktop/sam
Request Parameters:	FrameTabContainer.targetframe=selectedTab action=content provider=FrameTabContainer
Session Attributes:	None
Selected User Attributes:	First Name (givenname) = null Last Name (sn) = default
- Sun ONE Information**: Provides news and information about Sun, with links to "The latest word from Sun ONE..." and "The latest word from Sun Microsystems...".
- Bookmark Provider**: A form to enter a URL, with links to "Sun home page" and "Everything you want to know about Sun ONE...".
- XML Test Channel**: Displays stock market data for company22.com.

company22.com		NASDAQ, 15:47	
Last	16.240000	Open	16.8
Change	-0.85	Previous Close	17.090000
% Change	-4.97%	Bid	16.24
Volume	26786000	Ask	16.25
Day's High	16.99	52 Week High	64.6562
Day's Low	16.05	52 Week Low	12.85

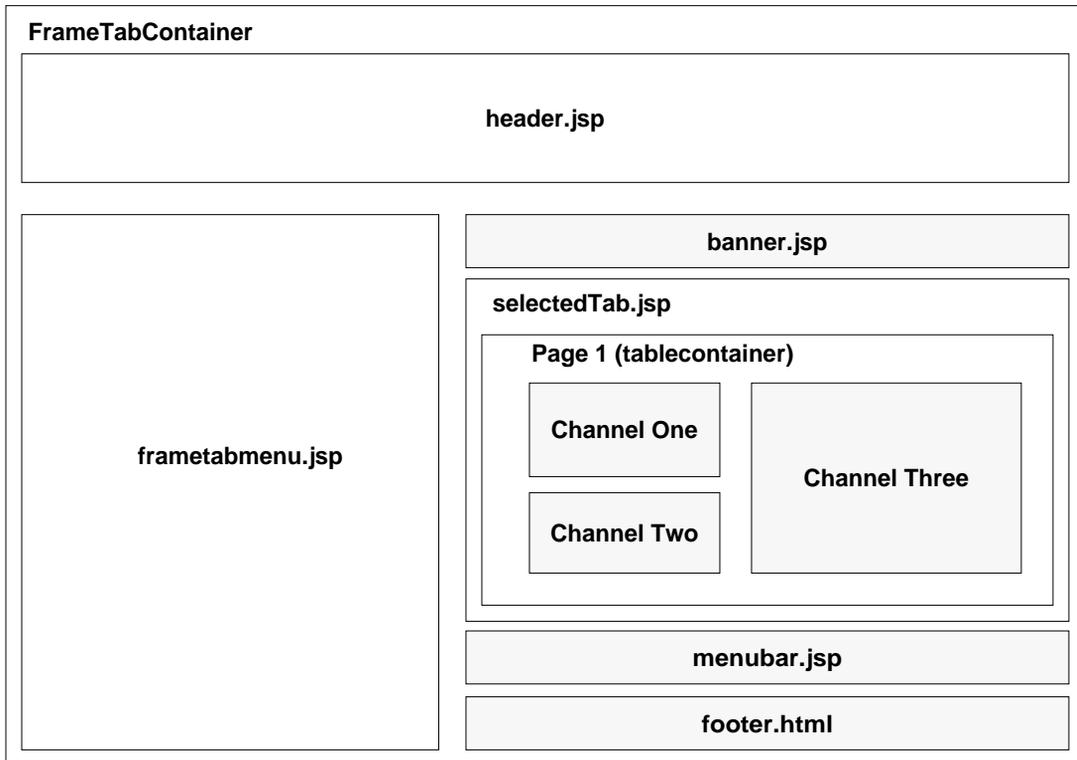
FrameTabContainer Architecture

Figure 6-6 on page 206 shows the FrameTabContainer architecture. In this figure, `frametab.jsp` is the top-level JSP file. The `frametab.jsp` file makes include calls to the `frametabmenu.jsp`, `header.jsp`, `banner.jsp`, `selectedTab.jsp`, `menubar.jsp`, `frameset.jsp`, and `footer.html` files.

See Appendix C, “JavaServer Pages Reference” more information on these JSP files.

FrameTabContainer is made up of two sub-containers, MyFrontPageFramePanelContainer and SamplesFramePanelContainer, as represented by Page 1 (tablecontainer) in the figure.

Figure 6-6 FrameTabContainer Architecture



FrameTabContainer Actions

This section describes the various actions that you can perform within FrameTabContainer.

Banner Links

- Home - Returns you to the Desktop Home page from customization pages and updates content.

- Sections - Lets you rename or select the start page, delete a page, and also create a new page. The link for this page is:

```
action=edit&provider=FrameTabContainer
```

- Theme - Lets you edit the Desktop theme. The theme enables you to set the color scheme and font type for the Desktop. You can choose from preset themes or make your own colors for a custom theme. The link for this page is:

```
action=edit&provider=JSPPresetThemeContainer
```

The custom theme channel is defined as a `JSPSingleContainerProvider` channel, and has an entry in the display profile. See [“JSPSingleContainerProvider” on page 50](#) for more information.

- Help - Displays the Desktop sample online help. The link for this page is:

```
../docs/locale/desktop/helppage.htm
```

- Log Out - Logs you out from the Desktop. The link for this page is:

```
action=logout
```

Content and Layout Links

The top-most JSP in the table container defines the Content and Layout links. `JSPContentContainer` is the container that displays the Content page, and `JSPLayoutContainer` is the container that displays the Layout page. The following describes the actions for these pages:

- Content - Lets you edit the content for this particular page on the Content page. The link for this page, when My Front Page is selected, is:

```
action=edit&provider=JSPContentContainer&container=MyFrontPageFramePanelContainer
```

- Layout - Lets you edit the layout of the channels for this particular page on the Layout page. The link for this page, when the My Front Page tab is selected, is:

```
action=edit&provider=JSPLayoutContainer&container=MyFrontPageTabPanelContainer&selected=MyFrontPageFramePanelContainer
```

FrameTabContainer Edit Pages

When you click the Sections link on the Content page, the Current Page Settings Edit page appears, where you can make changes.

- Start page - Lets you set the starting page.

- **Rename** - Lets you rename the page. (JavaScript handles the action.)
- **Delete** - Lets you delete a page from the Desktop. (JavaScript handles the action.)

When you click **Make a New Page**, the corresponding **Edit page** is shown. You decide what to name the page and what the page topics are. **Content Page** is displayed only when making a new tab from scratch. When other **Tab Topics** are selected, a new tab which looks similar to the **TabTopic** selected, is created and displayed.

Container's Display Profile

JSPTabContainerProvider Display Profile

The provider responsible for generating **FrameTabContainer** channel is **JSPTabContainerProvider**. See [“JSPTabContainerProvider” on page 49](#) for the display profile XML for this provider.

FrameTabContainer Channel Display Profile

The provider profile is the template which decides the properties for a container channel, but the container channel profile will ultimately decide the values for the container channel attributes. See [“FrameTabContainer” on page 152](#) for the display profile for **FrameTabContainer** channel.

Container's Display Profile Properties

The properties that make up **FrameTabContainer** work as follows:

- **contentPage** is set to `frametab.jsp`. This draws the **Content Page** for the frame container.
- **editPage** is set to `frametabedit.jsp`. This displays the **Edit page** for the frame container where new pages can be added, or existing pages removed or renamed.
- **startTab** sets the page that opens first on the Desktop as **MyFrontPageFramePanelContainer**.
- **maxTabs** allows four pages to be created. As there are currently two pages, two more can be added.
- **makeTabProvider** specifies **JSPFrameCustomTableContainerProvider** as the provider to create a new page on the Desktop.

- `channelNumber` specifies that a number is appended to a newly created page as the channel name. This number is increased each time a new page is created, so that the new page will have unique name. For example, to create a new page based on `MyFrontPageFramePanelContainer` in `FrameTabContainer`, the new page channel name would be `FrameTabContainer/MyFrontPageFramePanelContainer1`. (The new page name is actually the `channelName` property in the display profile plus the value of `channelNumber` property. The `channelNumber` property value is incremented by one each time a new page is created.)
- `contentChannel` specifies `JSPContentContainer` as the content channel that provides the Content page displaying channels to add to a user-created page.
- The collection `TabProperties` has `<Collection name=>` entries for each of the four containers that are available in the `FrameTabContainer`.
- The `Available` list describes all available channels for this container. The available channels are displayed in the content preference page for users to select from.
- The `Selected` list describes selected channels for this container. Only selected channels shows up in the Desktop.

Contained Containers' Display Profile

By default, in its available list, the frame tab container channel has [MyFrontPageFramePanelContainer](#), [SamplesFramePanelContainer](#), [PredefinedFrontPageFramePanelContainer](#), and [PredefinedSamplesFramePanelContainer](#) as its contained channels.

JSP Files Used by FrameTabContainer

Sun ONE Portal Server uses JSP files for a channel's presentation layer. `FrameTabContainer` references two main JSPs, `frametab.jsp` and `frametabedit.jsp`, through the `contentPage` and `editPage` properties.

Content template is responsible for the front page of the container channel and the file name for the tab container channel is `frametab.jsp`. The `frametab.jsp` file extensively uses the Desktop taglibs. For more information on taglibs, see [Appendix B, "JavaServer Pages Tag Library Reference."](#)

The Edit page is where you can add, remove, and rename pages. The `frametabedit.jsp` is used to display this page. For a description of all JSP files used by `FrameTabContainer`, see [Appendix C, "JavaServer Pages Reference."](#)

Template-based Containers (TemplateTabContainer and TemplateTableContainer)

[TemplateTabContainer](#) and [TemplateTableContainer](#) use the original template mechanism from iPlanet™ Portal Server 3.0. [TemplateTableContainer](#) is the same as the normal Desktop from iPlanet Portal Server 3.0, and [TemplateTabContainer](#) is the same as the tabbed Desktop from iPlanet Portal Server 3.0. Use these containers for migrating from existing iPlanet Portal Server 3.0 (Service Pack 3a or 4) custom Desktops to Sun ONE Portal Server 6.2 software.

See the *Sun ONE Portal Server 6.2 Migration Guide* for more information.

TemplateTabContainer

The [TemplateTabContainer](#) uses templates for the tabbed Desktop.

Customizing the Desktop Interface

This chapter provides common customization tasks for modifying the look and feel of the Desktop.

This chapter contains the following sections:

- [Desktop Customization Overview](#)
- [Adding Containers and Channels to the Desktop](#)
- [Customizing Container Tabs](#)
- [Customizing Channels](#)
- [Creating Customized Organization JSP and Template Files](#)
- [Customizing the Anonymous Desktop](#)
- [Customizing Authentication](#)
- [Modifying the Desktop Layout](#)
- [Branding the Desktop](#)
- [Changing Desktop Colors](#)
- [Customizing the Global Themes](#)
- [Miscellaneous JSP Information](#)

Desktop Customization Overview

This section provides general overview information on customizing the Desktop interface. It includes the following sections:

- [What Are the Areas for Customizing the Desktop?](#)

- [Channel Refresh Times and Container Caching](#)
- [Contained Containers](#)
- [Guidelines for Using the dpadmin Command](#)

NOTE The procedures in this chapter use the `dpadmin` command-line utility instead of the Sun™ ONE Identity Server software administration console to administer the display profile. See the *Sun ONE Portal Server 6.2 Administrator's Guide* for information on using the administration console to administer the display profile.

What Are the Areas for Customizing the Desktop?

In general, the Sun™ ONE 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 Sun ONE Identity Server software. This guide does not describe how to customize the authentication screens. See the *Sun ONE Identity Server 6.1 Programmer'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` directory. For more information on these directories, see [“What Is the Sample Portal?” in Chapter 1, “Introduction to Customizing the Desktop.”](#)

Some other static resources are located in the `WebContainer/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 8, “Customizing the Service Providers”](#) for more information.

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` 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 refresh time 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.

Contained Containers

Often, when working with the sample portal, you need to modify the appropriate “contained” container, which is part of the top-level container. The contained containers are:

- `MyFrontPageFramePanelContainer` (**parent** = `FrameTabContainer`)
- `MyFrontPageTabPanelContainer` (**parent** = `JSPTabContainer`)
- `MyFrontPageTemplatePanelContainer` (**parent** = `TemplateTabContainer`)
- `SamplesFramePanelContainer` (**parent**=`FrameTabContainer`)
- `SamplesTabPanelContainer` (**parent** = `JSPTabContainer`)
- `SearchTabPanelContainer` (**parent** = `JSPTabContainer`)

The following containers are used as `PredefinedTabs` and listed as `PredefinedTabs` on the `makeNewTab` page so that users can create new tabs based on these Tabs. These are not, by default, directly used for display on the sample portal.

- `PredefinedFrontPageTabPanelContainer`
- `PredefinedSamplesTabPanelContainer`

Guidelines for Using the `dpadmin` Command

Use the following guidelines when running the `dpadmin` command to update the display profile:

- Make sure no other administrator is currently using the Sun ONE Identity Server software administration console or `dpadmin` command to make display profile modifications. Such a situation could cause changes to be lost, if you are running `dpadmin modify`, as there is no locking mechanism to prevent `dpadmin` and the administration console from accessing the display profile at the same time.
- The preferred sequence when using `dpadmin` is to put your modifications into a file as an XML “fragment” then run the `dpadmin` command with the `add` subcommand. For example,

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

In this example, `newtheme.xml` is a file containing the XML “fragment” to be added to the display profile.

- If you edit a display profile document directly, first use the `dpadmin` command with the `list` subcommand to obtain the latest contents of the display profile, make your edits, then run the `dpadmin` command with the `modify` subcommand. For example,

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

(Edit the dp-org.xml file.)

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

See the *Sun ONE Portal Server 6.2 Administrator's Guide* for more information on the `dpadmin` command.

CAUTION Between the time you run the `dpadmin list` and `dpadmin modify` commands, do not change the display profile document in the LDAP server in any way (by using the administration console, `dpadmin`, or `ldapmodify` commands). Otherwise, those changes will be overwritten by the latest `dpadmin modify` command.

Adding Containers and Channels to the Desktop

Sun ONE Portal Server software uses channels to generate content for the Desktop. Generally, content is created by using JSP or template files. Configure channels by manipulating display profile properties that affect the content and layout of the channel on the Desktop. When you customize the value of properties that a channel receives, you customize that channel. (If you customize the value of properties for a provider, then all channels in the Desktop that use this provider are customized.)

Channels that contain other channels are called containers. Containers arrange channel content on the Desktop.

There are two methods you can use to add containers and channels to the Desktop:

- By using the Sun ONE Identity Server software administration console (Edit Channels and Channel and Container Management links).
- By editing the display profile directly and running the `dpadmin` command to load the display profile into LDAP.

The Sun ONE Portal Server software documentation considers adding containers and channels to be administrative tasks, hence, both of these methods are described in the *Sun ONE Portal Server 6.2 Administrator's Guide*.

Customizing Container Tabs

This section provides a variety of tasks to customize the container tabs.

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 select list in the display profile.

To Add a Tab to JSPTabContainer

A tab can be any container type but 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.

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-fontFacel" 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 (see [Step a on page 216](#)), 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 7-1 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-fontFacel" value="Sans-serif"/>
    <String name="refreshTime" value="" advanced="true"/>
    <String name="width" value="thin" advanced="true"/>
    <String name="fontFacel" value="Sans-serif"/>
    <String name="productName" value="Sun ONE 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"/>

```

Code Example 7-1 PredefinedNewTabPageContainerProvider Display Profile Definition (Continued)

```

<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="SampleURLScrapper"/>
  <String value="Notes"/>
  <String value="SampleSimpleWebService"/>
</Collection>
<Collection name="channelsColumn" advanced="true">
  <String name="SampleURLScrapper" value="2"/>
  <String name="Notes" value="2"/>
  <String name="SampleSimpleWebService" value="2"/>
</Collection>
<Collection name="channelsRow" advanced="true">
  <String name="SampleURLScrapper" 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"/>
<Collection name="channelsIsMovable"/>
<Collection name="channelsIsRemovable"/>
<Collection name="channelsIsDetachable"/>
<Collection name="borderlessChannels"/>
</Properties>
</Provider>

```

- e. Define the container channel based on the PredefinedNewTabPageContainerProvider.

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 7-2 PredefinedNewTabPanelContainer Channel Properties

```

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

```

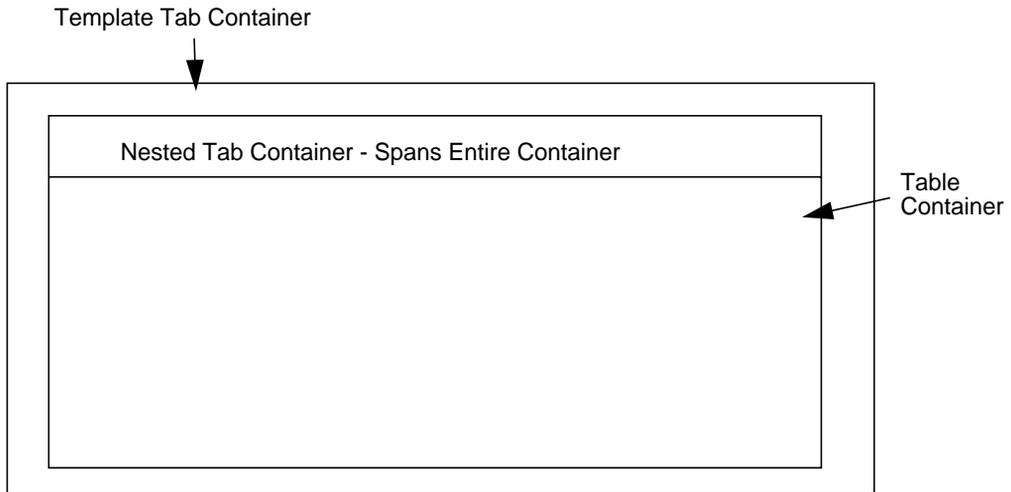
2. Load the display profile into LDAP by using the `dpadmin` command.
See [“Guidelines for Using the dpadmin Command” on page 214](#).
3. Bring up the Desktop and verify that the tab was added.

Creating a Tab Within a Tab

This is similar to [“To Add a Tab to JSPTabContainer” on page 216](#), except that instead of defining the tab based on `JSPTabContainerProvider`, you base the new tab on `JSPTabContainerProvider`.

To Stretch a Tab Across an Entire Container

[Figure 7-1 on page 221](#) 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 7-1 Tab Spanning 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.
See [“Guidelines for Using the dpadmin Command” on page 214](#).

To Change the Tab Image for JSP-based Tab Containers

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

1. Log in to the Sun ONE Identity Server 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 `tabNotchImage` property to the new image name. By default, the value for this property is `tabNotchImage`.
5. Copy the new image into `portal-server-install-root/SUNWps/web-src/desktop/tabs/images` directory.
6. Run the `portal-server-install-root/SUNWps/bin/deploy redeploy -deploy_admin_password password` command to deploy the new image.
7. Reload the Desktop to verify the change.

To Change the Color of Tabs

The background color of tabs are part of the themes.

1. Log in to the Sun ONE Identity Server 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.

To Make a Tab the Start Tab

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

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

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

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

See “[Guidelines for Using the dpadmin Command](#)” on page 214.

Setting a Role 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 `JSPTabContainer` 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="JSPTabContainer" provider="JSPTabContainerProvider"
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.

Customizing Channels

This section describes how customize channels. See the *Sun ONE Portal Server 6.2 Administrator's Guide* for instructions on adding a channel to the Desktop.

To Customize Window Preference

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

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.
See [“Guidelines for Using the dpadmin Command”](#) on page 214.

NOTE The intelligence has to be built with the help of JavaScript for that particular channel. Out of the box, this is supported by `BookmarkProvider` only.

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 [“Contained Containers”](#) on page 214.
2. Add the appropriate property (within the `<Properties>` `</Properties>`) tags of the container’s display profile you selected in [Step 1](#) from [Table 7-1 on page 226](#), 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 7-1 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"/>
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.
See [“Guidelines for Using the dpadmin Command”](#) on page 214.

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 7-2](#), 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 7-2 Channel Buttons and Corresponding Properties

Button	Property to Hide the Button
Minimize	<pre><Collection name="channelsIsMinimizable"> <Boolean name="channelname" value="false"/> </Collection></pre>
Maximize	<pre><Collection name="channelsIsMaximizable"> <Boolean name="channelname" value="false"/> </Collection></pre>
Detach	<pre><Collection name="channelsIsDetachable"> <Boolean name="channelname" value="false"/> </Collection></pre>
Remove	<pre><Collection name="channelsIsRemovable"> <Boolean name="channelname" value="false"/> </Collection></pre>

NOTE See [Table 7-1 on page 226](#) for information on the Help and Edit buttons.

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.

See [“Guidelines for Using the dpadmin Command” on page 214](#).

To Disable the User from Editing Instant Messaging Server Information

When a site has only a single 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 7-3](#) contains the Instant Messaging Server Information (snippet) in the `IMEdit.jsp` file.

Code Example 7-3 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">
    <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>

```

Code Example 7-3 Instant Messaging Server Information in IMEdit.jsp File (*Continued*)

```

        </td>
    </tr>
    <tr>
        <td valign="top">
            <font face="%=fontFace%" size="-1">In order to access a Sun ONE
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>
    </label></font><br></td>
    <td valign="middle">
        <input type="text" name="muxport" size="25" maxlength="40"
value="<dtpc:getStringProperty key="muxport"/>">
    </td>

```

Code Example 7-3 Instant Messaging Server Information in IMEdit.jsp File (*Continued*)

```

</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 ONE Identity Server 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.

To Automatically Close 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 7-4](#)). Merely uncomment this Javascript code (shown in [Code Example 7-4](#) in bold) to cause the window to close automatically.

Code Example 7-4 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 ONE 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>
    -->

```

To Customize 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 7-5](#)) in `IMContent.jsp` file.

Code Example 7-5 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"/>
  </a>
  <im:getContactName/></br>
</jx:when>
<jx:when test="$p=='AWAY'">

```

Code Example 7-5 Instant Messaging Offline Contact Information in `IMContent.jsp` File (Continued)

```

<jx:expr value="$anchortag"/>
  </a>
  <im:getContactName/></br>
</jx:when>
<jx:when test="$p=='FORWARDED'">
<jx:expr value="$anchortag"/>
  </a>
  <im:getContactName/></br>
</jx:when>
<jx:when test="$p=='OTHER'">
<jx:expr value="$anchortag"/>
  </a>
  <im:getContactName/></br>
</jx:when>
<jx:otherwise>
<jx:expr value="$anchortag"/>
   - Click to chat" border=0
align=absmiddle></a>
  <im:getContactName/></br>
</jx:otherwise>
-- End of offline states --%>

```

Creating Customized Organization JSP and Template Files

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/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/JSPPProvider/content.jsp`, copy this file to `/etc/opt/SUNWps/desktop/sesta/JSPPProvider/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.

See instructions in this chapter for the various types of customizations you can perform.

5. Change the dynamic Desktop Type attribute in the Sun ONE Identity Server software administration console to use the newly created directory.

See [“Changing the Desktop Type” on page 30](#) for more information.

Customizing the Anonymous Desktop

This section describes customizations you can make for the anonymous Desktop. When you install the sample portal, a copy of the anonymous Desktop display profile is located in the `portal-server-install-root/SUNWps/samples/desktop/dp-anon.xml` file, with the support files located in the `/etc/opt/SUNWps/desktop/anonymous` directory.

Configuring Anonymous Authentication

Sun ONE 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 Sun ONE 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 guide describes how to enable and disable both types of anonymous authentication. See the *Administering Users And Services* chapter in the *Sun ONE Portal Server 6.2 Administrator's Guide* for more information on enabling and disabling anonymous authentication.

To Enable Anonymous Log In

1. Log in to the Sun ONE Identity Server 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 Sun ONE Identity Server 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 Sun ONE Identity Server software administration console as administrator.
2. Create the authlessanonymous 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 Sun ONE Identity Server 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 Sun ONE 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 Sun ONE Identity Server software administration console as administrator.
2. Select the Service Configuration tab.
3. Click on the Desktop node.

The Desktop attributes page appears 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 Sun ONE Identity Server 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.

To Access the Anonymous Desktop Through the Identity Server 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=People,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 Sun ONE Portal Server software host, and *organization* is the name of the appropriate Sun ONE Identity Server software organization.

NOTE For anonymous, you must also make sure that only Anonymous authentication is enabled in the Sun ONE Identity Server software administration console.

3. Save the file and restart Sun ONE Portal Server software.

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

Users can now view the Anonymous Desktop by typing the Sun ONE 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?psdt.suid=uid=authlessanonym
ous,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.

To Disable the Initial Sun ONE Identity Server Software Login Page and Always Use Anonymous Log In

1. Log in to the Sun ONE Identity Server 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.
 - o UserID - anonymous
 - o First Name - (blank)
 - o Last Name - anonymous
 - o Full Name - anonymous
 - o Password - anonymous
 - o Password (confirm) - anonymous
11. Click the create button to create the user.
12. When users type the URL to access the portal server in a browser, the anonymous Desktop comes up, bypassing the Sun ONE Identity Server software login page. This Desktop will have the login channel, where users can log in if desired.

To Modify 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.

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"></td>
```

Replace `[surl:/images/productName.jpg]` with a reference to an alternate image.

For example, if you use

identity-server-install-root/SUNWam/public_html/images/newimage.gif, then use */images/newimage.gif* as your replacement text. The `[surl:]` tag references image files from the Sun ONE 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.

To Add 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 Sun ONE Identity Server 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 *Sun ONE Portal Server 6.2 Administrator's 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 Sun ONE Identity Server 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,

```
cd /opt/SUNWps/samples/desktop  
cp -r dp-anon.xml dp-anon-sesta.xml
```

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 `` to `` in all the display template files.

11. Set the Desktop type for the authless user.

- a. In the Sun ONE Identity Server 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
```

Customizing Authentication

Sun ONE Portal Server supports a number of authentication schemes, including LDAP, anonymous, membership, UNIX, and more. See the Sun ONE Portal Server 6.1 Administrator's Guide for information on configuring authentication, as well as users and roles.

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 *Sun ONE Portal Server 6.2 Administrator's 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`.

To Modify the Default Desktop (Container) for authlessanonymous User

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

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.

Modifying the Desktop Layout

This section describes how to modify the layout of channels and channel arrangement in the Desktop.

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.

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

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 *Sun ONE Portal Server 6.2 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`.

To Change the Desktop Column Layout (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 [“Guidelines for Using the dpadmin Command” on page 214.](#)

NOTE In the sample portal display profile, many width properties are defined with the `advanced=true` attribute, which means these properties are not visible in the Sun ONE Identity Server software administration console. Thus, if you use the sample portal display profile, you cannot change these values in the admin console. You need to edit the file manually then load it into LDAP by using the `dpadmin` command.

To Change the Desktop Column Layout (Sun ONE Identity Server Software Administration Console)

1. Log in to the Sun ONE Identity Server 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 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 ONE Identity Server software administration console Channel and Container Management link for the Desktop service to change the channel layout.

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 [“Guidelines for Using the dpadmin Command” on page 214.](#)

To Change the Channel Border Width

This procedure changes 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 `leafWrapper.jsp`.

Change the `cellpadding` to not use the `borderWidth` 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 JSFs.) After making the change to the display profile, which in the sample portal is `dp-org.xml`, you need to upload it to LDAP.

To Customize the Channel Border

Use this procedure 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
```

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>

```

Branding the Desktop

This section describes how to brand the Desktop with your site's logo and name.

To Change 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.

To Change the Logo (Image) in the Banner Header

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

<code>brandImage</code>	The brand image on the left of the header.
<code>brandImage2</code>	The brand image in the center of the header; if there's no need to have the second image, then use <code>spacer.gif</code> .
<code>brandImageBgColor</code>	The background color for the left image file.
<code>brandImage2BgColor</code>	The background color for the center image file.
<code>brandImageWidth</code>	The width of the left image file.
<code>previewImage</code>	The image that is displayed in the Theme/preset Themes page.

1. Log in to the Sun ONE Identity Server 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 `portal-server-install-root/SUNWps/web-src/images` directory.
6. Run `portal-server-install-root/SUNWps/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.

To Change 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:

1. Use the [Table 7-3](#) 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 7-3 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

NOTE Containers not listed in Table 6-3 on page 188 use defaultHeader.jsp and defaultMenubar.jsp. These two files are actually the same as singlePreferenceHeader.jsp and singlePreferenceMenubar.jsp. 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.

2. 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.

3. Edit the JSP files.

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

4. Run the touch command.

For example,

```
touch *.jsp
```

5. Reload the Desktop to verify the change.

Changing Desktop Colors

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

To Change Desktop Colors

Most of these colors are part of the global theme attributes. See [“Customizing the Global Themes” on page 259](#) for more information.

1. Use Table 6-4 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 7-4 Files to Customize to Change Desktop Colors

Desktop Component to Customize	File to Edit in <code>/etc/opt/SUNWps/desktop/default/</code>	Theme Attribute
Header background color	<p>JSP-based containers:</p> <ul style="list-style-type: none"> • <code>./JSPSingleContainerProvider/header.jsp</code> • <code>./JSPTabContainer/header.jsp</code> • <code>./JSPTableContainerProvider/header.jsp</code> • <code>./PredefinedFrontPageFramePanelContainerProvider/header.jsp</code> • <code>./PredefinedFrontPageTabPanelContainerProvider/header.jsp</code> • <code>./PredefinedSamplesTabPanelContainerProvider/header.jsp</code> <p>Frame-based containers:</p> <ul style="list-style-type: none"> • <code>./FrameTabContainer/banner.jsp</code> • <code>./PredefinedSamplesFramePanelContainerProvider/header.jsp</code> 	brandBgColor
Footer background color	<p>JSP-based containers:</p> <ul style="list-style-type: none"> • <code>./JSPSingleContainerProvider/menubar.jsp</code> • <code>./JSPTabContainer/menubar.jsp</code> • <code>./JSPTableContainerProvider/menubar.jsp</code> • <code>./PredefinedFrontPageFramePanelContainerProvider/menubar.jsp</code> • <code>./PredefinedFrontPageTabPanelContainerProvider/menubar.jsp</code> • <code>./PredefinedSamplesTabPanelContainerProvider/menubar.jsp</code> <p>Frame-based containers:</p> <ul style="list-style-type: none"> • <code>./FrameTabContainer/menubar.jsp</code> • <code>./PredefinedSamplesFramePanelContainerProvider/menubar.jsp</code> 	brandBgColor
Font color in the header and footer	The related JSPs are the <code>header.jsp</code> and <code>menubar.jsp</code> that listed in header background color and footer background color.	headerFontColor

Table 7-4 Files to Customize to Change Desktop Colors (*Continued*)

Desktop Component to Customize	File to Edit in <code>/etc/opt/SUNWps/desktop/default/</code>	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.

To Change 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 *Sun ONE Portal Server 6.2 Administrator's Guide* for more details.

- Define a new theme - In the display profile, you can define your own theme in the GlobalThemes collection. See [“Adding and Customizing Global Themes” on page 271](#) for more information.

Customizing the Global Themes

The Desktop theme provides the capability of creating a customizable user interface that allows the end users to select different look and feel for their Desktop.

The definition of a theme in Sun ONE Portal Server 6.2 software Desktop is a collection of user interface attributes that are used in the markup output from the Desktop. The attributes can be colors, fonts, and images. Out of the box, there are eight themes that come with the sample portal and each theme contains thirty eight (38) attributes.

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 ONE Identity Server 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.

NOTE The end user can change the following Theme attributes through the Theme customize page (please [“Theme Properties” on page 267](#) for information on these attributes):

- bgColor
 - titleBarColor
 - fontColor
 - borderColor
 - borderWidth
 - fontFace
 - fontSize
-

There are tag library functions defined to allow JSPs to retrieve the theme related values from the display profile (see [Appendix A, “Desktop Template and Tag Reference”](#) for more information.) 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`.

GlobalThemes Display Profile Definition

The following display profile XML fragment shows the eight themes defined in *portal-server-install-root/SUNWps/samples/desktop/dp-org.xml* file.

Code Example 7-6 Display Profile Definition for GlobalThemes

```
<Collection name="GlobalThemes" propagate="false">
  <Collection name="SunTheme">
    <String name="description" value="Sun"/>
    <String name="bgColor" value="white"/>
    <String name="titleBarColor" value="#999999"/>
    <String name="fontColor" value="black"/>
    <String name="borderColor" value="#999999"/>
    <String name="borderWidth" value="1"/>
    <String name="fontFace" value="Sans-serif"/>
    <String name="fontSize" value="3"/>
    <String name="activeBulletImage" value="branded_bullet_on.gif"/>
    <String name="inactiveBulletImage" value="branded_bullet_off.gif"/>
    <String name="brandImage" value="logo_sun.gif"/>
    <String name="brandImage2" value="sunONE_productlogo.gif"/>
    <String name="brandImageBgColor" value="#594FBF"/>
    <String name="brandImage2BgColor" value="#FBE249"/>
  </Collection>
</Collection>
```

Code Example 7-6 Display Profile Definition for GlobalThemes (Continued)

```

<String name="brandBgColor" value="#ffffff"/>
<String name="brandImageWidth" value="110"/>
<String name="headerBgColor" value="#e5e5e5"/>
<String name="headerFontColor" value="#594FBF"/>
<String name="headerText" value="sansSerif11Font"/>
<String name="tabNotchImage" value="tabNotch.gif"/>
<String name="tabColor" value="#cccccc"/>
<String name="tabFontColor" value="#000000"/>
<String name="contentLayoutLinkColor" value="#FFFFFF"/>
<String name="contentLayoutText" value="sansSerif10Font"/>
<String name="linkSeparatorColor" value="#CCCCCC"/>
<String name="tableBgColor" value="#FFFFFF"/>
<String name="titleFontColor" value="#ffffff"/>
<String name="titleText" value="sansSerif10Font"/>
<String name="channelHighlightColor" value="#ffffff"/>
<String name="channelLinkColor" value="#3a2eb5"/>
<String name="previewImage" value="Sun_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 name="FieryTheme">
  <String name="description" value="Fiery"/>
  <String name="bgColor" value="white"/>
  <String name="titleBarColor" value="#990000"/>
  <String name="fontColor" value="black"/>
  <String name="borderColor" value="#990000"/>
  <String name="borderWidth" value="1"/>
  <String name="fontFace" value="Sans-serif"/>
  <String name="fontSize" value="3"/>
  <String name="activeBulletImage" value="fiery_bullet_on.gif"/>
  <String name="inactiveBulletImage" value="fiery_bullet_off.gif"/>
  <String name="brandImage" value="Fiery_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="#FF3300"/>
  <String name="tabFontColor" value="#000000"/>
  <String name="contentLayoutLinkColor" value="#ff3300"/>
  <String name="contentLayoutText" value="sansSerif10Font"/>
  <String name="linkSeparatorColor" value="#000000"/>
  <String name="tableBgColor" value="#999999"/>
  <String name="titleFontColor" value="#ffffff"/>

```

Code Example 7-6 Display Profile Definition for GlobalThemes (Continued)

```

<String name="titleText" value="sansSerif10Font"/>
<String name="channelHighlightColor" value="#ffffff"/>
<String name="channelLinkColor" value="#3a2eb5"/>
<String name="previewImage" value="Fiery_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 name="BreezyTheme">
  <String name="description" value="Breezy"/>
  <String name="bgColor" value="white"/>
  <String name="titleBarColor" value="#336699"/>
  <String name="fontColor" value="black"/>
  <String name="borderColor" value="#336699"/>
  <String name="borderWidth" value="1"/>
  <String name="fontFace" value="Sans-serif"/>
  <String name="fontSize" value="3"/>
  <String name="activeBulletImage" value="breezy_bullet_on.gif"/>
  <String name="inactiveBulletImage" value="breezy_bullet_off.gif"/>
  <String name="brandImage" value="Breezy_FFFFFFFF.gif"/>
  <String name="brandImage2" value="spacer.gif"/>
  <String name="brandImageBgColor" value="#FFFFFF"/>
  <String name="brandImage2BgColor" value="#FFFFFF"/>
  <String name="brandBgColor" value="#FFFFFF"/>
  <String name="headerBgColor" value="#FFFFFF"/>
  <String name="brandImageWidth" value="243"/>
  <String name="headerFontColor" value="#000000"/>
  <String name="headerText" value="sansSerif11Font"/>
  <String name="tabNotchImage" value="tabNotch.gif"/>
  <String name="tabColor" value="#CCCC66"/>
  <String name="tabFontColor" value="#000000"/>
  <String name="contentLayoutLinkColor" value="#FFFFFF"/>
  <String name="contentLayoutText" value="sansSerif10Font"/>
  <String name="linkSeparatorColor" value="#ED471E"/>
  <String name="tableBgColor" value="#CCCCFF"/>
  <String name="titleFontColor" value="#ffffff"/>
  <String name="titleText" value="sansSerif10Font"/>
  <String name="channelHighlightColor" value="#ffffff"/>
  <String name="channelLinkColor" value="#3a2eb5"/>
  <String name="previewImage" value="Breezy_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>

```

Code Example 7-6 Display Profile Definition for GlobalThemes (Continued)

```

<Collection name="SereneTheme">
  <String name="description" value="Serene"/>
  <String name="bgColor" value="white"/>
  <String name="titleBarColor" value="#EAE7E7"/>
  <String name="fontColor" value="black"/>
  <String name="borderColor" value="#EAE7E7"/>
  <String name="borderWidth" value="0"/>
  <String name="fontFace" value="Sans-serif"/>
  <String name="fontSize" value="3"/>
  <String name="activeBulletImage" value="serene_bullet_on.gif"/>
  <String name="inactiveBulletImage" value="serene_bullet_off.gif"/>
  <String name="brandImage" value="Serene_330099.gif"/>
  <String name="brandImage2" value="spacer.gif"/>
  <String name="brandImageBgColor" value="#330099"/>
  <String name="brandImage2BgColor" value="#330099"/>
  <String name="brandBgColor" value="#330099"/>
  <String name="headerBgColor" value="#330099"/>
  <String name="brandImageWidth" value="243"/>
  <String name="headerFontColor" value="FFFFFF"/>
  <String name="headerText" value="sansSerif11Font"/>
  <String name="tabNotchImage" value="serene_tabend.gif"/>
  <String name="tabColor" value="#9999FF"/>
  <String name="tabFontColor" value="000000"/>
  <String name="contentLayoutLinkColor" value="#330099"/>
  <String name="contentLayoutText" value="sansSerif10Font"/>
  <String name="linkSeparatorColor" value="#9999FF"/>
  <String name="tableBgColor" value="FFFFFF"/>
  <String name="titleFontColor" value="#330099"/>
  <String name="titleText" value="sansSerif10Font"/>
  <String name="channelHighlightColor" value="ffffcc"/>
  <String name="channelLinkColor" value="#3a2eb5"/>
  <String name="previewImage" value="Serene_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 name="SportingTheme">
  <String name="description" value="Sporting"/>
  <String name="bgColor" value="white"/>
  <String name="titleBarColor" value="#009999"/>
  <String name="fontColor" value="black"/>
  <String name="borderColor" value="#009999"/>
  <String name="borderWidth" value="1"/>
  <String name="fontFace" value="Sans-serif"/>
  <String name="fontSize" value="3"/>
  <String name="activeBulletImage" value="sporting_bullet_on.gif"/>
  <String name="inactiveBulletImage" value="sporting_bullet_off.gif"/>
  <String name="brandImage" value="Sporting_000000.gif"/>
  <String name="brandImage2" value="spacer.gif"/>

```

Code Example 7-6 Display Profile Definition for GlobalThemes (Continued)

```

<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="CCCCCC"/>
<String name="tabFontColor" value="#000000"/>
<String name="contentLayoutLinkColor" value="#000000"/>
<String name="contentLayoutText" value="sansSerif10Font"/>
<String name="linkSeparatorColor" value="CCCCCC"/>
<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="Sporting_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 name="DustyTheme">
  <String name="description" value="Dusty"/>
  <String name="bgColor" value="white"/>
  <String name="titleBarColor" value="#330000"/>
  <String name="fontColor" value="black"/>
  <String name="borderColor" value="#330000"/>
  <String name="borderWidth" value="1"/>
  <String name="fontFace" value="Sans-serif"/>
  <String name="fontSize" value="3"/>
  <String name="activeBulletImage" value="dusty_bullet_on.gif"/>
  <String name="inactiveBulletImage" value="dusty_bullet_off.gif"/>
  <String name="brandImage" value="Dusty_FFFFFFFF.gif"/>
  <String name="brandImage2" value="spacer.gif"/>
  <String name="brandImageBgColor" value="FFFFFF"/>
  <String name="brandImage2BgColor" value="FFFFFF"/>
  <String name="brandBgColor" value="FFFFFF"/>
  <String name="headerBgColor" value="FFFFFF"/>
  <String name="brandImageWidth" value="243"/>
  <String name="headerFontColor" value="#330000"/>
  <String name="headerText" value="sansSerif11Font"/>
  <String name="tabNotchImage" value="tabNotch.gif"/>
  <String name="tabColor" value="#999966"/>
  <String name="tabFontColor" value="#000000"/>
  <String name="contentLayoutLinkColor" value="CC9900"/>
  <String name="contentLayoutText" value="sansSerif10Font"/>
  <String name="linkSeparatorColor" value="ED471E"/>

```

Code Example 7-6 Display Profile Definition for GlobalThemes (Continued)

```

<String name="tableBgColor" value="#CCCC99"/>
<String name="titleFontColor" value="#FFFFCC"/>
<String name="titleText" value="sansSerif10Font"/>
<String name="channelHighlightColor" value="#ffffff"/>
<String name="channelLinkColor" value="#3a2eb5"/>
<String name="previewImage" value="Dusty_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 name="CollegiateTheme">
  <String name="description" value="Collegiate"/>
  <String name="bgColor" value="white"/>
  <String name="titleBarColor" value="#000033"/>
  <String name="fontColor" value="black"/>
  <String name="borderColor" value="#000033"/>
  <String name="borderWidth" value="1"/>
  <String name="fontFace" value="Sans-serif"/>
  <String name="fontSize" value="3"/>
  <String name="activeBulletImage" value="collegiate_bullet_on.gif"/>
  <String name="inactiveBulletImage" value="collegiate_bullet_off.gif"/>
  <String name="brandImage" value="Collegiate_FFFFFFFF.gif"/>
  <String name="brandImage2" value="spacer.gif"/>
  <String name="brandImageBgColor" value="#FFFFFF"/>
  <String name="brandImage2BgColor" value="#FFFFFF"/>
  <String name="brandBgColor" value="#FFFFFF"/>
  <String name="headerBgColor" value="#FFFFFF"/>
  <String name="brandImageWidth" value="243"/>
  <String name="headerFontColor" value="#000000"/>
  <String name="headerText" value="sansSerif11Font"/>
  <String name="tabNotchImage" value="tabNotch.gif"/>
  <String name="tabColor" value="#EED23A"/>
  <String name="tabFontColor" value="#000000"/>
  <String name="contentLayoutLinkColor" value="#EED23A"/>
  <String name="contentLayoutText" value="sansSerif10Font"/>
  <String name="linkSeparatorColor" value="#ED471E"/>
  <String name="tableBgColor" value="#CCCCCC"/>
  <String name="titleFontColor" value="#ffffff"/>
  <String name="titleText" value="sansSerif10Font"/>
  <String name="channelHighlightColor" value="#ffffff"/>
  <String name="channelLinkColor" value="#3a2eb5"/>
  <String name="previewImage" value="Collegiate_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"/>

```

Code Example 7-6 Display Profile Definition for GlobalThemes (Continued)

```

    <String name="editImage" value="b_edit.gif"/>
</Collection>
<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="SunTheme" propagate="false"/>
<Collection name="CustomTheme" propagate="false">
  <String name="bgColor" value=""/>
  <String name="titleBarColor" value=""/>
  <String name="fontColor" value=""/>
  <String name="borderColor" value=""/>
  <String name="borderWidth" value=""/>
  <String name="fontFace" value=""/>
  <String name="fontSize" value=""/>
  <String name="activeBulletImage" value=""/>

```

Code Example 7-6 Display Profile Definition for GlobalThemes (Continued)

```

<String name="inactiveBulletImage" value=""/>
<String name="brandImage" value=""/>
<String name="brandImage2" value=""/>
<String name="brandImageBgColor" value=""/>
<String name="brandImage2BgColor" value=""/>
<String name="brandBgColor" value=""/>
<String name="headerBgColor" value=""/>
<String name="brandImageWidth" value=""/>
<String name="headerFontColor" value=""/>
<String name="headerText" value=""/>
<String name="tabNotchImage" value=""/>
<String name="tabColor" value=""/>
<String name="tabFontColor" value=""/>
<String name="contentLayoutLinkColor" value=""/>
<String name="contentLayoutText" value=""/>
<String name="linkSeparatorColor" value=""/>
<String name="tableBgColor" value=""/>
<String name="titleFontColor" value=""/>
<String name="titleText" value=""/>
<String name="channelHighlightColor" value=""/>
<String name="channelLinkColor" value=""/>
<String name="previewImage" value=""/>
<String name="helpImage" value=""/>
<String name="removeImage" value=""/>
<String name="minimizeImage" value=""/>
<String name="maximizeImage" value=""/>
<String name="normalizeImage" value=""/>
<String name="attachImage" value=""/>
<String name="detachImage" value=""/>
<String name="editImage" value=""/>
</Collection>
</Collection>

```

Theme Properties

The following is a list of theme properties that can be defined, modified, and/or customized in the display profile document. Please reference to the [“Glossary of Terms” on page 270](#) for more detailed description for the theme properties.

In the following table, column one lists the theme property name and column two provides a brief description of the corresponding theme property.

activeBulletImage	activeBulletGraphics
inactiveBulletImage	inactiveBulletGraphics
brandImage	logo image

<code>brandImage2</code>	Product name image
<code>brandImageBgColor</code>	header logo bg color
<code>brandImage2BgColor</code>	header product name bg color
<code>brandBgColor</code>	header link box bg color
<code>headerBgColor</code>	header bg color and footer bg color
<code>headerFontColor</code>	header font color
	footer font color
<code>headerText</code>	header font size
	footer font size
	header font face
	footer font face
<code>tabNotchImage</code>	tabNotch image
<code>titleText</code>	selected tab font face
	selected tab font size
	unselected tab font face
	unselected tab font size
	channel title font face
	channel title font size
<code>titleFontColor</code>	selected tab font color
	channel title font color
<code>fontSize</code>	channel font size
	channel link font size
<code>fontFace</code>	channel font face
	channel link font face

<code>titleBarColor</code>	selected tab bg color
<code>borderColor</code>	channel title bar bg color content/layout bar color channel border color page piping color (bottom) button bg color
<code>tabColor</code>	unselected tab bg color secondary channel title bar color
<code>tabFontColor</code>	unselected tab font color
<code>bgColor</code>	channel bg color
<code>fontColor</code>	channel font color
<code>borderWidth</code>	channel border width
<code>tableBgColor</code>	table bg color page piping, top
<code>channelHighlightColor</code>	highlight color for channel content (as seen in the Placida theme JSP channel)
<code>linkSeparatorColor</code>	link separator color (in the toolbar, between Content and Layout) footer link separator color
<code>channelLinkColor</code>	channel link color
<code>contentLayoutLinkColor</code>	content/layout link color
<code>contentLayoutText</code>	content/layout link font size content/layout link font face
<code>brandImageWidth</code>	brand image width
<code>previewImage</code>	preview image (on preset themes page)
<code>removeImage</code>	remove image (for the channel title bar)
<code>detachImage</code>	detach image (for the channel title bar)
<code>helpImage</code>	help image (for the channel title bar)
<code>editImage</code>	edit image (for the channel title bar)

<code>minimizeImage</code>	minimize image (for the channel title bar)
<code>maximizeImage</code>	maximize image (for the channel title bar)
<code>normalizeImage</code>	normalize image (in the maximized channel)
<code>attachImage</code>	attach image (in the popup window)

Glossary of Terms

The following table give some detailed description of where the theme attributes are actually used in the desktop.

Table 7-5 Glossary of Terms

Term	Description
Header	The banner area at the top of the portal page. Contains the branding and the global links.
Bullet graphics	The “dot” graphics that go next to the global links in the header
Logo, product name, link	The three areas in the header for the Sun theme
Footer	The narrow banner at the bottom of the portal page. Contains the global links
Tab notch	The graphic that goes in the upper left corner of the tab table cells
Selected tab	The tab whose contents are displayed
Unselected tab	The other tabs whose contents are not seen
Content/layout bar	The tool bar underneath the tabs that contains the content and layout links
Channel	The data containers displayed inside each tab
Page piping	The narrow bands of color at the top and bottom of the portal page
Table background	The areas the channels sit in
Highlight color for channel content	A contrasting color for tables inside channels
Secondary channel title bar	An extra color bar beneath the standard channel title bar
Link separator	A pipe used between links in the content/layout bar

Adding and Customizing Global Themes

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

To Add 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 Sun ONE Identity Server administration console, or via the `dpadmin` command.

2. Copy new images into the `portal-server-install-root/SUNWps/web-src`. That is:
 - o `activeBulletImage`, `inactiveBulletImage`, `brandImage`, `brandImage2`, `previewImage`: **these images must be copied in to the `portal-server-install-root/SUNWps/web-src/images` directory.**
 - o `helpImage`, `removeImage`, `minimizeImage`, `maximizeImage`, `normalImage`, `attachImage`, `detachImage`, `editImage`: **these images must be copied in to the `portal-server-install-root/SUNWps/web-src/desktop/images` directory.**
 - o `tabNotchImage` **must be copied in to the `portal-server-install-root/SUNWps/web-src/desktop/tabs/images` directory.**
3. Run the `deploy` command to deploy the image files. For example, type:


```
portal-server-install-root/SUNWps/bin/deploy redeploy -deploy_admin_password password
```
4. Verify that the new theme shows up on the Desktop's Theme page.

To Customize 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. See [“Guidelines for Using the dpadmin Command” on page 214](#) for loading the the display profile into LDAP by using the `dpadmin` command.

In the sample portal, the

portal-server-install-root/SUNWps/samples/desktop/dp-org.xml file defines eight themes. See also, “[GlobalThemes Display Profile Definition](#)” on page 260 for the default display profile XML fragment for GlobalThemes.

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

WebContainerInstance/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,

```
/opt/SUNWps/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.

Miscellaneous JSP Information

This section contains miscellaneous information on using JSPs when customizing the Desktop.

Performing 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).toString() + "?action=edit&provider=ipsdtJSPeditChannel" + "&targetprovider=" + p.getName() + "&redit=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).toString() + "?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,

```
touch tab.jsp
```

NOTE Once 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,” on page 149 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 Sun ONE 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.

Desktop Debug Information

Use the following to help debug the Desktop and JSP 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.)

The JSP classes are created at:

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

Debugging 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.

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 ONE 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 JSPPProvider to access Sun™ ONE Portal Server back end services. The Search provider uses 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 “[SearchProvider](#)” on page 86 for the display profile properties you can set for the provider.

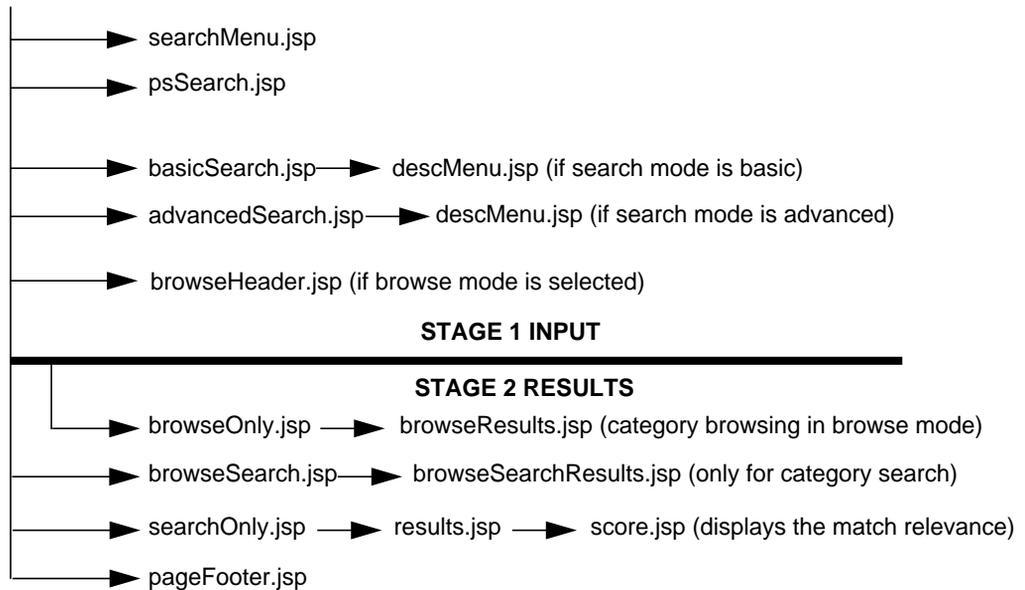
Search JavaServer Pages and Tag Libraries

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 8-1 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 8-1 JSP Layout for `searchContent.jsp`

searchContent.jsp

See [Appendix C, “JavaServer Pages Reference”](#) for more information on the Search JSP files.

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

- Jakarta tag library, for any generic tags
- Desktop tag library, for Sun ONE 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 [Appendix B, “JavaServer Pages Tag Library Reference”](#) 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 List 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 [Appendix C, “JavaServer Pages Reference.”](#)

Tips for Customizing the Service Providers

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

Debugging the Service Providers

Sun ONE 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-pssservername/portal
```

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

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

See the *Sun ONE Portal Server 6.2 Administrator's 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 276](#) and [“Recompiling JSPs” on page 276](#).

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` as follows:

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

Replace the bold portion with:

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

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

```
http://server:port/portal/dt?provider=JSPDynamicSingleContainer&JSPDynamicSing
leContainer.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 Sun ONE 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 *Sun ONE Portal Server 6.2 Administrator's Guide* for information on configuring the `searchServer` property for the Search provider.

To Modify the Default Search Server:

1. Log in to the Sun ONE Identity Server 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` 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` 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,

```
touch *.jsp
```

4. Reload the Desktop to verify the change.

To Remove content-length from Search Results

1. (Optional) Modify `searchOnly.jsp` 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` 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"/>
```

Remove `<search:getValue soifAttribute="content-length" id="sz"/>` from this file.

3. Run the `touch` command.

For example,

```
touch *.jsp
```

4. 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,

```
touch *.jsp
```

4. Reload the Desktop to verify the change.

To Remove author from the Advanced Search Interface

1. Comment out 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&nbsp;</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,

```
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&nbsp;</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,

```
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.
 - o `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>
```

- o `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:

- a. 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-length
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` 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` 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.

Customizing the Desktop End-User Online Help

This chapter provides information on customizing the Sun™ ONE 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 9-1](#) specifies different help files for different locales and different client type.

Code Example 9-1 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 9-1](#) 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:

portal-server-install-root/SUNWps/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:

portal-server-install-root/SUNWps/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:

portal-server-install-root/SUNWps/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.

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

To Edit 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 `portal-server-install-root/SUNWps/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 Sun ONE Portal Server software.

For example, if you use the `UserInfoProvider` that ships with the Sun ONE 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 299](#) for the online help file names and descriptions, and where they are installed on your system.

To Create 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 Sun ONE Identity Server software administration console.

To create a new online help file and to define the `helpURL` value manually, 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:

```
portal-server-install-root/SUNWps/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 `portal-server-install-root/SUNWps/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.”](#)

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 Sun ONE Identity Server 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:

```
portal-server-install-root/SUNWps/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 `portal-server-install-root/SUNWps/bin/deploy redeploy -deploy_admin_password password` command to deploy the file.
4. Login to the Sun ONE Identity Server 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.

Modifying the Desktop End-User Online Help HTML files

Desktop Template and Tag Reference

This appendix describes the Desktop template files and serves as a reference to the tags found in these templates.

This appendix contains the following sections:

- [Overview of Desktop Templates and Tags](#)
- [Desktop Templates in the default Directory and Tag Reference](#)
- [Desktop Templates in the sampleportal Directory and Tag Reference](#)

Overview of Desktop Templates and Tags

Sun™ ONE Portal Server uses two types of files for displaying Desktop channels and pages: JavaServer Pages™ (JSP™) and template files. JavaServer Pages are the preferred way of displaying the Desktop. This appendix describes only the Desktop templates. See [Appendix C, “JavaServer Pages Reference”](#) for information about JavaServer Pages.

The desktop default JSP and template files are installed in the `/etc/opt/SUNWps/desktop/default` directory and if you choose to install sample portal, the sample portal part of the JSP and template files are installed the `/etc/opt/SUNWps/desktop/sampleportal` directory. For more information on these two directories, see [Chapter 1, “Introduction to Customizing the Desktop.”](#)

The Desktop Type attribute in the Desktop attributes page of the Sun ONE Identity Server software administration console specifies from what subdirectory to retrieve either the JSP or template files for the Desktop. The default for this attribute is `sampleportal`, meaning the `sampleportal` subdirectory.

For more information on the desktop type attribute, see [Chapter 1, “Introduction to Customizing the Desktop.”](#)

How the Desktop Template Tags Work

A template produces a channel, page, or table. The tags in a template are swapped with for real values at runtime. For example, the `[tag:netmailSettings]` tag swaps in a partial HTML template that has mail server information. The `[tag:fontFace]` tag swaps in the value of the font that the user has chosen on the Theme page. The `[tag:switchColumns]` tag swaps in the `switchColumns.js` template file, a JavaScript™ template, that lets users change how the columns are displayed on their Desktops.

Template Lookup

For information on how the Desktop locates the template, see [“Changing the Desktop Type” on page 30](#). The directory search order for the template and JSP files is as follows:

desktoptype_locale/channelname/clientPath

desktoptype_locale/provider/clientPath

desktoptype_locale/channelname

desktoptype_locale/provider

desktoptype_locale/clientPath

desktoptype_locale

desktoptype/channelname/clientPath

desktoptype/provider/clientPath

desktoptype/channelname

desktoptype/provider

desktoptype/clientPath

desktoptype

default_locale/channelname/clientPath

default_locale/provider/clientPath

default_locale/channelname

default_locale/provider
default_locale/clientPath
default_locale
default/channelname/clientPath
default/provider/clientPath
default/channelname
default/provider
default/clientPath
default
templatroot

Where

- *desktopType* is the value of the `desktopType` property
- *locale* is the user's locale
- *channelname* is the name of the channel
- *provider* is the provider name
- *clientPath* is an optional file-path containing client-specific templates
- *templatroot* is, by default, `/etc/opt/SUNWps/desktop`

If there is no `clientPath` specified, then the directory search order is as follows:

desktopType_locale/channelname
desktopType_locale/provider
desktopType_locale
desktopType/channelname
desktopType/provider
desktopType
default_locale/channelname
default_locale/provider
default_locale
default/channelname
default/provider

default

templatroot

The root of the search directory (default value of `/etc/opt/SUNWps/desktop/`) can be changed by modifying the `templateBaseDir` property in the `desktopconfig.properties` file.

Kinds of Tags Used in the Desktop

The following is a two column table: column one lists the tag and column two provides a brief description of the corresponding tag.

<code>[url:url]</code>	<code>url</code> will be encoded at run time with the <code>ProviderContext.encodeURL()</code> method.
<code>[surl:url]</code>	<code>url</code> will be pre-appended by the static root at run time. For example, in a Sun™ ONE Web Server instance, <code>[surl:/desktop/imagesnothing.gif]->/var/opt/SUNWps/https-<i>servername</i>/portal/web-apps/desktop/images/nothing.gif</code>
<code>[dturl]</code>	<code>dturl</code> will be replaced by the desktop URL at run time. For example, <code>[dturl]?action=logout->http://<i>server</i>:<i>port</i>/portlet/dt?action=logout</code>

Desktop Templates in the default Directory and Tag Reference

The templates described in this appendix are in the following provider subdirectories, beneath the `/etc/opt/SUNWps/desktop/default` directory:

- [AddressBookProvider](#)
- [AppProvider](#)
- [BookmarkProvider](#)
- [CalendarProvider](#)

- [error](#)
- [LoginProvider](#)
- [LotusNotesAddressBookProvider](#), [LotusNotesCalendarProvider](#), and [LotusNotesMailProvider](#)
- [MailCheckProvider](#)
- [MailProvider](#)
- [MSEExchangeAddressBookProvider](#), [MSEExchangeCalendarProvider](#), and [MSEExchangeMailProvider](#)
- [TemplateTabContainerProvider](#) and [TemplateTableContainerProvider](#)
- [UserInfoProvider](#)
- [default](#) directory template files

AddressBookProvider

[Table A-1](#) lists the templates in the `html` subdirectory of the `AddressBookProvider` and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-1 AddressBookProvider Template Files

File Name and Description	Tag Name and Description
<code>display-clientURL.template</code> - Used for displaying the "Launch Address Book" link	[tag:ab-display-clientURL-appURL] - Used to display the application launch link
<code>display-entries.template</code> - Used for formatting the table headers	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:ab-display-entry-list] - Used as a placeholder to put in all of the address book entries
<code>display-entry.template</code> - Used for formatting the display of the address book entry	[tag:bgColor] - Provider background color chosen [tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:ab-display-entry-firstname] - Displays first name [tag:ab-display-entry-lastname] -Displays last name [tag:ab-display-entry-commonname] -Displays common name [tag:ab-display-entry-email] -Displays email [tag:ab-display-entry-email-link] -Displays email as a link

Table A-1 AddressBookProvider Template Files *(Continued)*

File Name and Description	Tag Name and Description
<code>display-error.template</code> - Used for displaying error messages	[tag:ab-display-error] - Displays error message
<code>display-summary.template</code> - Used for formatting number of total and unread messages	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:ab-display-summary-entries] - Displays a summary
<code>display.template</code> - Used for overall channel formatting	[tag:ab-display-summary] - Displays summary [tag:ab-display-entries] - Displays entries [tag:ab-display-clientURL]
<code>edit-checkbox.template</code> - Used for creating edit page checkboxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:attSelected] - Used to dynamically build edit pages. Should not be edited.
<code>edit-end.template</code> - Used for creating end of the edit page	No tags.
<code>edit-link.template</code> - Used for creating application helper edit link	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:editLink] - Used to display link for the application helper editing
<code>edit-password.template</code> - Used for creating edit page password boxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:attValue] - Used to dynamically build edit pages. Should not be edited.
<code>edit-select.template</code> - Used for creating edit page select boxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:options] - Used to dynamically build edit pages. Should not be edited.
<code>edit-selectoption.template</code> - Used for creating edit page select box options	[tag:attValue], [tag:selected], [tag:string] - Used to dynamically build edit pages. Should not be edited.

Table A-1 AddressBookProvider Template Files *(Continued)*

File Name and Description	Tag Name and Description
edit-start.template - Used for creating the start of the edit page	[tag:process], [tag:isAppHandler] - Used to dynamically build edit pages. Should not be edited.
	[tag:mail-display-error] - Displays error message
	[tag:borderColor] - Color of the channel border. The border color can be changed by user in the desktop custom theme page.
	[tag:bgColor] - Provider background color chosen
	[tag:titlebarColor] - Color of title bar
	[tag:fontFace] - Font chosen
edit-string.template - Used for creating edit page text boxes	[tag:fontColor] - Color of font chosen
	[tag:fontFace] - Font chosen
	[tag:fontColor] - Color of font chosen
	[tag:resourceName], [tag:header], [tag:attName], [tag:attValue] - Used to dynamically build edit pages. Should not be edited.

AppProvider

[Table A-2](#) lists the templates in the `AppProvider` subdirectory and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-2 AppProvider Template Files

App Template and Description	Tag Name and Description
display.template - Contains the JavaScript code that launches the windows that the HTML applications show up in.	[tag:apps] - Replaced with the application content

BookmarkProvider

[Table A-3 on page 312](#) lists the templates in the `html` subdirectory of `BookmarkProvider` and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-3 BookmarkProvider Template Files

Bookmark Template and Description	Tag Name and Description
<code>display.template</code> - Contains JavaScript code for the Bookmark provider to open new windows with the URLs typed in and do the correct URL <code>http://</code> prepending. Also contains a small bit of formatting for the URL entry box.	<p>[tag:windowOption] - JavaScript variable default for launching Bookmark windows (taken from prefs)</p> <p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:bookmarks] - List of bookmark links</p> <p>[tag:channelName] - Channel name as defined in the channel display profile definition.</p>
<code>edit.template</code> - Contains the formatting for the Edit page for the Bookmark provider.	<p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:resourceCount] - Number of bookmarks</p> <p>[tag:resourceName] - Name of the bookmark</p> <p>[tag:resourceURL] - URL of the bookmark</p> <p>[tag:resourceList] - Checkable list of bookmarks for Edit page in table format</p> <p>[tag:windowOptions] - Default for checkboxes of how the bookmark should be opened (new window, existing window, and so on)</p>
<code>editUrlWrapper.template</code> - Used by the edit page of Bookmark Provider to draw the part where the bookmarks are removed.	<p>[tag:index] - Used to reference the URLs in the edit page</p> <p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:targetName] - Name of Bookmark used on edit page</p> <p>[tag:targetValue] - URL for bookmark used on edit page</p>
<code>editWindowOption.template</code> - Contains the markup for the radio buttons used to select the window option.	<p>[tag:all_new_checked] - the value for this is either CHECKED or "" based on the window preference specified in edit page</p> <p>[tag:ownWindow] - localized string from resource bundle displaying the text for window options on the edit page.</p> <p>[tag:one_new_checked] - the value for this is either CHECKED or "" based on the window preference selected in edit page</p> <p>[tag:singleWindow] - localized string from resource bundle displaying the text for window options on the edit page.</p> <p>[tag:same_checked] - the value for this is either CHECKED or "" based on the window preference selected in edit page</p> <p>[tag:mainWindow] - localized string from resource bundle displaying the text for window options on the edit page.</p>

Table A-3 BookmarkProvider Template Files *(Continued)*

Bookmark Template and Description	Tag Name and Description
urlWrapper.template - Contains markup for each url shown in display.template.	[tag:fontFace1] - Default font for the Desktop (sans-serif)
	[tag:link] - URL for bookmark used for constructing the channel content.
	[tag:channelName] - Channel name as defined in the channel display profile definition.
	[tag:name] - Name of bookmark used in the channel content.

CalendarProvider

Table A-4 lists the templates in the `html` subdirectory of the `CalendarProvider` and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-4 CalendarProvider Template Files

CalendarProvider Template and Description	Tag Name and Description
display-clientURL.template - Used for displaying the Launch Calendar link.	[tag:calendar-display-client-uri] - Used to display the application launch link
display-dayView-emptyEventList.template - Used for displaying the message "No events scheduled for today."	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)
display-dayView-emptyTaskList.template - Used for displaying the message "No tasks are pending for today."	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)

Table A-4 CalendarProvider Template Files *(Continued)*

CalendarProvider Template and Description	Tag Name and Description
display-dayView-event.template - Used for formatting events	[tag:calendar-display-dayView-event-startHourOfDay0] - start hour for 0 based 24-hour clock
	[tag:calendar-display-dayView-event-startHourOfDay1] - start hour for 1 based 24-hour clock
	[tag:calendar-display-dayView-event-startHour0] - start hour for 0 based 12-hour clock
	[tag:calendar-display-dayView-event-endHourOfDay0] - end hour for 0 based 24-hour clock
	[tag:calendar-display-dayView-event-endHourOfDay1] - end hour for 1 based 24-hour clock
	[tag:calendar-display-dayView-event-endHour0] - end hour for 0 based 12-hour clock
	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:calendar-display-event-conflict] - if the event is in conflict, then it is identified by the resource bundle 'conflict' value.
	[tag:calendar-display-dayView-event-startHour2] - start hour based on user preference of 12 or 24 hour clock format
	[tag:calendar-display-dayView-event-startMinute] - minutes of the start time
	[tag:calendar-display-dayView-event-startAmPm] - am or pm identifier
	[tag:calendar-display-dayView-event-endHour2] - end hour based on user preference of 12 or 24 hour clock format
	[tag:calendar-display-dayView-event-endMinute] - minutes of the end time
	[tag:calendar-display-dayView-event-endAmPm] - am or pm identifier
	[tag:calendar-display-dayView-event-location] - event location
	[tag:calendar-display-dayView-event-summary] - event summary
[tag:calendar-display-dayView-description-seperator] - description seperator specified by the resource bundle 'seperatorDescription' value	
[tag:calendar-display-dayView-event-description] - event description	

Table A-4 CalendarProvider Template Files *(Continued)*

CalendarProvider Template and Description	Tag Name and Description
display-dayView-eventAllDay.template - Used for formatting an all day event	<p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-dayView-event-allDay] - if the event is an All Day event, then it is identified by the resource bundle 'CalendarProvider-allDayEvent' value</p> <p>[tag:calendar-display-dayView-event-location] - event location</p> <p>[tag:calendar-display-dayView-event-summary] - event summary</p> <p>[tag:calendar-display-dayView-description-seperator] - description seperator specified by the resource bundle 'seperatorDescription' value</p> <p>[tag:calendar-display-dayView-event-description] - event description</p>
display-dayView-otherTasks.template - Used for formatting other tasks	<p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-dayView-task-summary] - task summary</p> <p>[tag:calendar-display-dayView-description-seperator] - description seperator specified by the resource bundle 'seperatorDescription' value</p> <p>[tag:calendar-display-dayView-task-description] - task description</p>

Table A-4 CalendarProvider Template Files *(Continued)*

CalendarProvider Template and Description	Tag Name and Description
display-dayView-overdueTasks.template - Used for formatting overdue tasks	<p>[tag:calendar-display-dayView-task-dueHour1] - hour of task for 12-hour clock</p> <p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-dayView-task-dueMonth] - month the task was due</p> <p>[tag:calendar-display-dayView-task-dueDay] - day of the month the task was due</p> <p>[tag:calendar-display-dayView-task-dueYear] - year the task was due</p> <p>[tag:calendar-display-dayView-task-dueHour2] - end hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-dayView-task-dueMinute] - minutes of the due time</p> <p>[tag:calendar-display-dayView-task-dueAmPm] - am or pm identifier</p> <p>[tag:calendar-display-dayView-overdueTask-summary] - overdue task summary</p> <p>[tag:calendar-display-dayView-description-seperator] - description seperator specified by the resource bundle 'seperatorDescription' value</p> <p>[tag:calendar-display-dayView-overdueTask-description] - task description</p>

Table A-4 CalendarProvider Template Files *(Continued)*

CalendarProvider Template and Description	Tag Name and Description
display-dayView-task.template - Used for formatting a "normal" task	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:calendar-display-dayView-task-pendHour2] - hour based on user preference of 12 or 24 hour clock format
	[tag:calendar-display-dayView-task-pendMinute] - minutes of the task
	[tag:calendar-display-dayView-task-pendAmPm] - am or pm identifier
	[tag:calendar-display-dayView-task-location] - task location
	[tag:calendar-display-dayView-task-complete-start] - start time of completed task
	[tag:calendar-display-dayView-task-summary] - task summary
	[tag:calendar-display-dayView-description-seperator] - description seperator specified by the resource bundle 'seperatorDescription' value
	[tag:calendar-display-dayView-task-description] - task description

Table A-4 CalendarProvider Template Files *(Continued)*

CalendarProvider Template and Description	Tag Name and Description
display-dayView.template - Used for formatting the layout of all tasks and events	[tag:calendar-display-dayView-dayOfWeek] - today's day of week
	[tag:calendar-display-dayView-month] - today's month
	[tag:calendar-display-dayView-day] - today's day in month
	[tag:calendar-display-dayView-year] - today's year
	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:calendar-display-dayView-overdueTaskNum] - number of overdue tasks for today
	[tag:display-dayView-overdueTask-Header] - if overdue tasks exist, then the overdue task header is identified by the resource bundle 'overdueTasks' value
	[tag:calendar-display-dayView-overdueTaskList] - overdue task list content
	[tag:display-dayView-dueTask-Header] - if tasks exist, then the task header is identified by the resource bundle 'dueTasks' value
	[tag:calendar-display-dayView-taskList] - task list content
	[tag:display-dayView-dueEvent-Header] - if events exist, then the event header is identified by the resource bundle 'dueEvents' value
	[tag:calendar-display-dayView-eventList] - event list content
	[tag:display-dayView-otherTask-Header] - if other tasks exist, then the other tasks header is identified by the resource bundle 'otherTasks' value
	[tag:calendar-display-dayView-otherTaskList] - other tasks list content
display-error.template - Used for displaying error messages	[tag:calendar-display-error] - error message

Table A-4 CalendarProvider Template Files *(Continued)*

CalendarProvider Template and Description	Tag Name and Description
<code>display-monthView-dayOfWeek.template</code> - Used for creating week layout within the month	<p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-monthView-dayOfWeek0] - day of week 0</p> <p>[tag:calendar-display-monthView-dayOfWeek1] - day of week 1</p> <p>[tag:calendar-display-monthView-dayOfWeek2] - day of week 2</p> <p>[tag:calendar-display-monthView-dayOfWeek3] - day of week 3</p> <p>[tag:calendar-display-monthView-dayOfWeek4] - day of week 4</p> <p>[tag:calendar-display-monthView-dayOfWeek5] - day of week 5</p> <p>[tag:calendar-display-monthView-dayOfWeek6] - day of week 6</p>
<code>display-monthView-emptyEventList.template</code> - Used for formatting when there are no events	No tags.
<code>display-monthView-emptyTaskList.template</code> - Used for formatting when there are no tasks	No tags.

Table A-4 CalendarProvider Template Files *(Continued)*

CalendarProvider Template and Description	Tag Name and Description
display-monthView-event.template - Used for formatting an event	<p>[tag:calendar-display-monthView-event-startHour1] - start hour for 0 based 12-hour clock</p> <p>[tag:calendar-display-monthView-event-endHour1] - end hour for 0 based 12-hour clock</p> <p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-event-conflict] - if the event is in conflict, then it is identified by the resource bundle 'conflict' value.</p> <p>[tag:calendar-display-monthView-event-startHour2] - start hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-monthView-event-startMinute] - minutes of the start time</p> <p>[tag:calendar-display-monthView-event-startAmPm] - am or pm identifier</p> <p>[tag:calendar-display-monthView-event-endHour2] - end hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-monthView-event-endMinute] - minutes of the end time</p> <p>[tag:calendar-display-monthView-event-endAmPm] - am or pm identifier</p> <p>[tag:calendar-display-monthView-event-summary] - event summary</p>
display-monthView-eventAllDay.template - Used for formatting an all day event	<p>[tag:calendar-display-monthView-event-allDay] - if the event is an All Day event, then it is identified by the resource bundle 'CalendarProvider-allDayEvent' value</p> <p>[tag:calendar-display-monthView-event-summary] - event summary</p>
display-monthView-task.template - Used for formatting a task	<p>[tag:calendar-display-monthView-task-pendHour2] - hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-monthView-task-pendMinute] - minutes of the task</p> <p>[tag:calendar-display-monthView-task-pendAmPm] - am or pm identifier</p> <p>[tag:calendar-display-monthView-task-summary] - task summary</p>

Table A-4 CalendarProvider Template Files *(Continued)*

CalendarProvider Template and Description	Tag Name and Description
display-monthView-weekView.template - Used for formatting the week view with a month	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:calendar-display-monthView-day0] - day in month 0
	[tag:calendar-display-monthView-day1] - day in month 1
	[tag:calendar-display-monthView-day2] - day in month 2
	[tag:calendar-display-monthView-day3] - day in month 3
	[tag:calendar-display-monthView-day4] - day in month 4
	[tag:calendar-display-monthView-day5] - day in month 5
	[tag:calendar-display-monthView-day6] - day in month 6
	[tag:calendar-display-monthView-eventList0] - events for day in month 0
	[tag:calendar-display-monthView-taskList0] - tasks for day in month 0
	[tag:calendar-display-monthView-eventList1] - events for day in month 1
	[tag:calendar-display-monthView-taskList1] - tasks for day in month 1
	[tag:calendar-display-monthView-eventList2] - events for day in month 2
	[tag:calendar-display-monthView-taskList2] - tasks for day in month 2
	[tag:calendar-display-monthView-eventList3] - events for day in month 3
	[tag:calendar-display-monthView-taskList3] - tasks for day in month 3
	[tag:calendar-display-monthView-eventList4] - events for day in month 4
	[tag:calendar-display-monthView-taskList4] - tasks for day in month 4
	[tag:calendar-display-monthView-eventList5] - events for day in month 5
	[tag:calendar-display-monthView-taskList5] - tasks for day in month 5
	[tag:calendar-display-monthView-eventList6] - events for day in month 6
	[tag:calendar-display-monthView-taskList6] - tasks for day in month 6

Table A-4 CalendarProvider Template Files *(Continued)*

CalendarProvider Template and Description	Tag Name and Description
display-monthView.template - Used for generating the entire month layout	[tag:calendar-display-monthView-currentDayOfWeek] - today's day of week
	[tag:calendar-display-monthView-currentMonth] - today's month
	[tag:calendar-display-monthView-currentDay] - today's day in month
	[tag:calendar-display-monthView-currentYear] - today's year
	[tag:calendar-display-monthView-dayOfWeek] - day of week content
	[tag:calendar-display-monthView-weekView0] - week content for week 0
	[tag:calendar-display-monthView-weekView1] - week content for week 1
	[tag:calendar-display-monthView-weekView2] - week content for week 2
	[tag:calendar-display-monthView-weekView3] - week content for week 3
[tag:calendar-display-monthView-weekView4] - week content for week 4	
display-summary-events.template - Used to show number of events	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif) [tag:calendar-display-summary-events] - event summary information
display-summary-tasks.template - Used to show number of tasks	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif) [tag:calendar-display-summary-tasks] - task summary information
display-summary.template - Used for formatting number of total and unread messages	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif) [tag:calendar-display-summary-events] - event summary [tag:calendar-display-summary-tasks] - task summary

Table A-4 CalendarProvider Template Files *(Continued)*

CalendarProvider Template and Description	Tag Name and Description
<code>display-weekView-currentDayHeader.template</code> - Used for formatting header for week view	<p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-weekView-currentDayOfWeek] - today's day of week</p> <p>[tag:calendar-display-weekView-currentMonth] - today's month</p> <p>[tag:calendar-display-weekView-currentDay] - today's day in month</p> <p>[tag:calendar-display-weekView-currentYear] - today's year</p>
<code>display-weekView-emptyEventList.template</code> - Used for formatting an empty event list	No tags.
<code>display-weekView-emptyTaskList.template</code> - Used for formatting an empty task list	No tags.
<code>display-weekView-event.template</code> - Used for formatting an event	<p>[tag:calendar-display-weekView-event-startHour1] - start hour for 0 based 12-hour clock</p> <p>[tag:calendar-display-weekView-event-endHour1] - end hour for 0 based 12-hour clock</p> <p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-event-conflict] - if the event is in conflict, then it is identified by the resource bundle 'conflict' value.</p> <p>[tag:calendar-display-weekView-event-startHour2] - start hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-weekView-event-startMinute] - minutes of the start time</p> <p>[tag:calendar-display-weekView-event-startAmPm] - am or pm identifier</p> <p>[tag:calendar-display-weekView-event-endHour2] - end hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-weekView-event-endMinute] - minutes of the end time</p> <p>[tag:calendar-display-weekView-event-endAmPm] - am or pm identifier</p> <p>[tag:calendar-display-weekView-event-summary] - event summary</p>

Table A-4 CalendarProvider Template Files *(Continued)*

CalendarProvider Template and Description	Tag Name and Description
display-weekView-eventAllDay.template - Used for formatting an all day event	[tag:calendar-display-weekView-event-allDay] - if the event is an All Day event, then it is identified by the resource bundle 'CalendarProvider-allDayEvent' value [tag:calendar-display-weekView-event-summary] - event summary
display-weekView-task.template - Used for formatting a task	[tag:calendar-display-weekView-task-pendHour2] - hour based on user preference of 12 or 24 hour clock format [tag:calendar-display-weekView-task-pendMinute] - minutes of the end time [tag:calendar-display-weekView-task-pendAmPm] - am or pm identifier [tag:calendar-display-weekView-task-summary] - task summary

Table A-4 CalendarProvider Template Files *(Continued)*

CalendarProvider Template and Description	Tag Name and Description
display-weekView.template - Used for the overall week view	[tag:calendar-display-weekView-currentDayHeader] - today's date information
	[tag:calendar-display-weekView-dayOfWeek0] - day of week 0
	[tag:calendar-display-weekView-dayOfWeek1] - day of week 1
	[tag:calendar-display-weekView-dayOfWeek2] - day of week 2
	[tag:calendar-display-weekView-dayOfWeek3] - day of week 3
	[tag:calendar-display-weekView-dayOfWeek4] - day of week 4
	[tag:calendar-display-weekView-dayOfWeek5] - day of week 5
	[tag:calendar-display-weekView-dayOfWeek6] - day of week 6
	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:calendar-display-weekView-day0] - day in week 0
	[tag:calendar-display-weekView-day1] - day in week 1
	[tag:calendar-display-weekView-day2] - day in week 2
	[tag:calendar-display-weekView-day3] - day in week 3
	[tag:calendar-display-weekView-day4] - day in week 4
	[tag:calendar-display-weekView-day5] - day in week 5
	[tag:calendar-display-weekView-day6] - day in week 6
	[tag:calendar-display-weekView-eventList0] - events for day in week 0
	[tag:calendar-display-weekView-taskList0] - tasks for day in week 0
	[tag:calendar-display-weekView-eventList1] - events for day in week 1
	[tag:calendar-display-weekView-taskList1] - tasks for day in week 1
	[tag:calendar-display-weekView-eventList2] - events for day in week 2
	[tag:calendar-display-weekView-taskList2] - tasks for day in week 2

Table A-4 CalendarProvider Template Files *(Continued)*

CalendarProvider Template and Description	Tag Name and Description
	<p>[tag:calendar-display-weekView-eventList3] - events for day in week 3</p> <p>[tag:calendar-display-weekView-taskList3] - tasks for day in week 3</p> <p>[tag:calendar-display-weekView-eventList4] - events for day in week 4</p> <p>[tag:calendar-display-weekView-taskList4] - tasks for day in week 4</p> <p>[tag:calendar-display-weekView-eventList5] - events for day in week 5</p> <p>[tag:calendar-display-weekView-taskList5] - tasks for day in week 5</p> <p>[tag:calendar-display-weekView-eventList6] - events for day in week 6</p> <p>[tag:calendar-display-weekView-taskList6] - tasks for day in week 6</p>
<p><code>display.template</code> - Used for overall channel formatting</p>	<p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:ErrorMessage] - error content</p> <p>[tag:calendar-display-dayView-summary] - summary content</p> <p>[tag:calendar-display-dayView] - day content</p> <p>[tag:calendar-display-weekView] - week content</p> <p>[tag:calendar-display-monthView] - month content</p> <p>[tag:calendar-display-clientURL] - client application url</p>
<p><code>edit-checkbox.template</code> - Used for creating edit page checkboxes</p>	<p>[tag:fontFace] - Font chosen</p> <p>[tag:fontColor] - Color of font chosen</p> <p>[tag:resourceName] - resource name</p> <p>[tag:header] - display value</p> <p>[tag:attName] - element name</p> <p>[tag:attSelected] - if the element is selected, then marks it as selected</p>
<p><code>edit-end.template</code> - Used for creating end of the edit page</p>	<p>No tags.</p>

Table A-4 CalendarProvider Template Files *(Continued)*

CalendarProvider Template and Description	Tag Name and Description
edit-link.template - Used for creating application helper edit link	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:editLink] - URL to edit
edit-password.template - Used for creating edit page password boxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName] - resource name [tag:header] - display value [tag:attName] - element name [tag:attValue] - element value
edit-select.template - Used for creating edit page select boxes	[tag:fontFace1] - Default font for the Desktop (sans-serif) [tag:fontColor] - Color of font chosen [tag:resourceName] - resource name [tag:header] - display value [tag:attName] - element name [tag:options] - element options [tag:attValue] - element value
edit-selectoption.template - Used for creating edit page select box options	[tag:selected] - if the element is selected, then marks it as selected [tag:string] - display value
edit-start.template - Used for creating the start of the edit page	[tag:process], [tag:isAppHandler], [tag:mail-display-error] - Used to dynamically build edit pages. Should not be edited. [tag:borderColor] - Color of the channel border. The border color can be changed by user in the desktop custom theme page. [tag:bgColor] - Provider background color chosen [tag:titlebarColor] - Color of title bar [tag:fontFace] - Font chosen

Table A-4 CalendarProvider Template Files *(Continued)*

CalendarProvider Template and Description	Tag Name and Description
edit-string.template - Used for creating edit page text boxes	[tag:fontFace] - Font chosen
	[tag:fontColor] - Color of font chosen
	[tag:header] - display value
	[tag:attName] - element name
	[tag:attValue] - element value
url.template - Used for creating hyperlinks	[tag:resourceName] - resource name
	[tag:link] - URL location
	[tag:name] - display value

error

Table A-5 lists the templates in the `error` subdirectory and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-5 Error Template Files

Error Template and Description	Tag Name and Description
banner.template - The banner across the top of the Desktop pages.	[surl:/images/spacer.gif] - Used to space gifs or channels
	[surl:/images/productName.gif] - Logo gif of the product name
	[surl:/images/blueBullet.gif] - Blue button to denote Home, Help, Logout, and so on
	[tag:staticContent] - Directory you defined for the deployment URI during installation.
	[tag:localeString] - Directory designation for chosen locale
	[tag:logoutUrl] - the logout URL
	[surl:/desktop/images/nothing.gif] - Used to space gifs or channels

Table A-5 Error Template Files *(Continued)*

Error Template and Description	Tag Name and Description
banner_nocontext.template	<p>[surl:/images/spacer.gif] - Used to space gifs or channels</p> <p>[surl:/images/productName.gif] - Logo gif of the product name</p> <p>[surl:/docs/en/desktop/usedesk.htm] - Help link for Desktop</p> <p>[surl:/images/blueBullet.gif] - Blue button to denote Home, Help, Logout, and so on</p> <p>[surl:/desktop/images/nothing.gif] - Used to space gifs or channels</p>
error.template	<p>[tag:bulletColor] - Inserts <code>bulletColor.js</code> into this template</p> <p>[tag:banner] - The <code>banner.template</code> file</p> <p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:staticContent] - Directory you defined for the deployment URI during installation.</p> <p>[tag:localeString] - Directory designation for chosen locale</p> <p>[tag:logoutUrl] - The logout URL</p> <p>[tag:stackTrace] - Produces a stack trace</p> <p>[tag:productName] - Product name</p> <p>[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates</p>
error_nocontext.template - Displayed when no context is available.	<p>[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates</p> <p>[tag:bulletColor] - Inserts <code>bulletColor.js</code> into this template</p> <p>[tag:banner] - Inserts <code>banner.template</code> into this template</p> <p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[surl:/docs/en/desktop/usedesk.htm] - Help link for Desktop</p> <p>[surl:/docs/en/desktop/fdesktop.htm] - Help link for frames on the Desktop</p> <p>[tag:stackTrace] - Produces a stack trace</p> <p>[tag:productName] - Product name</p>

Table A-5 Error Template Files *(Continued)*

Error Template and Description	Tag Name and Description
<p><code>noneditablechannel.template</code> - Template that will be used when there is an error in desktop when an edit page for a channel which is not editable is accessed. Displayed only when the user attempts to edit a channel which cannot be edited.</p>	<p>[tag:productName] - Product name</p> <p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:staticContent] - Directory you defined for the deployment URI during installation</p> <p>[tag:localeString] - Directory designation for chosen locale</p> <p>[tag:logoutUrl] - The logout URL</p> <p>[tag:stackTrace] - Produce a stack trace</p> <p>[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates</p>
<p><code>noprivilege.template</code> - Template that will be used when there is an error in desktop when a user with no privilege to access the desktop is trying to access the desktop. Displayed when a user who doesn't have the privilege to see the desktop attempts to access the desktop.</p>	<p>[tag:productName] - Product name</p> <p>[tag:bulletColor] - Inserts <code>bulletColor.js</code> into this template</p> <p>[tag:banner] - Inserts <code>banner.template</code> into this template</p> <p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:staticContent] - Directory you defined for the deployment URI during installation</p> <p>[tag:localeString] - Directory designation for chosen locale</p> <p>[tag:logoutUrl] - The logout URL</p> <p>[tag:stackTrace] - Produce a stack trace</p>
<p><code>unknownchannel.template</code> - Template that will be used when there is an error in desktop when an undefined channel is being accessed. Displayed when the user is trying to access a channel which is not defined in the system</p>	<p>[tag:productName] - Product name</p> <p>[tag:bulletColor] - Inserts <code>bulletColor.js</code> into this template</p> <p>[tag:banner] - Inserts <code>banner.template</code> into this template</p> <p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:staticContent] - Directory you defined for the deployment URI during installation</p> <p>[tag:localeString] - Directory designation for chosen locale</p> <p>-</p> <p>[tag:logoutUrl] - The logout URL</p> <p>[tag:stackTrace] - Produce a stack trace</p>

LoginProvider

[Table A-6 on page 331](#) lists the templates in the `LoginProvider` subdirectory and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-6 LoginProvider Template Files

Login Template and Description	Tag Name and Description
<code>display.template</code> - Contains the JavaScript code that launches the login window.	<p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:persistentCookie] - Inserts the persistent cookie template</p> <p>[tag:libertyLogin] - Inserts the <code>libertyLogin.Template</code></p> <p>[tag:loginHelpUrl] - Help link for login</p>
<code>display_AuthLDAP.template</code> - Contains the JavaScript code that launches the LDAP login window.	<p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:persistentCookie] - Inserts the persistent cookie template</p> <p>[tag:loginHelpUrl] - Help link for login</p>
<code>display_AuthUnix.template</code> - Contains the JavaScript code that launches the UNIX login window.	<p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:persistentCookie] - Inserts the persistent cookie template</p> <p>[tag:loginHelpUrl] - Help link for login</p>
<code>libertyLogin.template</code>	<p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:preLoginURL] - Inserts the liberty preLogin URL. The value is specified in the channel property <code>preLoginURL</code> which is typically of the form:</p> <pre>http://www.siroe.com:80/amserver/preLogin?metaAlias=www.siroe.com&goto=http://www.siroe.com:80/portal/dt</pre>
<code>persistentCookie.template</code> - Partial HTML template for remembering the user's name and password.	[tag:fontFace1] - Default font for the Desktop (sans-serif)

LotusNotesAddressBookProvider

[Table A-7](#) lists the templates in the `html` subdirectory of the `LotusNotesAddressBookProvider` and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-7 LotusNotesAddressBookProvider Template Files

LotusNotesAddressBookProvider Template and Description	Tag Name and Description
<code>display-clientURL.template</code> - Used for displaying the Launch Address Book link	[tag:ab-display-clientURL-appURL] - Used to display the application launch link
<code>display-entries.template</code> - Used for formatting the table headers	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:ab-display-entry-list] - Used as a placeholder to put in all of the address book entries
<code>display-entry.template</code> - Used for formatting the display of the address book entry	[tag:bgColor] - Provider background color chosen [tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:ab-display-entry-firstname] - Display first name [tag:ab-display-entry-lastname] - Display last name [tag:ab-display-entry-commonname] - Display common name [tag:ab-display-entry-email] - Display email [tag:ab-display-entry-email-link] - Display email link
<code>display-error.template</code> - Used for displaying error messages	[tag:ab-display-error] - Display error message
<code>display-summary.template</code> - Used for formatting number of total and unread messages	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:ab-display-summary-entries]
<code>display.template</code> - Used for overall channel formatting	[tag:ab-display-summary] - Display summary [tag:ab-display-entries] - Display entries [tag:ab-display-clientURL]
<code>edit-checkbox.template</code> - Used for creating edit page checkboxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:attSelected] - Used to dynamically build edit pages. Should not be edited.
<code>edit-end.template</code> - Used for creating end of the edit page	No tags.

Table A-7 LotusNotesAddressBookProvider Template Files *(Continued)*

LotusNotesAddressBookProvider Template and Description	Tag Name and Description
edit-link.template - Used for creating application helper edit link	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:editLink] - Used to display link for the application helper editing
edit-password.template - Used for creating edit page password boxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:attValue] - Used to dynamically build edit pages. Should not be edited.
edit-select.template - Used for creating edit page select boxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:options] - Used to dynamically build edit pages. Should not be edited.
edit-selectoption.template - Used for creating edit page select box options	[tag:attValue], [tag:selected], [tag:string] - Used to dynamically build edit pages. Should not be edited.
edit-start.template - Used for creating the start of the edit page	[tag:process], [tag:isAppHandler] - Used to dynamically build edit pages. Should not be edited. [tag:mail-display-error] - Displays error message [tag:borderColor] - Color of the channel border. The border color can be changed by user in the desktop custom theme page. [tag:bgColor] - Provider background color chosen [tag:titlebarColor] - Color of title bar [tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen
edit-string.template - Used for creating edit page text boxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:attValue] - Used to dynamically build edit pages. Should not be edited.

LotusNotesCalendarProvider

[Table A-8 on page 334](#) lists the templates in the `html` subdirectory of the LotusNotesCalendarProvider and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-8 LotusNotesCalendarProvider Template Files

LotusNotesCalendarProvider Template and Description	Tag Name and Description
display-clientURL.template - Used for displaying the Launch Calendar link.	[tag:calendar-display-client-uri] - Used to display the application launch link
display-dayView-emptyEventList.template - Used for displaying the message "No events scheduled for today."	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)
display-dayView-emptyTaskList.template - Used for displaying the message "No tasks are pending for today."	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)

Table A-8 LotusNotesCalendarProvider Template Files *(Continued)*

LotusNotesCalendarProvider Template and Description	Tag Name and Description
display-dayView-event.template - Used for formatting events	<p>[tag:calendar-display-dayView-event-startHourOfDay0] - start hour for 0 based 24-hour clock</p> <p>[tag:calendar-display-dayView-event-startHourOfDay1] - start hour for 1 based 24-hour clock</p> <p>[tag:calendar-display-dayView-event-startHour0] - start hour for 0 based 12-hour clock</p> <p>[tag:calendar-display-dayView-event-endHourOfDay0] - end hour for 0 based 24-hour clock</p> <p>[tag:calendar-display-dayView-event-endHourOfDay1] - end hour for 1 based 24-hour clock</p> <p>[tag:calendar-display-dayView-event-endHour0] - end hour for 0 based 12-hour clock</p> <p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-event-conflict] - if the event is in conflict, then it is identified by the resource bundle 'conflict' value.</p> <p>[tag:calendar-display-dayView-event-startHour2] - start hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-dayView-event-startMinute] - minutes of the start time</p> <p>[tag:calendar-display-dayView-event-startAmPm] - am or pm identifier</p> <p>[tag:calendar-display-dayView-event-endHour2] - end hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-dayView-event-endMinute] - minutes of the end time</p> <p>[tag:calendar-display-dayView-event-endAmPm] - am or pm identifier</p> <p>[tag:calendar-display-dayView-event-location] - event location</p> <p>[tag:calendar-display-dayView-event-summary] - event summary</p> <p>[tag:calendar-display-dayView-description-seperator] - description seperator specified by the resource bundle 'seperatorDescription' value</p>

Table A-8 LotusNotesCalendarProvider Template Files *(Continued)*

LotusNotesCalendarProvider Template and Description	Tag Name and Description
display-dayView-eventAllDay.template - Used for formatting an all day event	<p>[tag:calendar-display-dayView-event-description] - event description</p> <p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-dayView-event-allDay] - if the event is an All Day event, then it is identified by the resource bundle 'CalendarProvider-allDayEvent' value</p> <p>[tag:calendar-display-dayView-event-location] - event location</p> <p>[tag:calendar-display-dayView-event-summary] - event summary</p> <p>[tag:calendar-display-dayView-description-seperator] - description seperator specified by the resource bundle 'seperatorDescription' value</p>
display-dayView-otherTasks.template - Used for formatting other tasks	<p>[tag:calendar-display-dayView-event-description] - event description</p> <p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-dayView-task-summary] - task summary</p> <p>[tag:calendar-display-dayView-description-seperator] - description seperator specified by the resource bundle 'seperatorDescription' value</p> <p>[tag:calendar-display-dayView-task-description] - task description</p>

Table A-8 LotusNotesCalendarProvider Template Files *(Continued)*

LotusNotesCalendarProvider Template and Description	Tag Name and Description
display-dayView-overdueTasks.template - Used for formatting overdue tasks	<p>[tag:calendar-display-dayView-task-dueHour1] - hour of task for 12-hour clock</p> <p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-dayView-task-dueMonth] - month the task was due</p> <p>[tag:calendar-display-dayView-task-dueDay] - day of the month the task was due</p> <p>[tag:calendar-display-dayView-task-dueYear] - year the task was due</p> <p>[tag:calendar-display-dayView-task-dueHour2] - end hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-dayView-task-dueMinute] - minutes of the due time</p> <p>[tag:calendar-display-dayView-task-dueAmPm] - am or pm identifier</p> <p>[tag:calendar-display-dayView-overdueTask-summary] - overdue task summary</p> <p>[tag:calendar-display-dayView-description-seperator] - description seperator specified by the resource bundle 'seperatorDescription' value</p> <p>[tag:calendar-display-dayView-overdueTask-description] - task description</p>

Table A-8 LotusNotesCalendarProvider Template Files *(Continued)*

LotusNotesCalendarProvider Template and Description	Tag Name and Description
display-dayView-task.template - Used for formatting a "normal" task	<p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-dayView-task-pendHour2] - hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-dayView-task-pendMinute] - minutes of the task</p> <p>[tag:calendar-display-dayView-task-pendAmPm] - am or pm identifier</p> <p>[tag:calendar-display-dayView-task-location] - task location</p> <p>[tag:calendar-display-dayView-task-complete-start] - start time of completed task</p> <p>[tag:calendar-display-dayView-task-summary] - task summary</p> <p>[tag:calendar-display-dayView-description-seperator] - description seperator specified by the resource bundle 'seperatorDescription' value</p> <p>[tag:calendar-display-dayView-task-description] - task description</p>

Table A-8 LotusNotesCalendarProvider Template Files *(Continued)*

LotusNotesCalendarProvider Template and Description	Tag Name and Description
display-dayView.template - Used for formatting the layout of all tasks and events	[tag:calendar-display-dayView-dayOfWeek] - today's day of week
	[tag:calendar-display-dayView-month] - today's month
	[tag:calendar-display-dayView-day] - today's day in month
	[tag:calendar-display-dayView-year] - today's year
	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:calendar-display-dayView-overdueTaskNum] - number of overdue tasks for today
	[tag:display-dayView-overdueTask-Header] - if overdue tasks exist, then the overdue task header is identified by the resource bundle 'overdueTasks' value
	[tag:calendar-display-dayView-overdueTaskList] - overdue task list content
	[tag:display-dayView-dueTask-Header] - if tasks exist, then the task header is identified by the resource bundle 'dueTasks' value
	[tag:calendar-display-dayView-taskList] - task list content
	[tag:display-dayView-dueEvent-Header] - if events exist, then the event header is identified by the resource bundle 'dueEvents' value
	[tag:calendar-display-dayView-eventList] - event list content
	[tag:display-dayView-otherTask-Header] - if other tasks exist, then the other tasks header is identified by the resource bundle 'otherTasks' value
	[tag:calendar-display-dayView-otherTaskList] - other tasks list content
display-error.template - Used for displaying error messages	[tag:calendar-display-error] - error message

Table A-8 LotusNotesCalendarProvider Template Files *(Continued)*

LotusNotesCalendarProvider Template and Description	Tag Name and Description
display-monthView-dayOfWeek.template - Used for creating week layout within the month	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif) [tag:calendar-display-monthView-dayOfWeek0] - day of week 0 [tag:calendar-display-monthView-dayOfWeek1] - day of week 1 [tag:calendar-display-monthView-dayOfWeek2] - day of week 2 [tag:calendar-display-monthView-dayOfWeek3] - day of week 3 [tag:calendar-display-monthView-dayOfWeek4] - day of week 4 [tag:calendar-display-monthView-dayOfWeek5] - day of week 5 [tag:calendar-display-monthView-dayOfWeek6] - day of week 6
display-monthView-emptyEventList.template - Used for formatting when there are no events	No tags.
display-monthView-emptyTaskList.template - Used for formatting when there are no tasks	No tags.

Table A-8 LotusNotesCalendarProvider Template Files *(Continued)*

LotusNotesCalendarProvider Template and Description	Tag Name and Description
display-monthView-event.template - Used for formatting an event	<p>[tag:calendar-display-monthView-event-startHour1] - start hour for 0 based 12-hour clock</p> <p>[tag:calendar-display-monthView-event-endHour1] - end hour for 0 based 12-hour clock</p> <p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-event-conflict] - if the event is in conflict, then it is identified by the resource bundle 'conflict' value.</p> <p>[tag:calendar-display-monthView-event-startHour2] - start hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-monthView-event-startMinute] - minutes of the start time</p> <p>[tag:calendar-display-monthView-event-startAmPm] - am or pm identifier</p> <p>[tag:calendar-display-monthView-event-endHour2] - end hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-monthView-event-endMinute] - minutes of the end time</p> <p>[tag:calendar-display-monthView-event-endAmPm] - am or pm identifier</p> <p>[tag:calendar-display-monthView-event-summary] - event summary</p>
display-monthView-eventAllDay.template - Used for formatting an all day event	<p>[tag:calendar-display-monthView-event-allDay] - if the event is an All Day event, then it is identified by the resource bundle 'CalendarProvider-allDayEvent' value</p> <p>[tag:calendar-display-monthView-event-summary] - event summary</p>
display-monthView-task.template - Used for formatting a task	<p>[tag:calendar-display-monthView-task-pendHour2] - hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-monthView-task-pendMinute] - minutes of the task</p> <p>[tag:calendar-display-monthView-task-pendAmPm] - am or pm identifier</p> <p>[tag:calendar-display-monthView-task-summary] - task summary</p>

Table A-8 LotusNotesCalendarProvider Template Files *(Continued)*

LotusNotesCalendarProvider Template and Description	Tag Name and Description
display-monthView-weekView.template - Used for formatting the week view with a month	<p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-monthView-day0] - day in month 0</p> <p>[tag:calendar-display-monthView-day1] - day in month 1</p> <p>[tag:calendar-display-monthView-day2] - day in month 2</p> <p>[tag:calendar-display-monthView-day3] - day in month 3</p> <p>[tag:calendar-display-monthView-day4] - day in month 4</p> <p>[tag:calendar-display-monthView-day5] - day in month 5</p> <p>[tag:calendar-display-monthView-day6] - day in month 6</p> <p>[tag:calendar-display-monthView-eventList0] - events for day in month 0</p> <p>[tag:calendar-display-monthView-taskList0] - tasks for day in month 0</p> <p>[tag:calendar-display-monthView-eventList1] - events for day in month 1</p> <p>[tag:calendar-display-monthView-taskList1] - tasks for day in month 1</p> <p>[tag:calendar-display-monthView-eventList2] - events for day in month 2</p> <p>[tag:calendar-display-monthView-taskList2] - tasks for day in month 2</p> <p>[tag:calendar-display-monthView-eventList3] - events for day in month 3</p> <p>[tag:calendar-display-monthView-taskList3] - tasks for day in month 3</p> <p>[tag:calendar-display-monthView-eventList4] - events for day in month 4</p> <p>[tag:calendar-display-monthView-taskList4] - tasks for day in month 4</p> <p>[tag:calendar-display-monthView-eventList5] - events for day in month 5</p> <p>[tag:calendar-display-monthView-taskList5] - tasks for day in month 5</p> <p>[tag:calendar-display-monthView-eventList6] - events for day in month 6</p> <p>[tag:calendar-display-monthView-taskList6] - tasks for day in month 6</p>

Table A-8 LotusNotesCalendarProvider Template Files *(Continued)*

LotusNotesCalendarProvider Template and Description	Tag Name and Description
display-monthView.template - Used for generating the entire month layout	[tag:calendar-display-monthView-currentDayOfWeek] - today's day of week [tag:calendar-display-monthView-currentMonth] - today's month [tag:calendar-display-monthView-currentDay] - today's day in month [tag:calendar-display-monthView-currentYear] - today's year [tag:calendar-display-monthView-dayOfWeek] - day of week content [tag:calendar-display-monthView-weekView0] - week content for week 0 [tag:calendar-display-monthView-weekView1] - week content for week 1 [tag:calendar-display-monthView-weekView2] - week content for week 2 [tag:calendar-display-monthView-weekView3] - week content for week 3 [tag:calendar-display-monthView-weekView4] - week content for week 4
display-summary-events.template - Used to show number of events	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif) [tag:calendar-display-summary-events] - event summary information
display-summary-tasks.template - Used to show number of tasks	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif) [tag:calendar-display-summary-tasks] - task summary information
display-summary.template - Used for formatting number of total and unread messages	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif) [tag:calendar-display-summary-events] - event summary [tag:calendar-display-summary-tasks] - task summary

Table A-8 LotusNotesCalendarProvider Template Files *(Continued)*

LotusNotesCalendarProvider Template and Description	Tag Name and Description
display-weekView-currentDayHeader.template - Used for formatting header for week view	<p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-weekView-currentDayOfWeek] - today's day of week</p> <p>[tag:calendar-display-weekView-currentMonth] - today's month</p> <p>[tag:calendar-display-weekView-currentDay] - today's day in month</p> <p>[tag:calendar-display-weekView-currentYear] - today's year</p>
display-weekView-emptyEventList.template - Used for formatting an empty event list	No tags.
display-weekView-emptyTaskList.template - Used for formatting an empty task list	No tags.

Table A-8 LotusNotesCalendarProvider Template Files *(Continued)*

LotusNotesCalendarProvider Template and Description	Tag Name and Description
display-weekView-event.template - Used for formatting an event	<p>[tag:calendar-display-weekView-event-startHour1] - start hour for 0 based 12-hour clock</p> <p>[tag:calendar-display-weekView-event-endHour1] - end hour for 0 based 12-hour clock</p> <p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-event-conflict] - if the event is in conflict, then it is identified by the resource bundle 'conflict' value.</p> <p>[tag:calendar-display-weekView-event-startHour2] - start hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-weekView-event-startMinute] - minutes of the start time</p> <p>[tag:calendar-display-weekView-event-startAmPm] - am or pm identifier</p> <p>[tag:calendar-display-weekView-event-endHour2] - end hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-weekView-event-endMinute] - minutes of the end time</p> <p>[tag:calendar-display-weekView-event-endAmPm] - am or pm identifier</p> <p>[tag:calendar-display-weekView-event-summary] - event summary</p>
display-weekView-eventAllDay.template - Used for formatting an all day event	<p>[tag:calendar-display-weekView-event-allDay] - if the event is an All Day event, then it is identified by the resource bundle 'CalendarProvider-allDayEvent' value</p> <p>[tag:calendar-display-weekView-event-summary] - event summary</p>
display-weekView-task.template - Used for formatting a task	<p>[tag:calendar-display-weekView-task-pendHour2] - hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-weekView-task-pendMinute] - minutes of the end time</p> <p>[tag:calendar-display-weekView-task-pendAmPm] - am or pm identifier</p> <p>[tag:calendar-display-weekView-task-summary] - task summary</p>

Table A-8 LotusNotesCalendarProvider Template Files *(Continued)*

LotusNotesCalendarProvider Template and Description	Tag Name and Description
display-weekView.template - Used for the overall week view	<p>[tag:calendar-display-weekView-currentDayHeader] - today's date information</p> <p>[tag:calendar-display-weekView-dayOfWeek0] - day of week 0</p> <p>[tag:calendar-display-weekView-dayOfWeek1] - day of week 1</p> <p>[tag:calendar-display-weekView-dayOfWeek2] - day of week 2</p> <p>[tag:calendar-display-weekView-dayOfWeek3] - day of week 3</p> <p>[tag:calendar-display-weekView-dayOfWeek4] - day of week 4</p> <p>[tag:calendar-display-weekView-dayOfWeek5] - day of week 5</p> <p>[tag:calendar-display-weekView-dayOfWeek6] - day of week 6</p> <p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-weekView-day0] - day in week 0</p> <p>[tag:calendar-display-weekView-day1] - day in week 1</p> <p>[tag:calendar-display-weekView-day2] - day in week 2</p> <p>[tag:calendar-display-weekView-day3] - day in week 3</p> <p>[tag:calendar-display-weekView-day4] - day in week 4</p> <p>[tag:calendar-display-weekView-day5] - day in week 5</p> <p>[tag:calendar-display-weekView-day6] - day in week 6</p> <p>[tag:calendar-display-weekView-eventList0] - events for day in week 0</p> <p>[tag:calendar-display-weekView-taskList0] - tasks for day in week 0</p> <p>[tag:calendar-display-weekView-eventList1] - events for day in week 1</p> <p>[tag:calendar-display-weekView-taskList1] - tasks for day in week 1</p> <p>[tag:calendar-display-weekView-eventList2] - events for day in week 2</p> <p>[tag:calendar-display-weekView-taskList2] - tasks for day in week 2</p>

Table A-8 LotusNotesCalendarProvider Template Files *(Continued)*

LotusNotesCalendarProvider Template and Description	Tag Name and Description
<p><code>display.template</code> - Used for overall channel formatting</p> <p><code>edit-checkbox.template</code> - Used for creating edit page checkboxes</p> <p><code>edit-end.template</code> - Used for creating end of the edit page</p>	[tag:calendar-display-weekView-eventList3] - events for day in week 3
	[tag:calendar-display-weekView-taskList3] - tasks for day in week 3
	[tag:calendar-display-weekView-eventList4] - events for day in week 4
	[tag:calendar-display-weekView-taskList4] - tasks for day in week 4
	[tag:calendar-display-weekView-eventList5] - events for day in week 5
	[tag:calendar-display-weekView-taskList5] - tasks for day in week 5
	[tag:calendar-display-weekView-eventList6] - events for day in week 6
	[tag:calendar-display-weekView-taskList6] - tasks for day in week 6
	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:ErrorMessage] - error content
	[tag:calendar-display-dayView-summary] - summary content
	[tag:calendar-display-dayView] - day content
	[tag:calendar-display-weekView] - week content
	[tag:calendar-display-monthView] - month content
[tag:calendar-display-clientURL] - client application url	
[tag:fontFace] - Font chosen	
[tag:fontColor] - Color of font chosen	
[tag:resourceName] - resource name	
[tag:header] - display value	
[tag:attName] - element name	
[tag:attSelected] - if the element is selected, then marks it as selected	
No tags.	

Table A-8 LotusNotesCalendarProvider Template Files *(Continued)*

LotusNotesCalendarProvider Template and Description	Tag Name and Description
edit-link.template - Used for creating application helper edit link	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:editLink] - URL to edit
edit-password.template - Used for creating edit page password boxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName] - resource name [tag:header] - display value [tag:attName] - element name [tag:attValue] - element value
edit-select.template - Used for creating edit page select boxes	[tag:fontFace1] - Default font for the Desktop (sans-serif) [tag:fontColor] - Color of font chosen [tag:resourceName] - resource name [tag:header] - display value [tag:attName] - element name [tag:options] - element options
edit-selectoption.template - Used for creating edit page select box options	[tag:attValue] - element value [tag:selected] - if the element is selected, then marks it as selected [tag:string] - display value
edit-start.template - Used for creating the start of the edit page	[tag:process], [tag:isAppHandler], [tag:mail-display-error] - Used to dynamically build edit pages. Should not be edited. [tag:borderColor] - Color of the channel border. The border color can be changed by user in the desktop custom theme page. [tag:bgColor] - Provider background color chosen [tag:titlebarColor] - Color of title bar [tag:fontFace] - Font chosen

Table A-8 LotusNotesCalendarProvider Template Files *(Continued)*

LotusNotesCalendarProvider Template and Description	Tag Name and Description
edit-string.template - Used for creating edit page text boxes	[tag:fontFace] - Font chosen
	[tag:fontColor] - Color of font chosen
	[tag:header] - display value
	[tag:attName] - element name
	[tag:attValue] - element value
url.template - Used for creating hyperlinks	[tag:resourceName] - resource name
	[tag:link] - URL location
	[tag:name] - display value

LotusNotesMailProvider

[Table A-9](#) lists the templates in the `html` subdirectory of the LotusNotesMailProvider and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-9 LotusNotesMailProvider Template Files

Template and Description	Tag Name and Description
display-clientURL.template - Used for displaying the Launch Mail link	[tag:mail-display-clientURL-uri] - Used to display application launch link
display-error.template - Used for displaying error messages	[tag:fontFace] - Font chosen
	[tag:mail-display-error] - Display error message
display-headers-message.template - Used for formatting individual message information	[tag:fontFace] - Font chosen
	[tag:bgColor] - Provider background color chosen
	[tag:fontColor] - Color of font chosen
	[tag:mail-display-headers-message-subject] - mail subject
	[tag:mail-display-headers-message-mailto] - Mail senders
	[tag:mail-display-headers-message-name] - Mail sender's name
	[tag:mail-display-headers-message-date] - Date message received
[tag:mail-display-headers-message-status] - Message status	

Table A-9 LotusNotesMailProvider Template Files *(Continued)*

Template and Description	Tag Name and Description
display-headers.template - Used for formatting message table headings	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:mail-display-headers-subject] - Displays subject [tag:mail-display-headers-from] - Displays From [tag:mail-display-headers-date] - Displays Date [tag:mail-display-headers-status] - Displays status
display-summary.template - Used for formatting number of total and unread messages	[tag:fontFace] - Font chosen [tag:mail-display-summary-unread] - displays number of unread messages [tag:mail-display-summary-total] - displays total number of messages
display.template - Used for overall channel formatting	[tag:mail-display-summary] - Displays the summary info [tag:mail-display-headers] - displays the messages headers [tag:mail-display-clientURL] - displays the application launch URL
edit-checkbox.template - Used for creating edit page checkboxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:attSelected] - Used to dynamically build edit pages. Should not be edited.
edit-end.template - Used for creating end of the edit page	No tags
edit-link.template - Used for creating application helper edit link	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:editLink] - Used to display link for the application helper editing
edit-password.template - Used for creating edit page password boxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:attValue] - Used to display link for the application helper editing

Table A-9 LotusNotesMailProvider Template Files *(Continued)*

Template and Description	Tag Name and Description
<code>edit-select.template</code> - Used for creating edit page select boxes	[tag:fontFace1] - Default font for the Desktop (sans-serif) [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:options] - Used to display link for the application helper editing
<code>edit-selectoption.template</code> - Used for creating edit page select box options	[tag:attValue], [tag:selected], [tag:string] - Used to dynamically build edit pages. Should not be edited.
<code>edit-start.template</code> - Used for creating the start of the edit page	[tag:process], [tag:isAppHandler] - Used to dynamically build edit pages. Should not be edited. [tag:mail-display-error] - Display error message [tag:bgColor] - Provider background color chosen [tag:titlebarColor] - Color of title bar [tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen
<code>edit-string.template</code> - Used for creating edit page text boxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:attValue] - Used to dynamically build edit pages. Should not be edited.
<code>url.template</code> - Used for creating hyperlinks	[tag:link, [tag:name] - Used to dynamically build edit pages. Should not be edited.

MailCheckProvider

Table A-10 lists the templates in the `MailCheckProvider` subdirectory and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-10 MailCheckProvider Template Files

Mailcheck Template and Description	Tag Name and Description
<code>display.template</code> - Contains MailCheck provider layout content.	[tag:mailCheckErrorMessages] - Contains error messages if the user has not filled in correct values for the IMAP Server Name, IMAP User and IMAP Password. [tag:mailCheckContents] - Displays the result of connecting to the given server with the user ID and password. The message could be "Mail Server is alive, 0 unread message(s)."
<code>edit.template</code> - Contains the formatting for the Edit page for the MailCheck provider.	[tag:fontFace1] - Default font for the Desktop (sans-serif) [tag:IMAPServerName] - Name of the IMAP mail server [tag:IMAPUserId] - user ID [tag:IMAPPassword] - Password for the mail server

MailProvider

Table A-11 lists the templates in the `html` subdirectory of the MailProvider and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-11 MailProvider Template Files

MailProvider Template and Description	Tag Name and Description
<code>display-clientURL.template</code> - Used for displaying the Launch Mail link	[tag:mail-display-clientURL-uri] - Used to display application launch link
<code>display-error.template</code> - Used for displaying error messages	[tag:fontFace] - Font chosen [tag:mail-display-error] - Display error message

Table A-11 MailProvider Template Files *(Continued)*

MailProvider Template and Description	Tag Name and Description
display-headers-message.template - Used for formatting individual message information	[tag:fontFace] - Font chosen [tag:bgColor] - Provider background color chosen [tag:fontColor] - Color of font chosen [tag:mail-display-headers-message-subject] - mail subject [tag:mail-display-headers-message-mailto] - Mail senders [tag:mail-display-headers-message-name] - Mail sender's name [tag:mail-display-headers-message-date] - Date message received [tag:mail-display-headers-message-status] - Message status
display-headers.template - Used for formatting message table headings	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:mail-display-headers-subject] - Displays subject [tag:mail-display-headers-from] - Displays From [tag:mail-display-headers-date] - Displays Date [tag:mail-display-headers-status] - Displays status
display-summary.template - Used for formatting number of total and unread messages	[tag:fontFace] - Font chosen [tag:mail-display-summary-unread] - displays number of unread messages [tag:mail-display-summary-total] - displays total number of messages
display.template - Used for overall channel formatting	[tag:mail-display-summary] - Displays the summary info [tag:mail-display-headers] - displays the messages headers [tag:mail-display-clientURL] - displays the application launch URL
edit-checkbox.template - Used for creating edit page checkboxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:attSelected] - Used to dynamically build edit pages. Should not be edited.
edit-end.template - Used for creating end of the edit page	No tags

Table A-11 MailProvider Template Files *(Continued)*

MailProvider Template and Description	Tag Name and Description
<code>edit-link.template</code> - Used for creating application helper edit link	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:editLink] - Used to display link for the application helper editing
<code>edit-password.template</code> - Used for creating edit page password boxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:attValue] - Used to display link for the application helper editing
<code>edit-select.template</code> - Used for creating edit page select boxes	[tag:fontFace1] - Default font for the Desktop (sans-serif) [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:options] - Used to display link for the application helper editing
<code>edit-selectoption.template</code> - Used for creating edit page select box options	[tag:attValue], [tag:selected], [tag:string] - Used to dynamically build edit pages. Should not be edited.
<code>edit-start.template</code> - Used for creating the start of the edit page	[tag:process], [tag:isAppHandler] - Used to dynamically build edit pages. Should not be edited. [tag:mail-display-error] - Display error message [tag:bgColor] - Provider background color chosen [tag:titlebarColor] - Color of title bar [tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen
<code>edit-string.template</code> - Used for creating edit page text boxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:attValue] - Used to dynamically build edit pages. Should not be edited.
<code>url.template</code> - Used for creating hyperlinks	[tag:link, [tag:name] - Used to dynamically build edit pages. Should not be edited.

MSEXchangeAddressBookProvider

Table A-15 lists the templates in the `html` subdirectory of the `MSEXchangeAddressBookProvider` and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-12 MSEXchangeAddressBookProvider Template Files

Template File and description	Tag Name and Description
<code>display-clientURL.template</code> - Used for displaying the "Launch Address Book" link	[tag:ab-display-clientURL-appURL] - Used to display the application launch link
<code>display-entries.template</code> - Used for formatting the table headers	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:ab-display-entry-list] - Used as a placeholder to put in all of the address book entries
<code>display-entry.template</code> - Used for formatting the display of the address book entry	[tag:bgColor] - Provider background color chosen [tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:ab-display-entry-firstname] - Displays first name [tag:ab-display-entry-lastname] - Displays last name [tag:ab-display-entry-commonname] - Displays common name [tag:ab-display-entry-email] - Displays email [tag:ab-display-entry-email-link] - Displays email as a link
<code>display-error.template</code> - Used for displaying error messages	[tag:ab-display-error] - Displays error message
<code>display-summary.template</code> - Used for formatting number of total and unread messages	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:ab-display-summary-entries] - Displays a summary
<code>display.template</code> - Used for overall channel formatting	[tag:ab-display-summary] - Displays summary [tag:ab-display-entries] - Displays entries [tag:ab-display-clientURL]
<code>edit-checkbox.template</code> - Used for creating edit page checkboxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:attSelected] - Used to dynamically build edit pages. Should not be edited.

Table A-12 MExchangeAddressBookProvider Template Files *(Continued)*

Template File and description	Tag Name and Description
edit-end.template - Used for creating end of the edit page	No tags.
edit-link.template - Used for creating application helper edit link	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:editLink] - Used to display link for the application helper editing
edit-password.template - Used for creating edit page password boxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:attValue] - Used to dynamically build edit pages. Should not be edited.
edit-select.template - Used for creating edit page select boxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:options] - Used to dynamically build edit pages. Should not be edited.
edit-selectoption.template - Used for creating edit page select box options	[tag:attValue], [tag:selected], [tag:string] - Used to dynamically build edit pages. Should not be edited.
edit-start.template - Used for creating the start of the edit page	[tag:process], [tag:isAppHandler] - Used to dynamically build edit pages. Should not be edited. [tag:mail-display-error] - Displays error message [tag:borderColor] - Color of the channel border. The border color can be changed by user in the desktop custom theme page. [tag:bgColor] - Provider background color chosen [tag:titlebarColor] - Color of title bar [tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen
edit-string.template - Used for creating edit page text boxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:attValue] - Used to dynamically build edit pages. Should not be edited.

MSExchangeCalendarProvider

Table A-13 lists the templates in the `html` subdirectory of the `MSExchangeCalendarProvider` and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-13 MSExchangeCalendarProvider Template Files

Template File and Description	Tag Name and Description
<code>display-clientURL.template</code> - Used for displaying the Launch Calendar link.	<code>[tag:calendar-display-client-uri]</code> - Used to display the application launch link
<code>display-dayView-emptyEventList.template</code> - Used for displaying the message "No events scheduled for today."	<code>[tag:iwtDesktop-fontFace1]</code> - Default font for the Desktop (sans-serif)
<code>display-dayView-emptyTaskList.template</code> - Used for displaying the message "No tasks are pending for today."	<code>[tag:iwtDesktop-fontFace1]</code> - Default font for the Desktop (sans-serif)

Table A-13 MExchangeCalendarProvider Template Files *(Continued)*

Template File and Description	Tag Name and Description
display-dayView-event.template - Used for formatting events	[tag:calendar-display-dayView-event-startHourOfDay0] - start hour for 0 based 24-hour clock
	[tag:calendar-display-dayView-event-startHourOfDay1] - start hour for 1 based 24-hour clock
	[tag:calendar-display-dayView-event-startHour0] - start hour for 0 based 12-hour clock
	[tag:calendar-display-dayView-event-endHourOfDay0] - end hour for 0 based 24-hour clock
	[tag:calendar-display-dayView-event-endHourOfDay1] - end hour for 1 based 24-hour clock
	[tag:calendar-display-dayView-event-endHour0] - end hour for 0 based 12-hour clock
	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:calendar-display-event-conflict] - if the event is in conflict, then it is identified by the resource bundle 'conflict' value.
	[tag:calendar-display-dayView-event-startHour2] - start hour based on user preference of 12 or 24 hour clock format
	[tag:calendar-display-dayView-event-startMinute] - minutes of the start time
	[tag:calendar-display-dayView-event-startAmPm] - am or pm identifier
	[tag:calendar-display-dayView-event-endHour2] - end hour based on user preference of 12 or 24 hour clock format
	[tag:calendar-display-dayView-event-endMinute] - minutes of the end time
	[tag:calendar-display-dayView-event-endAmPm] - am or pm identifier
	[tag:calendar-display-dayView-event-location] - event location
[tag:calendar-display-dayView-event-summary] - event summary	
[tag:calendar-display-dayView-description-seperator] - description seperator specified by the resource bundle 'seperatorDescription' value	
[tag:calendar-display-dayView-event-description] - event description	

Table A-13 MExchangeCalendarProvider Template Files *(Continued)*

Template File and Description	Tag Name and Description
display-dayView-eventAllDay.template - Used for formatting an all day event	<p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-dayView-event-allDay] - if the event is an All Day event, then it is identified by the resource bundle 'CalendarProvider-allDayEvent' value</p> <p>[tag:calendar-display-dayView-event-location] - event location</p> <p>[tag:calendar-display-dayView-event-summary] - event summary</p> <p>[tag:calendar-display-dayView-description-seperator] - description seperator specified by the resource bundle 'seperatorDescription' value</p> <p>[tag:calendar-display-dayView-event-description] - event description</p>
display-dayView-otherTasks.template - Used for formatting other tasks	<p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-dayView-task-summary] - task summary</p> <p>[tag:calendar-display-dayView-description-seperator] - description seperator specified by the resource bundle 'seperatorDescription' value</p> <p>[tag:calendar-display-dayView-task-description] - task description</p>

Table A-13 MExchangeCalendarProvider Template Files *(Continued)*

Template File and Description	Tag Name and Description
display-dayView-overdueTasks.template - Used for formatting overdue tasks	<p>[tag:calendar-display-dayView-task-dueHour1] - hour of task for 12-hour clock</p> <p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-dayView-task-dueMonth] - month the task was due</p> <p>[tag:calendar-display-dayView-task-dueDay] - day of the month the task was due</p> <p>[tag:calendar-display-dayView-task-dueYear] - year the task was due</p> <p>[tag:calendar-display-dayView-task-dueHour2] - end hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-dayView-task-dueMinute] - minutes of the due time</p> <p>[tag:calendar-display-dayView-task-dueAmPm] - am or pm identifier</p> <p>[tag:calendar-display-dayView-overdueTask-summary] - overdue task summary</p> <p>[tag:calendar-display-dayView-description-seperator] - description seperator specified by the resource bundle 'seperatorDescription' value</p> <p>[tag:calendar-display-dayView-overdueTask-description] - task description</p>

Table A-13 MExchangeCalendarProvider Template Files *(Continued)*

Template File and Description	Tag Name and Description
display-dayView-task.template - Used for formatting a "normal" task	<p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-dayView-task-pendHour2] - hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-dayView-task-pendMinute] - minutes of the task</p> <p>[tag:calendar-display-dayView-task-pendAmPm] - am or pm identifier</p> <p>[tag:calendar-display-dayView-task-location] - task location</p> <p>[tag:calendar-display-dayView-task-complete-start] - start time of completed task</p> <p>[tag:calendar-display-dayView-task-summary] - task summary</p> <p>[tag:calendar-display-dayView-description-seperator] - description seperator specified by the resource bundle 'seperatorDescription' value</p> <p>[tag:calendar-display-dayView-task-description] - task description</p>

Table A-13 MExchangeCalendarProvider Template Files *(Continued)*

Template File and Description	Tag Name and Description
display-dayView.template - Used for formatting the layout of all tasks and events	[tag:calendar-display-dayView-dayOfWeek] - today's day of week
	[tag:calendar-display-dayView-month] - today's month
	[tag:calendar-display-dayView-day] - today's day in month
	[tag:calendar-display-dayView-year] - today's year
	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:calendar-display-dayView-overdueTaskNum] - number of overdue tasks for today
	[tag:display-dayView-overdueTask-Header] - if overdue tasks exist, then the overdue task header is identified by the resource bundle 'overdueTasks' value
	[tag:calendar-display-dayView-overdueTaskList] - overdue task list content
	[tag:display-dayView-dueTask-Header] - if tasks exist, then the task header is identified by the resource bundle 'dueTasks' value
	[tag:calendar-display-dayView-taskList] - task list content
	[tag:display-dayView-dueEvent-Header] - if events exist, then the event header is identified by the resource bundle 'dueEvents' value
	[tag:calendar-display-dayView-eventList] - event list content
	[tag:display-dayView-otherTask-Header] - if other tasks exist, then the other tasks header is identified by the resource bundle 'otherTasks' value
[tag:calendar-display-dayView-otherTaskList] - other tasks list content	
display-error.template - Used for displaying error messages	[tag:calendar-display-error] - error message

Table A-13 MExchangeCalendarProvider Template Files *(Continued)*

Template File and Description	Tag Name and Description
display-monthView-dayOfWeek.template - Used for creating week layout within the month	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:calendar-display-monthView-dayOfWeek0] - day of week 0
	[tag:calendar-display-monthView-dayOfWeek1] - day of week 1
	[tag:calendar-display-monthView-dayOfWeek2] - day of week 2
	[tag:calendar-display-monthView-dayOfWeek3] - day of week 3
	[tag:calendar-display-monthView-dayOfWeek4] - day of week 4
	[tag:calendar-display-monthView-dayOfWeek5] - day of week 5
[tag:calendar-display-monthView-dayOfWeek6] - day of week 6	
display-monthView-emptyEventList.template - Used for formatting when there are no events	No tags.
display-monthView-emptyTaskList.template - Used for formatting when there are no tasks	No tags.

Table A-13 MExchangeCalendarProvider Template Files *(Continued)*

Template File and Description	Tag Name and Description
display-monthView-event.template - Used for formatting an event	<p>[tag:calendar-display-monthView-event-startHour1] - start hour for 0 based 12-hour clock</p> <p>[tag:calendar-display-monthView-event-endHour1] - end hour for 0 based 12-hour clock</p> <p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-event-conflict] - if the event is in conflict, then it is identified by the resource bundle 'conflict' value.</p> <p>[tag:calendar-display-monthView-event-startHour2] - start hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-monthView-event-startMinute] - minutes of the start time</p> <p>[tag:calendar-display-monthView-event-startAmPm] - am or pm identifier</p> <p>[tag:calendar-display-monthView-event-endHour2] - end hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-monthView-event-endMinute] - minutes of the end time</p> <p>[tag:calendar-display-monthView-event-endAmPm] - am or pm identifier</p> <p>[tag:calendar-display-monthView-event-summary] - event summary</p>
display-monthView-eventAllDay.template - Used for formatting an all day event	<p>[tag:calendar-display-monthView-event-allDay] - if the event is an All Day event, then it is identified by the resource bundle 'CalendarProvider-allDayEvent' value</p> <p>[tag:calendar-display-monthView-event-summary] - event summary</p>
display-monthView-task.template - Used for formatting a task	<p>[tag:calendar-display-monthView-task-pendHour2] - hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-monthView-task-pendMinute] - minutes of the task</p> <p>[tag:calendar-display-monthView-task-pendAmPm] - am or pm identifier</p> <p>[tag:calendar-display-monthView-task-summary] - task summary</p>

Table A-13 MExchangeCalendarProvider Template Files *(Continued)*

Template File and Description	Tag Name and Description
display-monthView-weekView.template - Used for formatting the week view with a month	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:calendar-display-monthView-day0] - day in month 0
	[tag:calendar-display-monthView-day1] - day in month 1
	[tag:calendar-display-monthView-day2] - day in month 2
	[tag:calendar-display-monthView-day3] - day in month 3
	[tag:calendar-display-monthView-day4] - day in month 4
	[tag:calendar-display-monthView-day5] - day in month 5
	[tag:calendar-display-monthView-day6] - day in month 6
	[tag:calendar-display-monthView-eventList0] - events for day in month 0
	[tag:calendar-display-monthView-taskList0] - tasks for day in month 0
	[tag:calendar-display-monthView-eventList1] - events for day in month 1
	[tag:calendar-display-monthView-taskList1] - tasks for day in month 1
	[tag:calendar-display-monthView-eventList2] - events for day in month 2
	[tag:calendar-display-monthView-taskList2] - tasks for day in month 2
	[tag:calendar-display-monthView-eventList3] - events for day in month 3
	[tag:calendar-display-monthView-taskList3] - tasks for day in month 3
	[tag:calendar-display-monthView-eventList4] - events for day in month 4
	[tag:calendar-display-monthView-taskList4] - tasks for day in month 4
	[tag:calendar-display-monthView-eventList5] - events for day in month 5
	[tag:calendar-display-monthView-taskList5] - tasks for day in month 5
	[tag:calendar-display-monthView-eventList6] - events for day in month 6
	[tag:calendar-display-monthView-taskList6] - tasks for day in month 6

Table A-13 MExchangeCalendarProvider Template Files *(Continued)*

Template File and Description	Tag Name and Description
display-monthView.template - Used for generating the entire month layout	[tag:calendar-display-monthView-currentDayOfWeek] - today's day of week
	[tag:calendar-display-monthView-currentMonth] - today's month
	[tag:calendar-display-monthView-currentDay] - today's day in month
	[tag:calendar-display-monthView-currentYear] - today's year
	[tag:calendar-display-monthView-dayOfWeek] - day of week content
	[tag:calendar-display-monthView-weekView0] - week content for week 0
	[tag:calendar-display-monthView-weekView1] - week content for week 1
	[tag:calendar-display-monthView-weekView2] - week content for week 2
	[tag:calendar-display-monthView-weekView3] - week content for week 3
	[tag:calendar-display-monthView-weekView4] - week content for week 4
display-summary-events.template - Used to show number of events	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:calendar-display-summary-events] - event summary information
display-summary-tasks.template - Used to show number of tasks	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:calendar-display-summary-tasks] - task summary information
display-summary.template - Used for formatting number of total and unread messages	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:calendar-display-summary-events] - event summary
	[tag:calendar-display-summary-tasks] - task summary

Table A-13 MExchangeCalendarProvider Template Files *(Continued)*

Template File and Description	Tag Name and Description
display-weekView-currentDayHeader.template - Used for formatting header for week view	<p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-weekView-currentDayOfWeek] - today's day of week</p> <p>[tag:calendar-display-weekView-currentMonth] - today's month</p> <p>[tag:calendar-display-weekView-currentDay] - today's day in month</p> <p>[tag:calendar-display-weekView-currentYear] - today's year</p>
display-weekView-emptyEventList.template - Used for formatting an empty event list	No tags.
display-weekView-emptyTaskList.template - Used for formatting an empty task list	No tags.
display-weekView-event.template - Used for formatting an event	<p>[tag:calendar-display-weekView-event-startHour1] - start hour for 0 based 12-hour clock</p> <p>[tag:calendar-display-weekView-event-endHour1] - end hour for 0 based 12-hour clock</p> <p>[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:calendar-display-event-conflict] - if the event is in conflict, then it is identified by the resource bundle 'conflict' value.</p> <p>[tag:calendar-display-weekView-event-startHour2] - start hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-weekView-event-startMinute] - minutes of the start time</p> <p>[tag:calendar-display-weekView-event-startAmPm] - am or pm identifier</p> <p>[tag:calendar-display-weekView-event-endHour2] - end hour based on user preference of 12 or 24 hour clock format</p> <p>[tag:calendar-display-weekView-event-endMinute] - minutes of the end time</p> <p>[tag:calendar-display-weekView-event-endAmPm] - am or pm identifier</p> <p>[tag:calendar-display-weekView-event-summary] - event summary</p>

Table A-13 MExchangeCalendarProvider Template Files *(Continued)*

Template File and Description	Tag Name and Description
display-weekView-eventAllDay.template - Used for formatting an all day event	[tag:calendar-display-weekView-event-allDay] - if the event is an All Day event, then it is identified by the resource bundle 'CalendarProvider-allDayEvent' value [tag:calendar-display-weekView-event-summary] - event summary
display-weekView-task.template - Used for formatting a task	[tag:calendar-display-weekView-task-pendHour2] - hour based on user preference of 12 or 24 hour clock format [tag:calendar-display-weekView-task-pendMinute] - minutes of the end time [tag:calendar-display-weekView-task-pendAmPm] - am or pm identifier [tag:calendar-display-weekView-task-summary] - task summary

Table A-13 MExchangeCalendarProvider Template Files *(Continued)*

Template File and Description	Tag Name and Description
display-weekView.template - Used for the overall week view	[tag:calendar-display-weekView-currentDayHeader] - today's date information
	[tag:calendar-display-weekView-dayOfWeek0] - day of week 0
	[tag:calendar-display-weekView-dayOfWeek1] - day of week 1
	[tag:calendar-display-weekView-dayOfWeek2] - day of week 2
	[tag:calendar-display-weekView-dayOfWeek3] - day of week 3
	[tag:calendar-display-weekView-dayOfWeek4] - day of week 4
	[tag:calendar-display-weekView-dayOfWeek5] - day of week 5
	[tag:calendar-display-weekView-dayOfWeek6] - day of week 6
	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:calendar-display-weekView-day0] - day in week 0
	[tag:calendar-display-weekView-day1] - day in week 1
	[tag:calendar-display-weekView-day2] - day in week 2
	[tag:calendar-display-weekView-day3] - day in week 3
	[tag:calendar-display-weekView-day4] - day in week 4
	[tag:calendar-display-weekView-day5] - day in week 5
	[tag:calendar-display-weekView-day6] - day in week 6
	[tag:calendar-display-weekView-eventList0] - events for day in week 0
	[tag:calendar-display-weekView-taskList0] - tasks for day in week 0
	[tag:calendar-display-weekView-eventList1] - events for day in week 1
	[tag:calendar-display-weekView-taskList1] - tasks for day in week 1
[tag:calendar-display-weekView-eventList2] - events for day in week 2	
[tag:calendar-display-weekView-taskList2] - tasks for day in week 2	

Table A-13 MExchangeCalendarProvider Template Files *(Continued)*

Template File and Description	Tag Name and Description
<p><code>display.template</code> - Used for overall channel formatting</p>	[tag:calendar-display-weekView-eventList3] - events for day in week 3
	[tag:calendar-display-weekView-taskList3] - tasks for day in week 3
	[tag:calendar-display-weekView-eventList4] - events for day in week 4
	[tag:calendar-display-weekView-taskList4] - tasks for day in week 4
	[tag:calendar-display-weekView-eventList5] - events for day in week 5
	[tag:calendar-display-weekView-taskList5] - tasks for day in week 5
	[tag:calendar-display-weekView-eventList6] - events for day in week 6
	[tag:calendar-display-weekView-taskList6] - tasks for day in week 6
	[tag:iwtDesktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:ErrorMessage] - error content
	[tag:calendar-display-dayView-summary] - summary content
	[tag:calendar-display-dayView] - day content
	[tag:calendar-display-weekView] - week content
[tag:calendar-display-monthView] - month content	
[tag:calendar-display-clientURL] - client application url	
<p><code>edit-checkbox.template</code> - Used for creating edit page checkboxes</p>	[tag:fontFace] - Font chosen
	[tag:fontColor] - Color of font chosen
	[tag:resourceName] - resource name
	[tag:header] - display value
	[tag:attName] - element name
[tag:attSelected] - if the element is selected, then marks it as selected	
<p><code>edit-end.template</code> - Used for creating end of the edit page</p>	No tags.

Table A-13 MExchangeCalendarProvider Template Files *(Continued)*

Template File and Description	Tag Name and Description
edit-link.template - Used for creating application helper edit link	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:editLink] - URL to edit
edit-password.template - Used for creating edit page password boxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName] - resource name [tag:header] - display value [tag:attName] - element name [tag:attValue] - element value
edit-select.template - Used for creating edit page select boxes	[tag:fontFace1] - Default font for the Desktop (sans-serif) [tag:fontColor] - Color of font chosen [tag:resourceName] - resource name [tag:header] - display value [tag:attName] - element name [tag:options] - element options [tag:attValue] - element value
edit-selectoption.template - Used for creating edit page select box options	[tag:selected] - if the element is selected, then marks it as selected [tag:string] - display value
edit-start.template - Used for creating the start of the edit page	[tag:process], [tag:isAppHandler], [tag:mail-display-error] - Used to dynamically build edit pages. Should not be edited. [tag:borderColor] - Color of the channel border. The border color can be changed by user in the desktop custom theme page. [tag:bgColor] - Provider background color chosen [tag:titlebarColor] - Color of title bar [tag:fontFace] - Font chosen

Table A-13 MExchangeCalendarProvider Template Files *(Continued)*

Template File and Description	Tag Name and Description
edit-string.template - Used for creating edit page text boxes	[tag:fontFace] - Font chosen
	[tag:fontColor] - Color of font chosen
	[tag:header] - display value
	[tag:attName] - element name
	[tag:attValue] - element value
url.template - Used for creating hyperlinks	[tag:resourceName] - resource name
	[tag:link] - URL location
	[tag:name] - display value

MExchangeMailProvider

[Table A-14](#) lists the templates in the `html` subdirectory of the `MExchangeMailProvider` and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-14 MExchangeMailProvider Template Files

Template and Description	Tag Name and Description
display-clientURL.template - Used for displaying the Launch Mail link	[tag:mail-display-clientURL-uri] - Used to display application launch link
display-error.template - Used for displaying error messages	[tag:fontFace] - Font chosen
	[tag:mail-display-error] - Display error message
display-headers-message.template - Used for formatting individual message information	[tag:fontFace] - Font chosen
	[tag:bgColor] - Provider background color chosen
	[tag:fontColor] - Color of font chosen
	[tag:mail-display-headers-message-subject] - mail subject
	[tag:mail-display-headers-message-mailto] - Mail senders
	[tag:mail-display-headers-message-name] - Mail sender's name
	[tag:mail-display-headers-message-date] - Date message received
[tag:mail-display-headers-message-status] - Message status	

Table A-14 MExchangeMailProvider Template Files *(Continued)*

Template and Description	Tag Name and Description
display-headers.template - Used for formatting message table headings	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:mail-display-headers-subject] - Displays subject [tag:mail-display-headers-from] - Displays From [tag:mail-display-headers-date] - Displays Date [tag:mail-display-headers-status] - Displays status
display-summary.template - Used for formatting number of total and unread messages	[tag:fontFace] - Font chosen [tag:mail-display-summary-unread] - displays number of unread messages [tag:mail-display-summary-total] - displays total number of messages
display.template - Used for overall channel formatting	[tag:mail-display-summary] - Displays the summary info [tag:mail-display-headers] - displays the messages headers [tag:mail-display-clientURL] - displays the application launch URL
edit-checkbox.template - Used for creating edit page checkboxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:attSelected] - Used to dynamically build edit pages. Should not be edited.
edit-end.template - Used for creating end of the edit page	No tags
edit-link.template - Used for creating application helper edit link	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:editLink] - Used to display link for the application helper editing
edit-password.template - Used for creating edit page password boxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:attValue] - Used to display link for the application helper editing

Table A-14 MExchangeMailProvider Template Files *(Continued)*

Template and Description	Tag Name and Description
edit-select.template - Used for creating edit page select boxes	[tag:fontFace1] - Default font for the Desktop (sans-serif) [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:options] - Used to display link for the application helper editing
edit-selectoption.template - Used for creating edit page select box options	[tag:attValue], [tag:selected], [tag:string] - Used to dynamically build edit pages. Should not be edited.
edit-start.template - Used for creating the start of the edit page	[tag:process], [tag:isAppHandler] - Used to dynamically build edit pages. Should not be edited. [tag:mail-display-error] - Display error message [tag:bgColor] - Provider background color chosen [tag:titlebarColor] - Color of title bar [tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen
edit-string.template - Used for creating edit page text boxes	[tag:fontFace] - Font chosen [tag:fontColor] - Color of font chosen [tag:resourceName], [tag:header], [tag:attName], [tag:attValue] - Used to dynamically build edit pages. Should not be edited.
url.template - Used for creating hyperlinks	[tag:link, [tag:name] - Used to dynamically build edit pages. Should not be edited.

TemplateTabContainerProvider

Table A-15 lists the templates in the `TemplateTabContainerProvider` subdirectory and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-15 TemplateTabContainerProvider Template Subdirectory

TemplateTableContainer Template and Description	Tag Name and Description
<code>banner.template</code> - The banner across the top of the Desktop pages, including the Edit, Layout, and Content pages.	<p>[<code>surl:/images/spacer.gif</code>] - Used to space gifs or channels</p> <p>[<code>surl:/images/productName.gif</code>] - Logo gif of the product name</p> <p>[<code>tag:productName</code>] - The product name</p> <p>[<code>surl:/images/blueBullet.gif</code>] - Blue button to denote Home, Help, Logout, and so on</p> <p>[<code>tag:frontContainerName</code>] - Name of the front container</p> <p>[<code>tag:tabContainerName</code>] - Name of the tab container</p> <p>[<code>tag:help_link</code>] - Help link for Desktop</p> <p>[<code>surl:/desktop/images/nothing.gif</code>] - Used to space gifs or channels</p>
<code>display.template</code>	<p>[<code>tag:productName</code>] - Product name</p> <p>[<code>surl:/desktop/css/style.css</code>] - Inserts style template</p> <p>[<code>tag:bulletColor</code>] - Inserts <code>bulletColor.js</code> into this template</p> <p>[<code>tag:banner</code>]- Inserts <code>banner.template</code> into this template</p> <p>[<code>tag:inlineError</code>] - Replaced with <code>inlinError.template</code> if error has occurred</p> <p>[<code>tag:content</code>] - Content of the channel</p> <p>[<code>tag:menubar</code>] - Inserts <code>menubar.template</code> into this template</p>
<code>editForm.template</code> - Edit page template for creating and removing tabs.	<p>[<code>tag:productName</code>] - Product name</p> <p>[<code>surl:/desktop/css/style.css</code>] - Style sheet used by the Desktop for the banner and tabs templates</p> <p>[<code>tag:bulletColor</code>] - Inserts <code>bulletColor.js</code> into this template</p> <p>[<code>tag:banner</code>] - Inserts <code>banner.template</code> into this template</p> <p>[<code>tag:inlineError</code>] - Replaced with <code>inlinError.template</code> if error has occurred</p> <p>[<code>tag:TemplateTabContainer.editPage</code>] - Edit tabs page</p> <p>[<code>tag:fontFace1</code>] - Default font for the Desktop (sans-serif)</p> <p>[<code>tag:desktop_url</code>] - URL of Desktop to return to</p> <p>[<code>tag:menubar</code>] - Inserts <code>menubar.template</code> into this template</p>

Table A-15 TemplateTabContainerProvider Template Subdirectory *(Continued)*

TemplateTableContainer Template and Description	Tag Name and Description
<code>inlineError.template</code> - HTML to show an error message.	[tag:errMessage] - The error message
<code>makeNewTab.template</code> - HTML used in <code>editForm.template</code> for creating a new tab.	[tag:fontFace1] - Default font for the Desktop (sans-serif) [tag:TemplateTabContainer.tabTopics] - HTML for selecting a tab topic
<code>menubar.template</code> - HTML for the menubar across the bottom of the Desktop pages, including the Edit, Layout, and Content pages, and containing Home, Help and Log Out links.	[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif) [tag:productName] - Product name [tag:tabContainerName] - Name of the tab container [tag:frontContainerName] - Name of the front container [tag:help_link] - Link to help from Desktop
<code>noCache.template</code> - HTML META headers to prevent browser caching of the pages.	(No Tags)
<code>removeRenameTab.template</code> - HTML used in <code>editForm.template</code> for removing and renaming existing tab(s).	[tag:fontFace1] - Default font for the Desktop (sans-serif) [tag:TemplateTabContainer.tabList] - List of available tabs to remove.
<code>selectedTab.template</code> - HTML used to show the currently selected tab on the Desktop.	[surl:/desktop/images/nothing.gif]- Used to space gifs or channels [surl:/desktop/tabs/images/tabNotch.gif]- Used to place notch in tab [tag:desktop-fontFace1] - Default font for the Desktop (sans-serif) [tag:TemplateTabContainer.tabName] - Tab name
<code>tab.template</code> - HTML used to show the unselected tab(s) on the Desktop.	[tag:TemplateTabContainer.tabURL] -Tab URL (When Tab is clicked on) [surl:/desktop/images/nothing.gif]- Used to space gifs or channels [surl:/desktop/tabs/images/tabNotch.gif]- Used to place notch in tab [tag:desktop-fontFace1] - Default font for the Desktop (sans-serif) [tag:TemplateTabContainer.tabName] - Tab name

Table A-15 TemplateTabContainerProvider Template Subdirectory (*Continued*)

TemplateTableContainer Template and Description	Tag Name and Description
<code>tabs.template</code> - Template of the tab provider on the Desktop with tabs on the left.	[<code>url:/desktop/images/nothing.gif</code>] - Used to space gifs or channels [<code>tag:tabs</code>] - Inserts either <code>tab.template</code> or <code>selectedTab.template</code> depending on whether the current tab being drawn is selected or not
<code>tabs_r.template</code> - Template of the tab provider on the Desktop with tabs on the right.	[<code>tag:tabs</code>] - Inserts either <code>tab.template</code> or <code>selectedTab.template</code> depending on whether the current tab being drawn is selected or not

TemplateTableContainerProvider

Table A-16 lists the templates in the `TemplateTableContainerProvider` subdirectory and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-16 TemplateTableContainerProvider Template Subdirectory

TemplateTableContainer Template and Description	Tag Name and Description
<code>arrangeProvider.js</code> - JavaScript code used in the Desktop Layout page.	(No Tags)
<code>banner.template</code> - The banner across the top of the Desktop pages, including the Edit, Layout, and Content pages.	[<code>url:/images/spacer.gif</code>] - Used to space gifs or channels [<code>url:/images/productName.gif</code>] - Logo gif of the product name [<code>url:/images/blueBullet.gif</code>] - Blue button to denote Home, Help, Logout, and so on [<code>tag:frontContainerName</code>] - Name of the front container [<code>tag:help_link</code>] - Help link for Desktop [<code>url:/desktop/images/nothing.gif</code>] - Used to space gifs or channels

Table A-16 TemplateTableContainerProvider Template Subdirectory *(Continued)*

TemplateTableContainer Template and Description	Tag Name and Description
bareProviderWrapper.template - Template for each provider wrapper with no titlebar.	<p>[tag:borderWidth] - Size of border (CELLPADDING) around provider HTML table</p> <p>[tag:size] - Table width (WIDTH), fixed at 100%</p> <p>[tag:borderColor] - Color of the channel border. The border color can be changed by user in the desktop custom theme page.</p> <p>[tag:bgColor] - Provider background color chosen</p> <p>[tag:fontColor] - Color of font chosen</p> <p>[tag:fontFace] - Font chosen</p> <p>[tag:content] - Content of the channel</p> <p>[surl:/desktop/images/nothing.gif] - Used to space gifs or channels</p>
contentBarInContent.template - Content bar for template displayed when a user selects Content on any other page's Content bar.	<p>[tag:toolbarRollover] - Inserts toolbarRollover.js into this template.</p> <p>[surl:/desktop/images/nothing.gif]- Used to space gifs or channels</p> <p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p>
contentBarInLayout.template - Content bar for template displayed when user selects Layout on any other page's Content bar.	<p>[tag:toolbarRollover] - Inserts toolbarRollover.js into this template.</p> <p>[surl:/desktop/images/nothing.gif]- Used to space gifs or channels</p> <p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p>
contentLayout.template - Content bar for template displayed on the Desktop before the user selects Content or Layout.	<p>[tag:toolbarRollover] - Inserts toolbarRollover.js into this template.</p> <p>[surl:/desktop/images/nothing.gif]- Used to space gifs or channels</p> <p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:selectedName] - Name of selected tab</p> <p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p>

Table A-16 TemplateTableContainerProvider Template Subdirectory (*Continued*)

TemplateTableContainer Template and Description	Tag Name and Description
<p><code>contentTemplate.template</code> - The HTML template for the Content (Channels) page that displays when a user selects Content .</p>	<p>[tag:productName] - Product name</p> <p>[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates</p> <p>[tag:banner] - Inserts <code>banner.template</code> into this template</p> <p>[tag:contentBarInContent] - Inserts <code>contentBarInContent.template</code> into this template</p> <p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:thinProviders] - Checkbox list of available thin providers to add to desktop</p> <p>[tag:wideProviders] - Checkbox list of available wide providers to add to desktop</p> <p>[tag:fullProviders] - Checkbox list of available full width providers to add to desktop</p> <p>[tag:desktop_url] - URL of Desktop to return to</p> <p>[tag:menubar] - Inserts <code>menubar.template</code> into template</p>
<p><code>launchPopup.js</code> - JavaScript code to launch a popup window.</p>	<p>(No Tags)</p>

Table A-16 TemplateTableContainerProvider Template Subdirectory *(Continued)*

TemplateTableContainer Template and Description	Tag Name and Description
layout1Template.template - left/thin, right/wide.	<p>[tag:productName] - Product name</p> <p>[tag:arrangeProvider] - Inserts arrangeProvider.js template into this template</p> <p>[tag:performSubstitution] - Inserts performSubstitution.js template</p> <p>[tag:performColumnSubstitution] - Inserts performColumnSubstitution.js template</p> <p>[tag:selectAll] - Inserts selectAll.js template</p> <p>[tag:switchColumns] - Inserts switchColumns.js template</p> <p>[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates</p> <p>[tag:bulletColor] - Inserts bulletColor.js into this template</p> <p>[tag:banner] - Inserts banner.template into this template</p> <p>[tag:contentBarInLayout] - Inserts contentBarInLayout.template into this template</p> <p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:layoutFullTop] - Inserts layoutFullTop.template into this template</p> <p>[tag:leftUserProviderList] - List that shows left column channels</p> <p>[surl:/desktop/images/b_up.gif] - Icon that shows up arrow</p> <p>[surl:/desktop/images/b_down.gif] - Icon that shows down arrow</p> <p>[tag:rightUserProviderList] - List that shows right column channels</p> <p>[tag:layoutFullBottom] - Inserts layoutFullBottom.template into this template</p> <p>[tag:desktop_url] - URL of Desktop to return to</p> <p>[tag:menubar] - Inserts menubar.template into this template</p>

Table A-16 TemplateTableContainerProvider Template Subdirectory (*Continued*)

TemplateTableContainer Template and Description	Tag Name and Description
layout2Template.template - left/wide, right/thin.	<p>[tag:productName] - Product name</p> <p>[tag:arrangeProvider] - Inserts arrangeProvider.js template</p> <p>[tag:performSubstitution] - Inserts performSubstitution.js template</p> <p>[tag:performColumnSubstitution] - Inserts performColumnSubstitution.js template</p> <p>[tag:selectAll] - Inserts selectAll.js template</p> <p>[tag:switchColumns] - Inserts switchColumns.js template</p> <p>[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates</p> <p>[tag:bulletColor] - Inserts bulletColor.js into this template</p> <p>[tag:banner] - Inserts banner.template into this template</p> <p>[tag:contentBarInLayout] - Inserts contentBarInLayout.template into this template</p> <p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:layoutFullTop] - Inserts layoutFullTop.template into this template</p> <p>[tag:leftUserProviderList] - List that shows left column channels</p> <p>[surl:/desktop/images/b_up.gif] - Icon that shows up arrow</p> <p>[surl:/desktop/images/b_down.gif] - Icon that shows down arrow</p> <p>[tag:rightUserProviderList] - List that shows right column channels</p> <p>[tag:layoutFullBottom] - Inserts layoutFullBottom.template template</p> <p>[tag:desktop_url] - URL of Desktop to return to</p> <p>[tag:menubar] - Inserts menubar.template into this template</p>

Table A-16 TemplateTableContainerProvider Template Subdirectory (*Continued*)

TemplateTableContainer Template and Description	Tag Name and Description
layout3Template.template - left/thin, center/wide, right/thin.	<p>[tag:productName] - Product name</p> <p>[tag:arrangeProvider] - Inserts arrangeProvider.js template</p> <p>[tag:performSubstitution] - Inserts performSubstitution.js template</p> <p>[tag:performColumnSubstitution] - Inserts performColumnSubstitution.js template</p> <p>[tag:selectAll] - Inserts selectAll.js template</p> <p>[tag:switchColumns] - Inserts switchColumns.js template</p> <p>[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates</p> <p>[tag:bulletColor] - Inserts bulletColor.js into this template</p> <p>[tag:banner] - Inserts banner.template into this template</p> <p>[tag:contentBarInLayout] - Inserts contentBarInLayout.template into this template</p> <p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:layoutFullTop] - Inserts layoutFullTop.template template</p> <p>[tag:leftUserProviderList] - List that shows left column channels</p> <p>[surl:/desktop/images/b_up.gif] - Icon that shows up arrow</p> <p>[surl:/desktop/images/b_left.gif] - Icon that shows arrow pointing left</p>

Table A-16 TemplateTableContainerProvider Template Subdirectory *(Continued)*

TemplateTableContainer Template and Description	Tag Name and Description
	[surl:/desktop/images/b_down.gif] - Icon that shows down arrow
	[surl:/desktop/images/b_right.gif] - Icon that shows arrow pointing right
	[tag:centerUserProviderList] - List that shows center column channels
	[tag:rightUserProviderList] - List that shows right column channels
	[tag:layoutFullBottom] - Inserts layoutFullBottom.template template
	[tag:desktop_url] - URL of Desktop to return to
	[tag:menubar] - Inserts menubar.template into template

Table A-16 TemplateTableContainerProvider Template Subdirectory (*Continued*)

TemplateTableContainer Template and Description	Tag Name and Description
layout4Template.template - left/thin, center/thin, right/thin.	<p>[tag:productName] - Product name</p> <p>[tag:arrangeProvider] - Inserts arrangeProvider.js template</p> <p>[tag:performSubstitution] - Inserts performSubstitution.js template</p> <p>[tag:performColumnSubstitution] - Inserts performColumnSubstitution.js template</p> <p>[tag:selectAll] - Inserts selectAll.js template</p> <p>[tag:switchColumns] - Inserts switchColumns.js template</p> <p>[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates</p> <p>[tag:bulletColor] - Inserts bulletColor.js into this template</p> <p>[tag:banner] - Inserts banner.template template</p> <p>[tag:contentBarInLayout] - Inserts contentBarInLayout.template into this template</p> <p>[tag:frontContainerName] - Name of container</p> <p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:layoutFullTop] - Inserts layoutFullTop.template template</p> <p>[tag:leftUserProviderList] - List that shows left column channels</p> <p>[surl:/desktop/images/b_up.gif] - Icon that shows up arrow</p> <p>[surl:/desktop/images/b_down.gif] - Icon that shows down arrow</p>

Table A-16 TemplateTableContainerProvider Template Subdirectory *(Continued)*

TemplateTableContainer Template and Description	Tag Name and Description
	[surl:/desktop/images/b_right.gif] - Icon that shows arrow pointing right
	[tag:centerUserProviderList] - List that shows center column channels
	[surl:/desktop/images/b_left.gif] - Icon that shows arrow pointing left
	[tag:rightUserProviderList] - List that shows right column channels
	[tag:layoutFullBottom] - Inserts layoutFullBottom.template template
	[tag:desktop_url] - URL of Desktop to return to
	[tag:menubar] - Inserts menubar.template template
layoutFullBottom.template - Partial HTML template for Layout pages when the user has a full width channel available at the bottom of the layout.	[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:fullbottomUserProviderList] - List of Full Width (Bottom) Channels to move
	[surl:/desktop/images/b_up.gif] - Icon that shows up arrow
	[surl:/desktop/images/b_down.gif] - Icon that shows down arrow
layoutFullTop.template - Partial HTML template for Layout pages when the user has a full width channel available at the top of the layout	[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:fulltopUserProviderList] - List of Full Width (Top) Channels to move
	[surl:/desktop/images/b_up.gif] - Icon that shows up arrow
	[surl:/desktop/images/b_down.gif] - Icon that shows down arrow

Table A-16 TemplateTableContainerProvider Template Subdirectory *(Continued)*

TemplateTableContainer Template and Description	Tag Name and Description
maximizedTemplate.template - Template of a provider when it's in its maximized state.	<p>[tag:launchPopup] - Inserts launchPoup.js template</p> <p>[tag:detachedContent] - Channel content in the detached window</p> <p>[tag:productName] - Product Name</p> <p>[tag:bulletColor] - Inserts bulletColor.js into this template</p> <p>[tag:banner] - Inserts banner.template template</p> <p>[tag:contentLayout] - Inserts contentLayout.template in this template</p> <p>[tag:inlineError] - Replaced with inlineError.template if error has occurred</p> <p>[tag:MaximizedContent] - Inserts content in the maximize mode</p> <p>[tag:menubar] - Inserts the menuber.template</p>
menubar.template - HTML for the menubar across the bottom of the Desktop pages, including the Edit, Layout, and Content pages and containing Home, Help and Log Out links.	<p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:productName] - Product name</p> <p>[tag:frontContainerName] - Name of container</p> <p>[tag:help_link] - Link to help from Desktop</p>
minimized.template - Template of a provider when it's in its minimized state - just the title/button bar showing, no content area showing.	<p>[tag:borderWidth] - Size of border (CELLPADDING) around channel (default 0)</p> <p>[tag:size] - Table width (WIDTH), fixed at 100%</p> <p>[tag:borderColor] - Color of the channel border. The border color can be changed by user in the desktop custom theme page.</p> <p>[tag:titleBarColor] - Color of title bar</p> <p>[tag:fontFace] - Default font for the Desktop (sans-serif)</p> <p>[tag:title] - The title of the provider/channel</p> <p>[tag:titleBarColor] - Title bar color</p> <p>[tag:provider_cmds] - Inserts the channel command button links (maximize, detach, and so on)</p> <p>[surl:/desktop/images/nothing.gif]- Used to space gifs or channels</p>

Table A-16 TemplateTableContainerProvider Template Subdirectory (*Continued*)

TemplateTableContainer Template and Description	Tag Name and Description
optionsTemplate.template - Template for the Options page of the Desktop.	<p>[tag:productName] - Product name</p> <p>[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates</p> <p>[tag:banner] - Inserts banner.template template</p> <p>[tag:frontContainerName] - Name of container</p> <p>[tag:inlineError] - Replaced with inlineError.template template if error has occurred.</p> <p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:serviceTimeout] - Provider Timeout in seconds</p> <p>[surl:/desktop/images/layout1.gif] - Icon for thin-wide layout</p> <p>[tag:layoutOneChecked] - Makes Radio button for layout one the default.</p> <p>[surl:/desktop/images/layout2.gif] - Icon for wide-thin layout</p> <p>[tag:layoutTwoChecked] - Makes Radio button for layout two the default.</p> <p>[surl:/desktop/images/layout3.gif] - Icon for thin-wide-thin layout</p> <p>[tag:layoutThreeChecked] - Makes Radio button for layout three the default.</p> <p>[surl:/desktop/images/layout4.gif] - Icon for thin-thin-thin layout</p> <p>[tag:layoutFourChecked] - Makes Radio button for layout four the default.</p> <p>[tag:desktop_url] - URL of Desktop to return to</p> <p>[tag:menubar] - Inserts menubar.template template</p>
performColumnSubstitution.js - JavaScript code used on the Layout page.	(No Tags)
performSubstitution.js - JavaScript code used on the Layout page.	(No Tags)
popupMenubar.template - Menubar to be used in popup windows.	<p>[surl:/images/spacer.gif]- Used to space gifs or channels</p> <p>[surl:/images/productName.gif]- Logo gif of Product name</p> <p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:frontContainerName] - Name of container</p> <p>[tag:providerName] - Provider name</p>

Table A-16 TemplateTableContainerProvider Template Subdirectory (Continued)

TemplateTableContainer Template and Description	Tag Name and Description
popupTemplate.template - Used to show provider/channel content in detached windows. Similar use to providerWrapper.template, but not in a table structure.	<p>[tag:productName] - Product name</p> <p>[tag:providerTitle] - Provider/Channel title</p> <p>[tag:openURLInParent] - Inserts openURLInParent.js template</p> <p>[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates</p> <p>[tag:popupMenubar] - Inserts popupMenubar.template template</p> <p>[tag:providerContent] - Content of the provider</p>
providerWrapper.template - Template that all providers and channels use for layout on Desktop. Defines the look of the border of the providers and channels on the screen.	<p>[tag:borderWidth] - size of border (CELLPADDING) around provider HTML table</p> <p>[tag:size] - Table width, fixed at 100%</p> <p>[tag:borderColor] - Color of the channel border. The border color can be changed by user in the desktop custom theme page.</p> <p>[tag:titleBarColor] - Color of title bar</p> <p>[tag:fontFace] - Chosen font for the channels</p> <p>[tag:title] - The title of the provider/channel</p> <p>[tag:provider_cmds] - Inserts the channel command button links (minimize, detach, and so on)</p> <p>[tag:bgColor] - Background color</p> <p>[tag:fontColor] - Color of chosen font for title</p> <p>[tag:content] - Content of the channel</p> <p>[surl:/desktop/images/nothing.gif]- Used to space gifs or channels</p>
removeProvider.js - JavaScript code used to remove a channel from the Desktop.	(No Tags)
selectAll.js - JavaScript code used on the Layout page.	(No Tags)
switchColumns.js - JavaScript code used on the Layout page.	(No Tags)

Table A-16 TemplateTableContainerProvider Template Subdirectory (*Continued*)

TemplateTableContainer Template and Description	Tag Name and Description
<p><code>userTemplate.template</code> - The base Desktop layout structure document. Very little in this file, as most of the content of the Desktop is swapped in during processing in the servlets.</p>	<p>[tag:launchPopup] - Inserts <code>launchPopup.js</code> template</p> <p>[tag:detachedContent] - Provider content</p> <p>[tag:productName] - Product name</p> <p>[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates</p> <p>[tag:bulletColor] - Inserts <code>bulletColor.js</code> into this template</p> <p>[tag:banner] - Inserts <code>banner.template</code> template</p> <p>[tag:contentLayout] - Inserts <code>contentLayout.template</code> in this template</p> <p>[tag:inlineError] - Replaced with <code>inlineError.template</code> if error has occurred</p> <p>[tag:fullTopContent] - Provider/channel content (HTML) inserted here</p> <p>[tag:leftWidth] - Percentage of total width for left column</p> <p>[tag:leftContent] - Channels for left column</p> <p>[tag:centerWidth] - Percentage of total width for center column</p> <p>[tag:centerContent] - Channels for center column</p> <p>[tag:rightWidth] - Percentage of total width for right column</p> <p>[tag:rightContent] - Channels for right column</p> <p>[tag:fullBottomContent] - The content generated by the full-bottom channel</p> <p>[tag:menubar] - Inserts <code>menubar.template</code> template</p>

UserInfoProvider

[Table A-17](#) lists the templates in the `html` subdirectory of `UserInfoProvider` and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-17 UserInfo Template Subdirectory

UserInfo Template and Description	Tag Name and Description
<code>content.template</code> - Content page for the UserInfo provider.	<p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:greeting] - User's greeting</p> <p>[tag:cn] - User's common or full name</p> <p>[tag:currentDate] - Date</p> <p>[tag:timeLeft] - Time left in user's session</p> <p>[tag:maxIdle] - Maximum idle time</p>
<code>edit.template</code> - Edit page for the UserInfo provider.	<p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:cn] - User's common or full name</p> <p>[tag:greeting] - User's greeting</p> <p>[tag:timezoneList] - List of time zones</p> <p>[tag:localeList] - List of available locales</p> <p>[tag:netmailSettings] - Inserts <code>netmailSettings.template</code> into this page to get the following information: IMAP server name, SMTP server name, IMAP user ID and IMAP password</p> <p>[tag:passwordHandler] - Inserts <code>passwordHandler-Membership.template</code> (if available) to change membership password</p>
<code>netmailSettings.template</code> - Partial template inserted into <code>edit.template</code> for Mail information.	<p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:iplanet-ps-netmail-imap-server-name] - IMAP server name</p> <p>[tag:iplanet-ps-netmail-smtp-server-name] - SMTP server name</p> <p>[tag:iplanet-ps-netmail-imap-userid] - IMAP user ID</p> <p>[tag:iplanet-ps-netmail-imap-password] - IMAP password</p>
<code>passwordHandler-Membership.template</code> - Partial template inserted into <code>edit.template</code> if Membership Authentication is used, to allow the user to change their password.	<p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p>

default

Table A-18 lists the templates in the `default` directory and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-18 Template Files in the default Directory

File Name and Description	Tag Name and Description
<code>AttachCommand.template</code> - Handles reattaching a detached channel.	<p>[tag:detachAttachURL] - URL of the channel to be detached</p> <p>?action=process&provider=<i>thecontainername</i>&<i>thecontainername</i>.channelAction=detach&<i>thecontainername</i>.targetProvider=<i>providername</i></p> <p>[surl:/desktop/images/b_attach.gif] - A URL that points to the attach image</p> <p>[tag:detachAttachText] - Alternated string for the detach/attach image in the title bar, this is a localized string. The detach or attach mode is determined at run time.</p> <p>[tag:providerName] - Provider name</p>
<code>MaximizeCommand.template</code> - Allows the channel to be displayed in the "maximize" mode so that the channel occupies the entire Desktop.	<p>[tag:maximizeURL] - A URL to show the channel in maximize mode</p> <p>[surl:/desktop/images/b_maximize.gif] - A URL that points to the maximize image</p> <p>[tag:maximizeText] - A text that is used as an alternate for the maximize icon</p> <p>[tag:providerName] - The channel name</p>
<code>banner.template</code> - The banner across the top of the Desktop pages	<p>[surl:/images/spacer.gif] - Used to space gifs or channels</p> <p>[surl:/images/productName.gif] - Logo gif of the Product name</p> <p>[tag:productName] - The product name</p> <p>[surl:/images/blueBullet.gif] - Blue button to denote Home, Help, Logout, and so on</p> <p>[tag:frontContainerName] - Name of container</p> <p>[tag:help_link] - Help link for Desktop</p> <p>[surl:/desktop/images/nothing.gif] - Used to space gifs or channels</p>
<code>bulletColor.js</code> - JavaScript code used to select and display bullet color.	<p>[surl:/images/blueBullet.gif] - Blue button to denote home, help, logout, and so forth.</p> <p>[surl:/images/redBullet.gif] - Red button to denote home, help, logout, or whatever has been chosen.</p>

Table A-18 Template Files in the default Directory (*Continued*)

File Name and Description	Tag Name and Description
<p>contentLayout.template - Content bar for template displayed on the Desktop before the user selects Content or Layout.</p>	<p>[tag:toolbarRollover] - Inserts toolbarRollover.js into this template.</p> <p>[url:/desktop/images/nothing.gif]- Used to space gifs or channels</p> <p>[tag:frontContainerName] - Name of container</p> <p>[selectedName] - Name of selected tab</p> <p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p>
<p>detachCommand.template - Handles detaching a channel.</p>	<p>[tag:detachAttachURL] - URL of the channel to be detached</p> <p>action=process&provider=<i>thecontainername</i>&<i>thecontainername</i>.channelAction=attach&<i>thecontainername</i>.targetProvider=<i>providename</i></p> <p>[tag:detachAttachText] - Text for the alt tag for the detach icon</p> <p>[tag:providerName] - Provider name</p>
<p>detachEditCommand.template - Handles link to the Edit page for the detached channel.</p>	<p>[tag:editURL] - URL of Edit page for this channel</p> <p>[tag:editTag] - Text for the alt tag for the edit icon</p> <p>[tag:providerName] - Provider name</p>
<p>detachRemoveCommand.template - Handles closing or removing a detached channel.</p>	<p>[tag:removeURL] - URL of the detached channel to be removed</p> <p>?action=process&provider=<i>thecontainernames</i>&<i>thecontainername</i>.channelAction=remove&<i>thecontainername</i>.targetProvider=<i>providename</i></p> <p>[tag:removeText] - Text for the alt tag for the remove icon</p> <p>[tag:providerName] - Provider name</p>
<p>editCommand.template - Handles link to the Edit page for this channel.</p>	<p>[tag:editURL] - URL of Edit page for this channel</p> <p>?action=edit&provider=<i>theeditcontainername</i>&targetprovider=<i>providename</i>&containerName=<i>thecontainername</i></p> <p>[tag:editTag] - Text for the alt tag for the edit icon</p> <p>[tag:providerName] - Provider name</p>
<p>helpHref.template - Generates the help URL for each of the channels. Displays the help contents in a new window.</p>	<p>[tag:help_link] - Help link for Desktop</p> <p>[tag:help_icon] - Icon for help in channel title bar</p> <p>[tag:help_tag] - Text for alt message</p> <p>[tag:providerName] - Provider name</p>

Table A-18 Template Files in the default Directory (Continued)

File Name and Description	Tag Name and Description
<code>inlineError.template</code>	[tag:errMessage] - The error message
<code>menubar.template</code> - HTML for the menubar across the bottom of the Desktop pages.	[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif) [tag:productName] - Product name [tag:frontContainerName] - Name of container [tag:help_link] - Help link
<code>minMaximizeCommand.template</code> - Handles minimizing and maximizing a channel.	[tag:minMaximizeURL] - URL of channel to minimize or maximize <i>?action=process&provider=thecontainername&thecontainername.channelAction=maximize&thecontainername.targetProvider=providername</i> <i>?action=process&provider=thecontainername&thecontainername.channelAction=minimize&thecontainername.targetProvider=providername</i> [tag:minMaximizeIcon] - Inserts minimize /maximize icon [tag:minMaximizeText] - Alternated string for the minimize or maximize image in the title bar, this is a localized string, the state of minimize or maximize is determined at run time. [tag:providerName] - Provider Name
<code>minimizeCommand.template</code> - Allows the channel to be displayed in the minimize mode so that only the title bar of the channel is displayed and no content of the channel is displayed.	[tag:minMaximizeURL] - A URL to show the channel in either the minimize or the normal mode, the mode is decided at run time [surl:/desktop/images/b_minimize.gif] - A URL that points to the maximize image [tag:maximizeText] - A text that is used as an alternate for the maximize icon [tag:providerName] - The channel name
<code>noCache.template</code> - HTML META headers to prevent browser caching of the pages.	(No Tags)

Table A-18 Template Files in the default Directory (Continued)

File Name and Description	Tag Name and Description
<code>normalizeCommand.template</code> - Allows the channel to be displayed in the normal mode so that the channel is displayed in the Desktop, with all other channels in the same table container.	<p>[tag:minMaximizeURL] - A URL to show the channel in either the minimize or the normal mode, the mode is decided at run time</p> <p>[surl:/desktop/images/b_normal.gif] - A URL that points to the normal image</p> <p>[tag:maximizeText] - A text that is used as an alternate for the maximize icon</p> <p>[tag:providerName] - The channel name</p>
<code>openURLInParent.js</code> - Javascript to open a URL in the parent window. Used in popup windows.	(No Tags)
<code>providerCommands.template</code> - Commands available in title bar.	<p>[tag:minMaximizeCommand] - Inserts the <code>minMaximizeCommand.template</code></p> <p>[tag:helpCommand] - Inserts <code>helpHref.template</code>.</p> <p>[tag:editCommand]- Inserts the <code>editCommand.template</code></p> <p>[tag:detachAttachCommand] - If the channel is detached, insert the <code>detachCommand.template</code>; if the channel is attached, then insert the <code>attachCommand.template</code>.</p> <p>[tag:removeCommand] - Inserts the <code>removeCommand.template</code></p>
<code>redirect.template</code>	[tag:url] - URL of new location
<code>removeCommand.template</code> - Removes the channel.	<p>[tag:removeURL] - URL of the channel to be removed</p> <p>?action=process&provider=<i>thecontainername&thecontainername</i>.channelAction=remove&<i>thecontainername</i>.targetProvider=<i>providername</i></p> <p>[tag:providerName] - Provider Name</p>
<code>toolbarRollovers.js</code> - JavaScript code use to display selection of Content or Layout by color change.	(No tags)

Desktop Templates in the sampleportal Directory and Tag Reference

The templates described in this appendix are in the following provider subdirectories, beneath the `/etc/opt/SUNWps/desktop/sampleportal` directory:

- [MyFrontPageTemplatePanelContainer](#)

- [PredefinedFrontPageTemplatePanelContainerProvider](#) and [PredefinedSamplesTemplatePanelContainerProvider](#)
- [SamplesTemplatePanelContainer](#) and [ToolsTemplatePanelContainer](#)
- [TemplateTabContainerProvider](#), [TemplateTabCustomTableContainerProvider](#), and [TemplateTableContainer](#)
- [sampleportal](#)

MyFrontPageTemplatePanelContainer

[Table A-19](#) lists the templates in the `MyFrontPageTemplatePanelContainer` directory and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-19 Template Files in the `MyFrontPageTemplatePanelContainer` Subdirectory

File Name and Description	Tag Name and Description
<code>banner.template</code> - The banner across the top of the Desktop pages.	<p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:parentContainerName] - The top level container name for the <code>TemplateTabContainer</code></p> <p>[tag:theme_channel] - The theme edit channel name at run time, this will be either the <code>presetThemeContainer</code> or the <code>customThemeContainer</code>.</p> <p>[tag:help_link] - Help link for Desktop</p>
<code>menubar.template</code> - HTML for the menubar across the bottom of the Desktop pages, including the Edit, Layout, and Content pages, and containing Home, Help and Log Out links.	<p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:productName] - Product name</p> <p>[tag:parentContainerName] - The top level container name for the <code>TemplateTabContainer</code></p> <p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:theme_channel] - The theme edit channel name at run time, this will be either the <code>presetThemeContainer</code> or the <code>customThemeContainer</code>.</p> <p>[tag:help_link] - Help link for Desktop</p>

PredefinedFrontPageTemplatePanelContainerProvider

Table A-20 lists the templates in the

PredefinedFrontPageTemplatePanelContainerProvider directory and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-20 Template Files in the PredefinedFrontPageTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
banner.template - The banner across the top of the Desktop pages.	<p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:parentContainerName] - The top level container name for the TemplateTabContainer</p> <p>[tag:theme_channel] - The theme edit channel name at run time, this will be either the presetThemeContainer or the customThemeContainer.</p> <p>[tag:help_link] - Help link for Desktop</p>
bareProviderWrapper.template - Template for each provider wrapper with no titlebar.	<p>[tag:borderWidth] - Size of border (CELLPADDING) around provider HTML table</p> <p>[tag:size] - Table width (WIDTH), fixed at 100%</p> <p>[tag:borderColor] - Color of the channel border. The border color can be changed by user in the desktop custom theme page.</p> <p>[tag:bgColor] - Provider background color chosen</p> <p>[tag:fontColor] - Color of font chosen</p> <p>[tag:fontFace] - Font chosen</p> <p>[tag:content] - Content of the channel</p>
contentBarInContent.template - Content bar for template displayed when a user selects Content on any other page's Content bar.	<p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p>
contentBarInLayout.template - Content bar for template displayed when user selects Layout on any other page's Content bar.	<p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p>
contentLayout.template - Content bar for template displayed on the Desktop before the user selects Content or Layout.	<p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:selectedName] - The selected name</p>

Table A-20 Template Files in the PredefinedFrontPageTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
contentTemplate.template - The HTML template for the Content (Channels) page that displays when a user selects Content.	<p>[tag:productName] - Product name</p> <p>[tag:banner] - Inserts banner.template into this template</p> <p>[tag:contentBarInContent] - Inserts contentBarInContent.template into this template</p> <p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:thinProviders]</p> <p>[tag:wideProviders]</p> <p>[tag:desktop_url] - URL of Desktop to return to</p> <p>[tag:menubar] - Inserts menubar.template into this template</p>

Table A-20 Template Files in the PredefinedFrontPageTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
layout1Template.template - left/thin, right/wide.	<p>[tag:productName] - Product name</p> <p>[tag:arrangeProvider] - Inserts arrangeProvider.js template into this template</p> <p>[tag:performSubstitution] - Inserts performSubstitution.js template</p> <p>[tag:performColumnSubstitution] - Inserts performColumnSubstitution.js template</p> <p>[tag:selectAll] - Inserts selectAll.js template</p> <p>[tag:switchColumns] - Inserts switchColumns.js template</p> <p>[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates</p> <p>[tag:bulletColor] - Inserts bulletColor.js into this template</p> <p>[tag:banner] - Inserts banner.template into this template</p> <p>[tag:contentBarInLayout] - Inserts contentBarInLayout.template into this template</p> <p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:layoutFullTop] - Inserts layoutFullTop.template into this template</p> <p>[tag:leftUserProviderList] - List that shows left column channels</p> <p>[surl:/desktop/images/b_up.gif] - Icon that shows up arrow</p> <p>[surl:/desktop/images/b_down.gif] - Icon that shows down arrow</p> <p>[tag:rightUserProviderList] - List that shows right column channels</p> <p>[tag:layoutFullBottom] - Inserts layoutFullBottom.template into this template</p> <p>[tag:desktop_url] - URL of Desktop to return to</p> <p>[tag:menubar] - Inserts menubar.template into this template</p>

Table A-20 Template Files in the PredefinedFrontPageTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
layout2Template.template - left/wide, right/thin.	<p>[tag:productName] - Product name</p> <p>[tag:arrangeProvider] - Inserts arrangeProvider.js template</p> <p>[tag:performSubstitution] - Inserts performSubstitution.js template</p> <p>[tag:performColumnSubstitution] - Inserts performColumnSubstitution.js template</p> <p>[tag:selectAll] - Inserts selectAll.js template</p> <p>[tag:switchColumns] - Inserts switchColumns.js template</p> <p>[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates</p> <p>[tag:bulletColor] - Inserts bulletColor.js into this template</p> <p>[tag:banner] - Inserts banner.template into this template</p> <p>[tag:contentBarInLayout] - Inserts contentBarInLayout.template into this template</p> <p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:layoutFullTop] - Inserts layoutFullTop.template into this template</p> <p>[tag:leftUserProviderList] - List that shows left column channels</p> <p>[surl:/desktop/images/b_up.gif] - Icon that shows up arrow</p> <p>[surl:/desktop/images/b_down.gif] - Icon that shows down arrow</p> <p>[tag:rightUserProviderList] - List that shows right column channels</p> <p>[tag:layoutFullBottom] - Inserts layoutFullBottom.template template</p> <p>[tag:desktop_url] - URL of Desktop to return to</p> <p>[tag:menubar] - Inserts menubar.template into this template</p>

Table A-20 Template Files in the PredefinedFrontPageTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
layout3Template.template - left/thin, center/wide, right/thin.	<p>[tag:productName] - Product name</p> <p>[tag:arrangeProvider] - Inserts arrangeProvider.js template</p> <p>[tag:performSubstitution] - Inserts performSubstitution.js template</p> <p>[tag:performColumnSubstitution] - Inserts performColumnSubstitution.js template</p> <p>[tag:selectAll] - Inserts selectAll.js template</p> <p>[tag:switchColumns] - Inserts switchColumns.js template</p> <p>[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates</p> <p>[tag:bulletColor] - Inserts bulletColor.js into this template</p> <p>[tag:banner] - Inserts banner.template into this template</p> <p>[tag:contentBarInLayout] - Inserts contentBarInLayout.template into this template</p> <p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:layoutFullTop] - Inserts layoutFullTop.template template</p> <p>[tag:leftUserProviderList] - List that shows left column channels</p> <p>[surl:/desktop/images/b_up.gif] - Icon that shows up arrow</p> <p>[surl:/desktop/images/b_left.gif] - Icon that shows arrow pointing left</p>

Table A-20 Template Files in the PredefinedFrontPageTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
	[surl:/desktop/images/b_down.gif] - Icon that shows down arrow
	[surl:/desktop/images/b_right.gif] - Icon that shows arrow pointing right
	[tag:centerUserProviderList] - List that shows center column channels
	[tag:rightUserProviderList] - List that shows right column channels
	[tag:layoutFullBottom] - Inserts layoutFullBottom.template template
	[tag:desktop_url] - URL of Desktop to return to
	[tag:menubar] - Inserts menubar.template into template

Table A-20 Template Files in the PredefinedFrontPageTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
layout4Template.template - left/thin, center/thin, right/thin.	<p>[tag:productName] - Product name</p> <p>[tag:arrangeProvider] - Inserts arrangeProvider.js template</p> <p>[tag:performSubstitution] - Inserts performSubstitution.js template</p> <p>[tag:performColumnSubstitution] - Inserts performColumnSubstitution.js template</p> <p>[tag:selectAll] - Inserts selectAll.js template</p> <p>[tag:switchColumns] - Inserts switchColumns.js template</p> <p>[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates</p> <p>[tag:bulletColor] - Inserts bulletColor.js into this template</p> <p>[tag:banner] - Inserts banner.template template</p> <p>[tag:contentBarInLayout] - Inserts contentBarInLayout.template into this template</p> <p>[tag:frontContainerName] - Name of container</p> <p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:layoutFullTop] - Inserts layoutFullTop.template template</p> <p>[tag:leftUserProviderList] - List that shows left column channels</p> <p>[surl:/desktop/images/b_up.gif] - Icon that shows up arrow</p> <p>[surl:/desktop/images/b_down.gif] - Icon that shows down arrow</p>

Table A-20 Template Files in the PredefinedFrontPageTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
	[surl:/desktop/images/b_right.gif] - Icon that shows arrow pointing right
	[tag:centerUserProviderList] - List that shows center column channels
	[surl:/desktop/images/b_left.gif] - Icon that shows arrow pointing left
	[tag:rightUserProviderList] - List that shows right column channels
	[tag:layoutFullBottom] - Inserts layoutFullBottom.template template
	[tag:desktop_url] - URL of Desktop to return to
	[tag:menubar] - Inserts menubar.template template
layoutFullBottom.template - Partial HTML template for Layout pages when the user has a full width channel available at the bottom of the layout.	[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:fullbottomUserProviderList] - List of Full Width (Bottom) Channels to move
	[surl:/desktop/images/b_up.gif] - Icon that shows up arrow
	[surl:/desktop/images/b_down.gif] - Icon that shows down arrow
layoutFullTop.template - Partial HTML template for Layout pages when the user has a full width channel available at the top of the layout	[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:fulltopUserProviderList] - List of Full Width (Top) Channels to move
	[surl:/desktop/images/b_up.gif] - Icon that shows up arrow
	[surl:/desktop/images/b_down.gif] - Icon that shows down arrow

Table A-20 Template Files in the PredefinedFrontPageTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
maximizedTemplate.template - Template of a provider when it's in its maximized state.	[tag:launchPopup] - Inserts launchPoup.js template
	[tag:detachedContent] - Channel content in the detached window
	[tag:productName] - Product Name
	[tag:bulletColor] - Inserts bulletColor.js into this template
	[tag:banner] - Inserts banner.template template
	[tag:contentLayout] - Inserts contentLayout.template in this template
	[tag:inlineError] - Replaced with inlineError.template if error has occurred
	[tag:MaximizedContent] - Inserts content in the maximize mode
	[tag:menubar] - Inserts the menuber.template
	menubar.template - HTML for the menubar across the bottom of the Desktop pages, including the Edit, Layout, and Content pages and containing Home, Help and Log Out links.
[tag:productName] - Product name	
[tag:frontContainerName] - Name of container	
[tag:help_link] - Link to help from Desktop	
[tag:borderWidth] - Size of border (CELLPADDING) around channel (default 0)	
minimized.template - Template of a provider when it's in its minimized state - just the title/button bar showing, no content area showing.	[tag:size] - Table width (WIDTH), fixed at 100%
	[tag:borderColor] - Color of the channel border. The border color can be changed by user in the desktop custom theme page.
	[tag:titleBarColor] - Color of title bar
	[tag:fontFace] - Default font for the Desktop (sans-serif)
	[tag:title] - The title of the provider/channel
	[tag:titleBarColor] - Title bar color
	[tag:provider_cmds] - Inserts the channel command button links (maximize, detach, and so on)
	[surl:/desktop/images/nothing.gif]- Used to space gifs or channels

Table A-20 Template Files in the PredefinedFrontPageTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
optionsTemplate.template - Template for the Options page of the Desktop.	[tag:productName] - Product name
	[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates
	[tag:banner] - Inserts banner.template template
	[tag:frontContainerName] - Name of container
	[tag:inlineError] - Replaced with inlineError.template template if error has occurred.
	[tag:fontFace1] - Default font for the Desktop (sans-serif)
	[tag:serviceTimeout] - Provider Timeout in seconds
	[surl:/desktop/images/layout1.gif] - Icon for thin-wide layout
	[tag:layoutOneChecked] - Makes Radio button for layout one the default.
	[surl:/desktop/images/layout2.gif] - Icon for wide-thin layout
	[tag:layoutTwoChecked] - Makes Radio button for layout two the default.
	[surl:/desktop/images/layout3.gif] - Icon for thin-wide-thin layout
	[tag:layoutThreeChecked] - Makes Radio button for layout three the default.
	[surl:/desktop/images/layout4.gif] - Icon for thin-thin-thin layout
	[tag:layoutFourChecked] - Makes Radio button for layout four the default.
	[tag:desktop_url] - URL of Desktop to return to
	[tag:menubar] - Inserts menubar.template template
popupMenubar.template - Menubar to be used in popup windows.	[surl:/images/productName.gif]
	[tag:productName] - Product name
	[tag:fontFace1] - Default font for the Desktop (sans-serif)
	[tag:frontContainerName] - Name of the front container
	[tag:providerName] - The channel name

Table A-20 Template Files in the PredefinedFrontPageTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
<p>popupTemplate.template - Used to show provider/channel content in detached windows. Similar use to providerWrapper.template, but not in a table structure.</p>	<p>[tag:productName] - Product name</p> <p>[tag:openURLInParent] - Inserts openURLInParent.js template</p> <p>[tag:popupMenubar] - Inserts popupMenubar.template template</p> <p>[tag:providerContent] - Content of the provider</p>
<p>providerWrapper.template - Template that all providers and channels use for layout on Desktop. Defines the look of the border of the providers and channels on the screen.</p>	<p>[tag:borderWidth] - Size of border (CELLPADDING) around provider HTML table</p> <p>[tag:size] - Table width (WIDTH), fixed at 100%</p> <p>[tag:borderColor] - Color of the channel border. The border color can be changed by user in the desktop custom theme page.</p> <p>[tag:titleBarColor] - Color of title bar</p> <p>[tag:fontFace] - Font chosen</p> <p>[tag:title] - The title of the provider/channel</p> <p>[tag:provider_cmds] - Inserts the channel command button links (maximize, detach, and so on)</p> <p>[tag:bgColor] - Provider background color chosen</p> <p>[tag:fontColor] - Color of font chosen</p> <p>[tag:content] - Content of the channel</p>

Table A-20 Template Files in the PredefinedFrontPageTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
userTemplate.template - The base Desktop layout structure document. Very little in this file, as most of the content of the Desktop is swapped in during processing in the servlets.	[tag:launchPopup] - Inserts launchPoup.js template
	[tag:detachedContent] - Channel content in the detached window
	[tag:productName] - Product name
	[tag:bulletColor] - Inserts bulletColor.js into this template
	[tag:banner] - Inserts banner.template into this template
	[tag:contentLayout] - Inserts contentLayout.template in this template
	[tag:inlineError] - Replaced with inlineError.template if error has ocured
	[tag:fullTopContent] - Provider/channel content (HTML) inserted here
	[tag:leftWidth] - Percentage of total width for left column
	[tag:leftContent] - Channels for left column
	[tag:centerWidth] - Percentage of total width for center column
	[tag:centerContent] - Channels for center column
	[tag:rightWidth] - Percentage of total width for right column
	[tag:rightContent] - Channels for right column
[tag:fullBottomContent] - The content generated by the full-bottom channel	
[tag:menubar] - Inserts menubar.template into this template	
arrangeProvider.js - JavaScript code used in the Desktop Layout page.	No tags.
launchPopup.js - JavaScript code to launch a popup window.	No tags.
performColumnSubstitution.js - JavaScript code used on the Layout page.	No tags.
performSubstitution.js - JavaScript code used on the Layout page.	No tags.
removeProvider.js - JavaScript code used to remove a channel from the Desktop.	No tags.

Table A-20 Template Files in the PredefinedFrontPageTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
<code>selectAll.js</code> - JavaScript code used on the Layout page.	No tags.
<code>switchColumns.js</code> - JavaScript code used on the Layout page.	No tags.

PredefinedSamplesTemplatePanelContainerProvider

Table A-21 lists the templates in the `PredefinedSamplesTemplatePanelContainerProvider` directory and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-21 Template Files in the PredefinedSamplesTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
<code>banner.template</code> - The banner across the top of the Desktop pages.	<p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:parentContainerName] - The top level container name for the TemplateTabContainer</p> <p>[tag:theme_channel] - The theme edit channel name at run time, this will be either the presetThemeContainer or the customThemeContainer.</p> <p>[tag:help_link] - Help link for Desktop</p>
<code>bareProviderWrapper.template</code> - Template for each provider wrapper with no titlebar.	<p>[tag:borderWidth] - Size of border (CELLPADDING) around provider HTML table</p> <p>[tag:size] - Table width (WIDTH), fixed at 100%</p> <p>[tag:borderColor] - Color of the channel border. The border color can be changed by user in the desktop custom theme page.</p> <p>[tag:bgColor] - Provider background color chosen</p> <p>[tag:fontColor] - Color of font chosen</p> <p>[tag:fontFace] - Font chosen</p> <p>[tag:content] - Content of the channel</p> <p>[surl:/desktop/images/nothing.gif] - Used to space gifs or channels</p>

Table A-21 Template Files in the PredefinedSamplesTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
<code>contentBarInContent.template</code> - Content bar for template displayed when a user selects Content on any other page's Content bar.	<p>[tag:toolbarRollover] - Inserts <code>toolbarRollover.js</code> into this template.</p> <p>[url:/desktop/images/nothing.gif]- Used to space gifs or channels</p> <p>[tag:frontContainerName] - Name of container</p> <p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p>
<code>contentBarInLayout.template</code> - Content bar for template displayed when user selects Layout on any other page's Content bar.	<p>[tag:toolbarRollover] - Inserts <code>toolbarRollover.js</code> into this template.</p> <p>[url:/desktop/images/nothing.gif]- Used to space gifs or channels</p> <p>[tag:frontContainerName] - Name of container</p> <p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p>
<code>contentLayout.template</code> - Content bar for template displayed on the Desktop before the user selects Content or Layout.	<p>[tag:toolbarRollover] - Inserts <code>toolbarRollover.js</code> into this template.</p> <p>[url:/desktop/images/nothing.gif]- Used to space gifs or channels</p> <p>[tag:frontContainerName] - Name of container</p> <p>[tag:selectedName] - Name of selected tab</p> <p>[tag:fontFace1] - Default font for the Desktop (sans-serif)</p>
<code>contentTemplate.template</code> - The HTML template for the Content (Channels) page that displays when a user selects Content.	<p>[tag:productName] - Product name</p> <p>[tag:banner] - Inserts <code>banner.template</code> into this template</p> <p>[tag:contentBarInContent] - Inserts <code>contentBarInContent.template</code> into this template</p> <p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:thinProviders]</p> <p>[tag:wideProviders]</p> <p>[tag:desktop_url] - URL of Desktop to return to</p> <p>[tag:menubar] - Inserts <code>menubar.template</code> into this template</p>

Table A-21 Template Files in the PredefinedSamplesTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
layout1Template.template - left/thin, right/wide.	<p>[tag:productName] - Product name</p> <p>[tag:arrangeProvider] - Inserts arrangeProvider.js template into this template</p> <p>[tag:performSubstitution] - Inserts performSubstitution.js template</p> <p>[tag:performColumnSubstitution] - Inserts performColumnSubstitution.js template</p> <p>[tag:selectAll] - Inserts selectAll.js template</p> <p>[tag:switchColumns] - Inserts switchColumns.js template</p> <p>[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates</p> <p>[tag:bulletColor] - Inserts bulletColor.js into this template</p> <p>[tag:banner] - Inserts banner.template into this template</p> <p>[tag:contentBarInLayout] - Inserts contentBarInLayout.template into this template</p> <p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:layoutFullTop] - Inserts layoutFullTop.template into this template</p> <p>[tag:leftUserProviderList] - List that shows left column channels</p> <p>[surl:/desktop/images/b_up.gif] - Icon that shows up arrow</p> <p>[surl:/desktop/images/b_down.gif] - Icon that shows down arrow</p> <p>[tag:rightUserProviderList] - List that shows right column channels</p> <p>[tag:layoutFullBottom] - Inserts layoutFullBottom.template into this template</p> <p>[tag:desktop_url] - URL of Desktop to return to</p> <p>[tag:menubar] - Inserts menubar.template into this template</p>

Table A-21 Template Files in the PredefinedSamplesTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
layout2Template.template - left/wide, right/thin.	<p>[tag:productName] - Product name</p> <p>[tag:arrangeProvider] - Inserts arrangeProvider.js template</p> <p>[tag:performSubstitution] - Inserts performSubstitution.js template</p> <p>[tag:performColumnSubstitution] - Inserts performColumnSubstitution.js template</p> <p>[tag:selectAll] - Inserts selectAll.js template</p> <p>[tag:switchColumns] - Inserts switchColumns.js template</p> <p>[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates</p> <p>[tag:bulletColor] - Inserts bulletColor.js into this template</p> <p>[tag:banner] - Inserts banner.template into this template</p> <p>[tag:contentBarInLayout] - Inserts contentBarInLayout.template into this template</p> <p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:layoutFullTop] - Inserts layoutFullTop.template into this template</p> <p>[tag:leftUserProviderList] - List that shows left column channels</p> <p>[surl:/desktop/images/b_up.gif] - Icon that shows up arrow</p> <p>[surl:/desktop/images/b_down.gif] - Icon that shows down arrow</p> <p>[tag:rightUserProviderList] - List that shows right column channels</p> <p>[tag:layoutFullBottom] - Inserts layoutFullBottom.template template</p> <p>[tag:desktop_url] - URL of Desktop to return to</p> <p>[tag:menubar] - Inserts menubar.template into this template</p>

Table A-21 Template Files in the PredefinedSamplesTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
layout3Template.template - left/thin, center/wide, right/thin.	<p>[tag:productName] - Product name</p> <p>[tag:arrangeProvider] - Inserts arrangeProvider.js template</p> <p>[tag:performSubstitution] - Inserts performSubstitution.js template</p> <p>[tag:performColumnSubstitution] - Inserts performColumnSubstitution.js template</p> <p>[tag:selectAll] - Inserts selectAll.js template</p> <p>[tag:switchColumns] - Inserts switchColumns.js template</p> <p>[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates</p> <p>[tag:bulletColor] - Inserts bulletColor.js into this template</p> <p>[tag:banner] - Inserts banner.template into this template</p> <p>[tag:contentBarInLayout] - Inserts contentBarInLayout.template into this template</p> <p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:layoutFullTop] - Inserts layoutFullTop.template template</p> <p>[tag:leftUserProviderList] - List that shows left column channels</p> <p>[surl:/desktop/images/b_up.gif] - Icon that shows up arrow</p> <p>[surl:/desktop/images/b_left.gif] - Icon that shows arrow pointing left</p>

Table A-21 Template Files in the PredefinedSamplesTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
	[surl:/desktop/images/b_down.gif] - Icon that shows down arrow
	[surl:/desktop/images/b_right.gif] - Icon that shows arrow pointing right
	[tag:centerUserProviderList] - List that shows center column channels
	[tag:rightUserProviderList] - List that shows right column channels
	[tag:layoutFullBottom] - Inserts layoutFullBottom.template template
	[tag:desktop_url] - URL of Desktop to return to
	[tag:menubar] - Inserts menubar.template into template

Table A-21 Template Files in the PredefinedSamplesTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
layout4Template.template - left/thin, center/thin, right/thin.	<p>[tag:productName] - Product name</p> <p>[tag:arrangeProvider] - Inserts arrangeProvider.js template</p> <p>[tag:performSubstitution] - Inserts performSubstitution.js template</p> <p>[tag:performColumnSubstitution] - Inserts performColumnSubstitution.js template</p> <p>[tag:selectAll] - Inserts selectAll.js template</p> <p>[tag:switchColumns] - Inserts switchColumns.js template</p> <p>[surl:/desktop/css/style.css] - Style sheet used by the Desktop for the banner and tabs templates</p> <p>[tag:bulletColor] - Inserts bulletColor.js into this template</p> <p>[tag:banner] - Inserts banner.template template</p> <p>[tag:contentBarInLayout] - Inserts contentBarInLayout.template into this template</p> <p>[tag:frontContainerName] - Name of container</p> <p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:layoutFullTop] - Inserts layoutFullTop.template template</p> <p>[tag:leftUserProviderList] - List that shows left column channels</p> <p>[surl:/desktop/images/b_up.gif] - Icon that shows up arrow</p> <p>[surl:/desktop/images/b_down.gif] - Icon that shows down arrow</p>

Table A-21 Template Files in the PredefinedSamplesTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
	[surl:/desktop/images/b_right.gif] - Icon that shows arrow pointing right
	[tag:centerUserProviderList] - List that shows center column channels
	[surl:/desktop/images/b_left.gif] - Icon that shows arrow pointing left
	[tag:rightUserProviderList] - List that shows right column channels
	[tag:layoutFullBottom] - Inserts layoutFullBottom.template template
	[tag:desktop_url] - URL of Desktop to return to
	[tag:menubar] - Inserts menubar.template template
layoutFullBottom.template - Partial HTML template for Layout pages when the user has a full width channel available at the bottom of the layout.	[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:fullbottomUserProviderList] - List of Full Width (Bottom) Channels to move
	[surl:/desktop/images/b_up.gif] - Icon that shows up arrow
	[surl:/desktop/images/b_down.gif] - Icon that shows down arrow
layoutFullTop.template - Partial HTML template for Layout pages when the user has a full width channel available at the top of the layout	[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:fulltopUserProviderList] - List of Full Width (Top) Channels to move
	[surl:/desktop/images/b_up.gif] - Icon that shows up arrow
	[surl:/desktop/images/b_down.gif] - Icon that shows down arrow

Table A-21 Template Files in the PredefinedSamplesTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
maximizedTemplate.template - Template of a provider when it's in its maximized state.	[tag:launchPopup] - Inserts launchPoup.js template
	[tag:detachedContent] - Channel content in the detached window
	[tag:productName] - Product name
	[tag:bulletColor] - Inserts bulletColor.js into this template
	[tag:banner] - Inserts banner.template into this template
	[tag:contentLayout] - Inserts contentLayout.template in this template
	[tag:inlineError] - Replaced with inlineError.template if error has ocured
menubar.template - HTML for the menubar across the bottom of the Desktop pages, including the Edit, Layout, and Content pages, and containing Home, Help and Log Out links.	[tag:MaximizedContent] - Inserts content in the maximize mode
	[tag:menubar] - Inserts menubar.template into this template
	[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:productName] - Product name
	[tag:parentContainerName] - The top level container name for the TemplateTabContainer
	[tag:frontContainerName] - Name of the front container
	[tag:theme_channel] - The theme edit channel name at run time, this will be either the presetThemeContainer or the customThemeContainer.
minimized.template - Template of a provider when it's in its minimized state - just the title/button bar are displayed and no content area is displayed.	[tag:help_link] - Help link for Desktop
	[tag:borderWidth] - Size of border (CELLPADDING) around provider HTML table
	[tag:size] - Table width (WIDTH), fixed at 100%
	[tag:borderColor] - Color of the channel border. The border color can be changed by user in the desktop custom theme page.
	[tag:titleBarColor] - Color of title bar
	[tag:fontFace] - Font chosen
	[tag:title] - The title of the provider/channel
[tag:provider_cmds] - Inserts the channel command button links (maximize, detach, and so on)	

Table A-21 Template Files in the PredefinedSamplesTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
optionsTemplate.template - Template for the Options page of the Desktop.	[tag:productName] - Product name
	[tag:bulletColor] - Inserts bulletColor.js into this template
	[tag:banner] - Inserts banner.template into this template
	[tag:frontContainerName] - Name of the front container
	[tag:inlineError] - Replaced with inlineError.template if error has occurred
	[tag:fontFace1] - Default font for the Desktop (sans-serif)
	[tag:serviceTimeout] - Provider Timeout in seconds
	[tag:layoutOneChecked] - Makes Radio button for layout one the default.
	[url:/desktop/images/layout2.gif] - Icon for wide-thin layout
	[tag:layoutTwoChecked] - Makes Radio button for layout two the default.
	[url:/desktop/images/layout3.gif] - Icon for thin-wide-thin layout
	[tag:layoutThreeChecked] - Makes Radio button for layout three the default.
	[url:/desktop/images/layout4.gif] - Icon for thin-thin-thin layout
	[tag:layoutFourChecked] - Makes Radio button for layout four the default.
	[tag:desktop_url] - URL of Desktop to return to
[tag:menubar] - Inserts menubar.template template	
popupMenubar.template - Menubar to be used in popup windows.	[url:/images/productName.gif]
	[tag:productName] - Product name
	[tag:fontFace1] - Default font for the Desktop (sans-serif)
	[tag:frontContainerName] - Name of the front container
	[tag:providerName] - The channel name

Table A-21 Template Files in the PredefinedSamplesTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
<p>popupTemplate.template - Used to show provider/channel content in detached windows. Similar use to providerWrapper.template, but not in a table structure.</p>	<p>[tag:productName] - Product name</p> <p>[tag:openURLInParent] - Inserts openURLInParent.js template</p> <p>[tag:popupMenubar] - Inserts popupMenubar.template template</p> <p>[tag:providerContent] - Content of the provider</p>
<p>providerWrapper.template - Template that all providers and channels use for layout on Desktop. Defines the look of the border of the providers and channels on the screen.</p>	<p>[tag:borderWidth] - Size of border (CELLPADDING) around provider HTML table</p> <p>[tag:size] - Table width (WIDTH), fixed at 100%</p> <p>[tag:borderColor] - Color of the channel border. The border color can be changed by user in the desktop custom theme page.</p> <p>[tag:titleBarColor] - Color of title bar</p> <p>[tag:fontFace] - Font chosen</p> <p>[tag:title] - The title of the provider/channel</p> <p>[tag:provider_cmds] - Inserts the channel command button links (maximize, detach, and so on)</p> <p>[tag:bgColor] - Provider background color chosen</p> <p>[tag:fontColor] - Color of font chosen</p> <p>[tag:content] - Content of the channel</p>

Table A-21 Template Files in the PredefinedSamplesTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
userTemplate.template - The base Desktop layout structure document. Very little in this file, as most of the content of the Desktop is swapped in during processing in the servlets.	[tag:launchPopup] - Inserts launchPoup.js template
	[tag:detachedContent] - Channel content in the detached window
	[tag:productName] - Product name
	[tag:bulletColor] - Inserts bulletColor.js into this template
	[tag:banner] - Inserts banner.template into this template
	[tag:contentLayout] - Inserts contentLayout.template in this template
	[tag:inlineError] - Replaced with inlineError.template if error has ocured
	[tag:fullTopContent] - Provider/channel content (HTML) inserted here
	[tag:leftWidth] - Percentage of total width for left column
	[tag:leftContent] - Channels for left column
	[tag:centerWidth] - Percentage of total width for center column
	[tag:centerContent] - Channels for center column
	[tag:rightWidth] - Percentage of total width for right column
[tag:rightContent] - Channels for right column	
[tag:fullBottomContent] - The content generated by the full-bottom channel	
[tag:menubar] - Inserts menubar.template into this template	
launchPopup.js - JavaScript code to launch a popup window.	No Tags
launchPopup.js - JavaScript code to launch a popup window.	No Tags
performColumnSubstitution.js - JavaScript code used on the Layout page.	No Tags
performSubstitution.js - JavaScript code used on the Layout page.	No Tags
removeProvider.js - JavaScript code used to remove a channel from the Desktop.	No Tags

Table A-21 Template Files in the PredefinedSamplesTemplatePanelContainerProvider Subdirectory

File Name and Description	Tag Name and Description
<code>selectAll.js</code> - JavaScript code used on the Layout page.	No Tags
<code>switchColumns.js</code> - JavaScript code used on the Layout page.	No Tags

SamplesTemplatePanelContainer

[Table A-22](#) lists the templates in the `SamplesTemplatePanelContainer` directory and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-22 Template Files in the SamplesTemplatePanelContainer Subdirectory

File Name and Description	Tag Name and Description
<code>banner.template</code> - The banner across the top of the Desktop pages.	<p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:parentContainerName] - The top level container name for the TemplateTabContainer</p> <p>[tag:theme_channel] - The theme edit channel name at run time, this will be either the presetThemeContainer or the customThemeContainer.</p> <p>[tag:help_link] - Help link for Desktop</p>
<code>menubar.template</code> - HTML for the menubar across the bottom of the Desktop pages, including the Edit, Layout, and Content pages, and containing Home, Help and Log Out links.	<p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:productName] - Product name</p> <p>[tag:parentContainerName] - The top level container name for the TemplateTabContainer</p> <p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:theme_channel] - The theme edit channel name at run time, this will be either the presetThemeContainer or the customThemeContainer.</p> <p>[tag:help_link] - Help link for Desktop</p>

TemplateTabContainerProvider

[Table A-23](#) lists the templates in the `TemplateTabContainerProvider` directory and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-23 Template Files in the TemplateTabContainerProvider Subdirectory

File Name and Description	Tag Name and Description
banner.template - The banner across the top of the Desktop pages.	<p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:parentContainerName] - The top level container name for the TemplateTabContainer</p> <p>[tag:theme_channel] - The theme edit channel name at run time, this will be either the presetThemeContainer or the customThemeContainer.</p> <p>[tag:help_link] - Help link for Desktop</p>
menubar.template - HTML for the menubar across the bottom of the Desktop pages, including the Edit, Layout, and Content pages, and containing Home, Help and Log Out links.	<p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:productName] - Product name</p> <p>[tag:parentContainerName] - The top level container name for the TemplateTabContainer</p> <p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:theme_channel] - The theme edit channel name at run time, this will be either the presetThemeContainer or the customThemeContainer.</p> <p>[tag:help_link] - Help link for Desktop</p>

TemplateTabCustomTableContainerProvider

[Table A-24](#) lists the templates in the

TemplateTabCustomTableContainerProvider directory and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-24 Template Files in the TemplateTabCustomTableContainerProvider Subdirectory

File Name and Description	Tag Name and Description
banner.template - The banner across the top of the Desktop pages.	<p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:parentContainerName] - The top level container name for the TemplateTabContainer</p> <p>[tag:theme_channel] - The theme edit channel name at run time, this will be either the presetThemeContainer or the customThemeContainer.</p> <p>[tag:help_link] - Help link for Desktop</p>

Table A-24 Template Files in the TemplateTabCustomTableContainerProvider Subdirectory *(Continued)*

File Name and Description	Tag Name and Description
menubar.template - HTML for the menubar across the bottom of the Desktop pages, including the Edit, Layout, and Content pages, and containing Home, Help and Log Out links.	[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:productName] - Product name
	[tag:parentContainerName] - The top level container name for the TemplateTabContainer
	[tag:frontContainerName] - Name of the front container
	[tag:theme_channel] - The theme edit channel name at run time, this will be either the presetThemeContainer or the customThemeContainer.
	[tag:help_link] - Help link for Desktop

TemplateTableContainer

Table A-25 lists the templates in the TemplateTableContainer directory and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-25 Template Files in the TemplateTableContainer Subdirectory

File Name and Description	Tag Name and Description
banner.template - The banner across the top of the Desktop pages.	[surl:/images/productName.gif]
	[tag:productName] - Product name
	[tag:theme_channel] - The theme edit channel name at run time, this will be either the presetThemeContainer or the customThemeContainer.
	[tag:frontContainerName] - Name of the front container
	[tag:help_link] - Help link for Desktop
menubar.template - HTML for the menubar across the bottom of the Desktop pages, including the Edit, Layout, and Content pages, and containing Home, Help and Log Out links.	[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)
	[tag:productName] - Product name
	[tag:frontContainerName] - Name of the front container
	[tag:theme_channel] - The theme edit channel name at run time, this will be either the presetThemeContainer or the customThemeContainer.
	[tag:help_link] - Help link for Desktop

ToolsTemplatePanelContainer

[Table A-17](#) lists the templates in the `ToolsTemplatePanelContainer` directory and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-26 Template Files in the `ToolsTemplatePanelContainer` Subdirectory

File Name and Description	Tag Name and Description
<code>banner.template</code> - The banner across the top of the Desktop pages.	<p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:parentContainerName] - The top level container name for the <code>TemplateTabContainer</code></p> <p>[tag:theme_channel] - The theme edit channel name at run time, this will be either the <code>presetThemeContainer</code> or the <code>customThemeContainer</code>.</p> <p>[tag:help_link] - Help link for Desktop</p>
<code>menubar.template</code> - HTML for the menubar across the bottom of the Desktop pages, including the Edit, Layout, and Content pages, and containing Home, Help and Log Out links.	<p>[tag:desktop-fontFace1] - Default font for the Desktop (sans-serif)</p> <p>[tag:productName] - Product name</p> <p>[tag:parentContainerName] - The top level container name for the <code>TemplateTabContainer</code></p> <p>[tag:frontContainerName] - Name of the front container</p> <p>[tag:theme_channel] - The theme edit channel name at run time, this will be either the <code>presetThemeContainer</code> or the <code>customThemeContainer</code>.</p> <p>[tag:help_link] - Help link for Desktop</p>

sampleportal

[Table A-27](#) lists the templates in the `sampleportal` directory and includes a brief description of the template file. It also includes a listing with description of the tags used in each template file.

Table A-27 Template Files in the sampleportal Directory

File Name and Description	Tag Name and Description
<p><code>AttachCommand.template</code> - Handles reattaching a detached channel.</p>	<p>[tag:detachAttachURL] - URL of the channel to be detached</p> <p><code>?action=process&provider=<i>thecontainername</i>&thecontainername.channelAction=detach&<i>thecontainername</i>.targetProvider=<i>providername</i></code></p> <p>[surl:/desktop/images/b_attach.gif] - A URL that points to the attach image</p> <p>[tag:detachAttachText] - Alternated string for the detach/attach image in the title bar, this is a localized string. The detach or attach mode is determined at run time.</p> <p>[tag:providerName] - The channel name</p>
<p><code>MaximizeCommand.template</code> - Allows the channel to be displayed in the "maximize" mode so that the channel occupies the entire Desktop.</p>	<p>[tag:maximizeURL] - A URL to show the channel in maximize mode</p> <p>[surl:/desktop/images/b_maximize.gif] - A URL that points to the maximize image</p> <p>[tag:maximizeText] - A text that is used as an alternate for the maximize icon</p> <p>[tag:providerName] - The channel name</p>
<p><code>detachEditCommand.template</code> - Handles link to the Edit page for the detached channel.</p>	<p>[tag:editURL] - URL of Edit page for this channel</p> <p>[tag:s_editImage] - Run time path for the edit image</p> <p>[tag:editTag] - Text for the alt tag for the edit icon</p> <p>[tag:providerName] - The channel name</p>
<p><code>detachCommand.template</code> - Handles detaching a channel.</p>	<p>[tag:detachAttachURL] - URL of the channel to be detached</p> <p>[tag:s_detachImage] - Run time path for the detach image</p> <p>[tag:detachAttachText] - Alternated string for the detach/attach image in the title bar, this is a localized string. The detach or attach mode is determined at run time.</p> <p>[tag:providerName] - The channel name</p>

Table A-27 Template Files in the sampleportal Directory *(Continued)*

File Name and Description	Tag Name and Description
<code>detachRemoveCommand.template</code> - Handles closing or removing a detached channel.	<p>[tag:removeURL] - URL of the channel to be removed</p> <p>[tag:s_removeImage] - Run time path for the remove image</p> <p>[tag:removeTag] - Alternated string for the remove image in the title bar, this is a localized string</p> <p>[tag:providerName] - The channel name</p>
<code>editCommand.template</code> - Handles link to the Edit page for the detached channel.	<p>[tag:editURL] - URL of Edit page for this channel</p> <p>[tag:s_editImage] - Run time path for the edit image</p> <p>[tag:editTag] - Text for the alt tag for the edit icon</p> <p>[tag:providerName] - The channel name</p>
<code>helpHref.template</code> - Generates the help URL for each of the channels. Displays the help contents in a new window.	<p>[tag:help_link] - Help link for Desktop -</p> <p>[tag:s_helpImage] - Run time path for the help image</p> <p>[tag:providerName] - The channel name</p> <p>[tag:help_tag] - Text for alt message</p>
<code>minimizeCommand.template</code> - Allows the channel to be displayed in the "minimize" mode so that only the title bar of the channel is displayed and no content of the channel is displayed.	<p>[tag:minMaximizeURL] - A URL to show the channel in either the minimize or the normal mode, the mode is decided at run time</p> <p>[tag:s_minimizeImage] - Run time path for the minimized or the maximized image, the state of minimize or maximize is determined at run time.</p> <p>[tag:minMaximizeText] - Alternated string for the minimize or normalize image in the title bar, this is a localized string, the state of minimize or maximize is determined at run time.</p> <p>[tag:providerName] - The channel name</p>
<code>normalizeCommand.template</code> - Allows the channel to be displayed in the "normal" mode so that the channel is displayed in the Desktop, with all other channels in the same table container.	<p>[tag:minMaximizeURL] - A URL to show the channel in either the minimize or the normal mode, the mode is decided at run time</p> <p>[tag:s_normalizeImage] - Run time path for the normalized image</p> <p>[tag:minMaximizeText] - Alternated string for the minimize or normalize image in the title bar, this is a localized string, the state of minimize or maximize is determined at run time.</p> <p>[tag:providerName] - The channel name</p>

Table A-27 Template Files in the sampleportal Directory *(Continued)*

File Name and Description	Tag Name and Description
removeCommand.template - Removes the channel.	[tag:removeURL] - URL of the channel to be removed [tag:s_removeImage] - Run time path for the remove image [tag:removeTag] - Alternated string for the remove image in the title bar, this is a localized string [tag:providerName] - The channel name

JavaServer Pages Tag Library Reference

This appendix describes the Desktop custom tag library and serves as a reference to the tags available within the library.

This appendix contains the following sections:

- [Tag Library Overview](#)
- [Using Tags in JSPs](#)
- [Tag Reference](#)
- [Tag Overview](#)

See the following supplemental documentation for more information on JavaServer Pages™ (JSP™) Custom Tags and Apache's JSP Standard Tag Library (JSTL):

<http://developer.java.sun.com/developer/technicalArticles/xml/WebAppDev3/>

<http://jakarta.apache.org/taglibs/doc/standard-doc/>

Tag Library Overview

In JavaServer Pages technology, actions are elements that can create and access programming language objects and affect the output stream. JSP technology supports reusable modules called custom actions. You invoke a custom action by using a custom tag in a JSP. A tag library is a collection of custom tags. The Desktop custom tag library contains tags that you use to perform Desktop operations for JSPs.

Before tag libraries, JSPs were difficult to maintain because you were forced to use JavaBeans™ components and scriptlets as the main mechanism for performing tasks. Custom actions, that is, a tag library, alleviate this problem by bringing the benefits of another level of componentization to JSP. A tag library encapsulates recurring tasks so that they can be reused across more than one application.

The Sun™ ONE Portal Server software Desktop tag library consists of six parts:

- Core tags that can be used on any provider or container that implement the PAPI interface.
- Tags that can be used to operate on a provider or container that support the `ProviderContext` and `ContainerProviderContext` interfaces.
- Tags that operate on specific container building-block providers (`SingleContainer`, `TableContainer`, `TabContainer`, and so on).
- JSP Standard tag libraries from Apache.
- Tags that support the Search function.
- Tags that provide theme support in the Desktop.

The Desktop tag library has the following Tag Library Descriptors (TLDs) in the `/etc/opt/SUNWps/desktop/default/tld` directory. The tag library is exposed, for convenience, through using multiple TLDs, so that tags are in their appropriate functional area.

<code>desktop.tld</code>	Contains core Desktop tags that bring forward the functionality available from the <code>ProviderContext</code> interface.
<code>desktopContainerProviderContext.tld</code>	Contains tags that bring forward the functionality available from the <code>ContainerProviderContext</code> interface.
<code>desktopProviderContext.tld</code>	Contains tags to operate on providers that extend the <code>ProviderContext</code>
<code>desktopSingle.tld</code>	Contains tags to operate on single container channels. Single container channels are based on <code>JSPSingleContainerProvider</code> or a subclass thereof.
<code>desktopTable.tld</code>	Contains tags to operate on table container channels. Table container channels are based on <code>JSPTableContainerProvider</code> or a subclass thereof.

<code>desktopTab.tld</code>	Contains tags to operate on tab container channels. Tab container channels are based on <code>JSPTabContainerProvider</code> or a subclass thereof.
<code>desktopTheme.tld</code>	Contains tags for theme support in the Desktop.
<code>im.tld</code>	Contains tags for <code>IMProvider</code> class.
<code>jr.tld</code>	Contains tags that accept <code>rtexprvalues</code> for their attributes. This is a JSP Standard tag library from Apache.
<code>jx.tld</code>	Contains tags that accept attribute values specified using the “expression languages” that JSPTL introduces, which currently is only simplest possible expression language (SPEL). This is a JSP Standard tag library from Apache.
<code>search.tld</code>	Contains tags for search support in the Desktop.

The many of the Java™ classes that support these tags reside in a JAR file `desktop.tl.jar` in the *Web-Container-Instance/portal/web-apps/WEB-INF/lib* directory. The classes for the `jx.tld` and `jr.tld` tags reside in the `jsptl.jar` file.

Desktop Tag Library Hierarchy

The Desktop tag library can be viewed as a wrapper of `PAPI`, `ProviderContext`, `ContainerProviderContext`, and specific container building-block providers in Sun ONE Portal Server software. Thus, a hierarchy is implied.

For example, the `ContainerProviderContext` (interface) extends `ProviderContext` (interface). When you use a tag in `desktopContainerProviderContext.tld`, it also make sense to use it in `desktopProviderContext.tld`. Similarly, when you use a tag in `desktopProviderContext.tld`, it also makes sense to use provider tags in `desktop.tld` because `ProviderAdapter` implements `Provider`. By putting the tag in the level that provides the most use, you will not have to make duplicate tags.

At the bottom of this chain are the specific containers (single, table, and tab). Because all these containers extend `JSPContainerProviderAdapter`, they can use tags in their respective TLDs, as well as tags in `desktopContainerProviderContext.tld`, `desktopProviderContext.tld`, and `desktop.tld`.

Using Tags in JSPs

In your JSP you can use HTML, Java, JavaScript™, and tags to bring in repetitive Java functions. Tags encapsulate core functionality common to many JSP applications.

In [Code Example B-1](#), a JSP for a single channel, the tag libraries are defined at the top. The library's symbol is then used with the tag name to bring in the appropriate information or function.

Code Example B-1 Listing of `single.jsp`

```
<%--
  Copyright 2001 Sun Microsystems, Inc. All rights reserved.
  PROPRIETARY/CONFIDENTIAL. Use of this product is subject to license terms.
--%>
<%-- single.jsp --%>

<%@ taglib uri="/tld/desktop.tld" prefix="dt" %>
<%@ taglib uri="/tld/desktopSingle.tld" prefix="dtsingle" %>

<%@ page session="false" %>

<dt:obtainContainer container="$JSPProvider">
  <dtsingle:singleContainerProvider>
    <dtsingle:obtainSelectedChannel>
<%@ include file="header.jsp" %>
<br>

<!-- provider content goes here -->
  <dt:getContent/>
  </dtsingle:obtainSelectedChannel>
<%@ include file="menubar.jsp" %>
<%@ include file="footer.html" %>
  </dtsingle:singleContainerProvider>
</dt:obtainContainer>
```

In [Code Example B-1](#), the following lines define the tag libraries that are used in the `single.jsp` template file.

```
<%@ taglib uri="/tld/desktop.tld" prefix="dt" %>
<%@ taglib uri="/tld/desktopSingle.tld" prefix="dtsingle" %>
```

NOTE You need to include a taglib directive in the page before any custom tag is used.

The `dt:obtainContainer container="$JSPProvider"` tag gets a return value by reference (see [“Attributes and Return Values” on page 436](#) for more information).

Three tags are used out of the `desktop.tld` library. They are `obtainContainer`, `scontent`, and `getContent`. See [Code Example B-2](#) for a partial listing of `desktop.tld` including these three tags. The class and property information about each tag is defined here.

Code Example B-2 Partial Listing of `desktop.tld`

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE taglib PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.1//EN"
"http://java.sun.com/j2ee/dtds/web-jsptaglibrary_1_1.dtd">

<taglib>
  <tlibversion>0.1</tlibversion>
  <jspversion>1.1</jspversion>
  <shortname>desktop</shortname>

  <tag>
    <name>obtainChannel</name>
    <tagclass>com.sun.portal.desktop.taglib.provider.ObtainChannelTag</tagclass>
    <bodycontent>JSP</bodycontent>
    <attribute>
      <name>channel</name>
      <required>true</required>
      <rtexprvalue>>false</rtexprvalue>
    </attribute>
  </tag>
  <tag>
    <name>obtainContainer</name>
    <tagclass>com.sun.portal.desktop.taglib.container.ObtainContainerTag</tagclass>
    <bodycontent>JSP</bodycontent>
    <attribute>
      <name>container</name>
      <required>true</required>
      <rtexprvalue>>false</rtexprvalue>
    </attribute>
  </tag>

  -----clip

  <tag>
    <name>getContent</name>
    <tagclass>com.sun.portal.desktop.taglib.provider.GetContentTag</tagclass>
```

Code Example B-2 Partial Listing of `desktop.tld` (Continued)

```

        <bodycontent>empty</bodycontent>
    </tag>
-----clip

    <!-- Utility Tag, can be used anywhere -->
    <tag>
        <name>scontent</name>
        <tagclass>com.sun.portal.desktop.taglib.util.SContentTag</tagclass>
        <bodycontent>empty</bodycontent>
    </tag>
</taglib>

```

In [Code Example B-2 on page 431](#), the `bodycontent` property has a value of `JSP` or `empty`. The `JSP` value means that you can add more content including more Java code or JSP tags between the tag open and tag close for this particular tag. The `empty` value means that a tag can contain no extra content and is complete in itself.

In [Code Example B-2 on page 431](#), the `required` property has a value of `true` meaning that a value for that attribute is required. If the `required` property has a value of `false`, a value for the attribute is optional. The `rtexprvalue` or run time expression value means that an attribute value can be evaluated at run time or can have a hardcoded value. By default, all of the core, provider, and container tags have this value as `false`.

Two tags are used out of the `desktopSingle.tld` library. They are `singleContainerProvider` and `obtainSelectedChannel`. See [Code Example B-3](#) for the listing of `desktopSingle.tld` with these two tags. The class and property information about each tag is defined here.

Code Example B-3 Listing of `desktopSingle.tld`

```

<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE taglib PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.1//EN"
"http://java.sun.com/j2ee/dtds/web-jsptaglibrary_1_1.dtd">

<taglib>
    <tlibversion>0.1</tlibversion>
    <jspversion>1.1</jspversion>
    <shortname>desktopSingle</shortname>

    <!-- Validator Tag -->

    <tag>
        <name>singleContainerProvider</name>

```

Code Example B-3 Listing of `desktopSingle.tld` (*Continued*)

```

<tagclass>com.sun.portal.desktop.taglib.container.single.SingleContainerProviderTag</tagclass>
</tag>
  <bodycontent>JSP</bodycontent>
</tag>

  <tag>
    <name>obtainSelectedChannel</name>

<tagclass>com.sun.portal.desktop.taglib.container.single.ObtainSelectedChannelTag</tagclass>
>
  <bodycontent>JSP</bodycontent>
</tag>
</taglib>

```

The first tag in `desktopSingle.tld` is a validator tag. The validator tag ensures that the JSP using the tag library is a valid type, looking for single container information, not tab information.

Using the Desktop Tag Library in Your Application

To use the Desktop tag library within your web application, you need to set up the context properly by using the Context Setup tags (tags with the prefix `obtain`). See [Table B-1 on page 438](#) for the Context Setup tags.

Because it is possible to have provider(s) within a container, to access a channel within the container you would need the following (partial JSP with `obtainContainer` and `obtainChannelFromContainer` tags):

Code Example B-4 Partial JSP

```

<desktop:obtainContainer container="$JSPPProvider">
  <!-- make sure we can use the obtainChannelsFromContainer ->
  <desktopcpc:containerProviderContext>
    <desktopcpc:obtainChannelFromContainer channel="myChannel">
      <!-- tags to operate on channel "myChannel" within the container stored as attribute
      "JSPPProvider" in page context-->

```

Code Example B-4 Partial JSP (Continued)

```

    </desktopcpc:obtainChannelFromContainer>
  </desktopcpc:containerProviderContext>
</desktop:obtainContainer>

```

The Desktop tag library also provides an `obtainParentContainer` tag, which gives you access to the parent container anytime (adding the `obtainParentContainer` tag to the partial JSP):

Code Example B-5 Partial JSP with `obtainParentContainer` Tag

```

<desktop:obtainContainer container="$JSPProvider">
  <!-- make sure we can use the obtainChannelsFromContainer ->
  <desktopcpc:containerProviderContext>
    <desktopcpc:obtainChannelFromContainer channel="myChannel">

      <!-- tags to operate on channel "myChannel" within the container stored as attribute
      "JSPProvider" in page context -->

      <desktopcpc:obtainParentContainer>
        <!-- tags to operate on the container stored as attribute "JSPProvider" in page
        context -->
      </desktopcpc:obtainParentContainer>

      <!-- more tags to operate on channel "myChannel" within the container stored as
      attribute "JSPProvider" in page context -->

      </desktopcpc:obtainChannelFromContainer>
    </desktopcpc:containerProviderContext>
  </desktop:obtainContainer>

```

NOTE The current design does not allow multiple levels of containment hierarchy in a single JSP, thus the tag library does not support deeper nesting of the obtain tags.

Example Tab Page with Selected Channels

The following example shows how the `obtainContainer`, `obtainChannelsFromContainer`, and `obtainParentContainer` tags can be used in a JSP to produce a tab page with selected channels.

```

<desktop:obtainContainer container="$JSPProvider">
  <!-- get the selected channel list and store it in attribute selectedChannel
  -->
  <desktop:getSelectedChannels id="selectedChannel"/>
  <!-- make sure we can use the obtainChannelsFromContainer -->
  <desktopcpc:containerProviderContext>
    <!-- loop through each channel using Jakarta forEach tag. With
    each iteration, the next -->
    <!-- channel's value (name of the channel) will be stored in
    attribute "channel" -->
    <jx:forEach var="channel" items="$selectedChannel">
      <desktopcpc:obtainChannelFromContainer channel="myChannel">
        <!-- print out the title of this channel -->
        <desktop:getTitle/>
        <br>

        <desktopcpc:obtainParentContainer>
          <!-- make sure the container is a tab container be we
          use a tab container tag -->
          <desktopoptab:tabContainerProvider>
            <!-- store the selected tab name in attribute
            selectedTabName -->
            <desktopoptab:getSelectedTabName
            id="selectedTabName"/>
            <!-- declare selectedTabName as String using Jakarta
            tag, so the value can be used in the jsp -->
            <jx:declare id="selectedTabName"
            type="java.lang.String"/>
            <desktop:getName/> container has the tab
            <%=selectedTabName%> selected.
          </desktopoptab:tabContainerProvider>
          </desktopcpc:obtainParentContainer>
          </desktopcpc:obtainChannelFromContainer>
        </jx:forEach>
      </desktopcpc:containerProviderContext>
    </desktop:obtainContainer>
  
```

The `desktop:getSelectedChannels id="selectedChannel"/` tag provides an example of an empty bodycontent property. The `jx:forEach var="channel" items="$selectedChannel"` tag is a function from one of the JSP Standard tag libraries included with the Sun ONE Portal Server software.

Tag Reference

This section describes the tag attributes and exceptions.

Attributes and Return Values

You can pass two kinds of attributes to the Desktop library tags:

- **String** - The simplest way to pass in an attribute is to specify it as a string, for example:

```
<dt:sometag attribute1="myAttribute"/>
```

The string `myAttribute` becomes the value of `attribute1`. If a tag expects an integer attribute, use the following:

```
<dt:sometag attribute1="12345"/>
```

The tag uses the corresponding Java classes (`java.lang.Integer`) to translate the string to an integer value. The same applies to all the primitive Java types (boolean, int, and so on).

- **Reference** - Sometimes you cannot pass in an attribute as a string. For example, it is impossible to specify a provider object as a string. In this case, the attribute needs to be passed in as a reference defined in the `pageContext`. You do so by concatenating the character `$` with the name of the object stored in the `pageContext`, for example:

```
<dt:sometag attribute1="$myAttribute"/>
```

In this example, the value of `attribute1` is whatever object is stored in the `pageContext`. The method `pageContext.findAttribute()` is used, not `getAttribute()`. So it is possible to define some value in the request object and pass this object in as a reference.

NOTE This mechanism is the main form of communication between tags in the Desktop tag library as well as other tags in other tag libraries (for example, jsptl).

Tags with attributes `id` and `scope` (even if they are optional) are expected to return a value as a result of using the tags. Values can be “returned” in two ways:

- **Print to the JspWriter stream** - If the attribute `id` is not given when using a tag, the tag prints the value to the JspWriter stream. This results in the value showing up in the HTML code generated by the JSPs. In this case, the attribute scope is ignored.
- **Store the value in pageContext** - If you specify the attribute `id` when using a tag, the tag stores the value to the pageContext as an attribute with a name specified by the `id` attribute. This operation overwrites whatever original value the attribute has. In addition to the `id`, you can also define the attribute scope. It can have one of the following values: `page`, `request`, `session`, or `application`. These values correspond to the `PAGE_SCOPE`, `REQUEST_SCOPE`, `SESSION_SCOPE`, and `APPLICATION_SCOPE` defined in `javax.servlet.jsp.PageContext`. This specifies the scope in which to save the value. If no or unrecognized scope is given, the default is to save it in page scope.

Tag Library Exceptions

The tag library provides five types of exceptions:

1. **Invalid Tag Sequence** - Occurs when obtain tags are nested in an invalid sequence. See [“Using the Desktop Tag Library in Your Application” on page 433](#) for the sequencing of the obtain tags. The name of the first tag that violates the sequence is provided.
2. **Invalid Provider Type** - Occurs when a validation test is not passed. That is, the TLD does not support the current provider in the context. The name of the validator tag that failed is provided.
3. **Invalid Parameter** - Occurs when an attribute passed in with the tag is not a legal parameter for the tag. The name of the attribute that causes the problem is provided.
4. **Undefined Parameter** - Occurs when an attribute is passed in as reference, but the attribute is not defined in the page context. The name of the attribute is provided.
5. **Empty Context** - Occurs when there is no container or provider in the context to operate on. Most likely, the appropriate obtain tag is missing so the context is not set up properly.

Tag Overview

Essentially, the Desktop tag library is a wrapper of the PAPI, the ProviderContext, and ContainerProviderContext interfaces, and the specific table containers in Sun ONE Portal Server software. The tags in the Desktop tag library fall into three basic groups: context setup tags, validator tags, and normal tags.

Context Setup Tags

These tags, which start with the prefix `obtain`, set up the context (storing the container or provider in question into the `pageContext`). Whatever tag operation that happens within these tags is done on the provider that is set in the context.

[Table B-1](#) describes the context setup tags in the tag libraries. The table has four columns: the first column lists the tag name, the second describes what the tag does, the third column gives what TLD file the tag is in, and the fourth lists that tag's attributes with brief comments.

All the context setup tags contain a value of `JSP` for the `bodycontent` tag.

Table B-1 Context Setup Tags

Tag Name	Description	in TLD File	Attributes/Descriptions
<code>obtainChannel</code>	Gets channel object.	<code>desktop.tld</code>	<code>channel</code> (required) - the name of the channel
<code>obtainContainer</code>	Gets container object.	<code>desktop.tld</code>	<code>container</code> (required) - the name of the container
<code>obtainParentContainer</code>	Gets the parent container name.	<code>desktopContainerProviderContext.tld</code>	none
<code>obtainChannelFromContainer</code>	Gets the channel name.	<code>desktopContainerProviderContext.tld</code>	<code>channel</code> (required) - the name of the channel
<code>obtainSelectedChannel</code>	Gets the selected channel name.	<code>desktopSingle.tld</code>	none
<code>obtainTab</code>	Gets the tab object.	<code>desktopTab.tld</code>	<code>tab</code> (required) - the name of the tab
<code>obtainTabByName</code>	Gets the tab name.	<code>desktopTab.tld</code>	<code>name</code> (required) - the name of the tab

Validator Tags

These tags validate that the provider in context can legally use the tags in the TLD. Each TLD, with the exception of `desktop.tld`, has a validator tag defined (usually with the name of the TLD file). If the provider in context cannot use the tags defined in the TLD, an exception is thrown that is displayed on the screen and processing stops.

Users should surround tags that belong to a specific TLD with the respective validator tag. However, it is not possible to enforce that in a JSP environment. To make it easier for users to “guess” which TLD they can use or to debug the JSPs, `getProviderClassName()` and `getContainerClassName()` tags are provided in the `desktop.tld`. They return the class name of the container or provider in the context.

The following is a two column table: column one lists the tag name; column two provides a brief description.

<code>providerContext</code>	Validates that the provider in the context can legally use the tags in <code>desktopproviderContext.tld</code> .
<code>containerProviderContext</code>	Validates that the provider in the context can legally use the tags in <code>desktopcontainerProviderContext.tld</code> .
<code>singleContainerProvider</code>	Validates that the provider in the context can legally use the tags in <code>desktopSingle.tld</code> .
<code>obtainSelectedChannelFromRequest</code>	Gets the channel name from request. This tag is in <code>desktopSingle.tld</code> .
<code>tableContainerProvider</code>	Validates that the provider in the context can legally use the tags in <code>desktopTable.tld</code> .
<code>tabContainerProvider</code>	Validates that the provider in the context can legally use the tags in <code>desktopTab.tld</code> .

Normal Tags

These tags serve as wrappers of PAPI, `ProviderContext`, `ContainerProviderContext`, and the specific containers in Sun ONE Portal Server.

For the remaining part of this chapter, unless specified, “tag” refers to the normal tag in the Desktop tag library.

Table B-2 through Table B-7 on page 450 describe the normal tags in the tag libraries. Each table has three columns: the first column lists the tag name, the second a description of the tag's function, and the third column lists that tag's attributes with brief comments.

The tables are:

- [Table B-2](#) `desktop.tld`
- [Table B-3](#) `desktopProviderContext.tld`
- [Table B-4](#) `desktopContainerProviderContext.tld`
- [Table B-5](#) `desktopTab.tld`
- [Table B-6](#) `desktopTable.tld`
- [Table B-7](#) `desktopTheme.tld`
- [Search Tags](#) `search.tld`

NOTE The `desktopSingle.tld` tag library has no normal tags.

All of the normal tags in the `desktop.tld` file have a bodycontent of empty.

Table B-2 `desktop.tld` Normal Tags with Attributes

Tag Name	Description	Attributes/Descriptions
<code>getProviderClassName</code>	Returns the class name of the provider that backs the channel.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getContent</code>	Returns a string buffer with the contents of the provider's object's default view. This method is called by the clients of the provider object to request the provider's default view. This method may return null if the provider does not implement a default view. In this case, the provider should return <code>false</code> from its <code>isPresentable()</code> method.	none
<code>getTitle</code>	Returns a string with the title of the channel.	<code>id</code> (optional) <code>scope</code> (optional) <code>silentException</code> (optional)

Table B-2 desktop.tld Normal Tags with Attributes *(Continued)*

Tag Name	Description	Attributes/Descriptions
getDescription	Returns a string with the description for the channel.	id (optional) scope (optional) silentException (optional)
getEdit	Returns a string buffer with the provider's Edit page.	id (optional) scope (optional)
getHelp	Returns the help URL for this provider. The returned help URL can be either fully qualified URL string (<code>http://server:port/portal/docs/en/desktop/usedesk.htm</code>) or a relative path (<code>desktop/usedesk.htm</code>). When it is a relative path, the Desktop software resolves it to the full URL.	id (optional) scope (optional) silentException (optional)
getName	Returns a string with the name of the provider, which must match the name of the provider the channel was initialized with.	id (optional) scope (optional)
getRefreshTime	Returns a long with the refresh time for this provider in seconds. Use this value to determine if you should fetch a fresh default view for the provider. If the return value from this method is X, you may choose not to fetch fresh content (and use a cached copy instead) if less than X seconds has elapsed since the last time the content was refreshed. If provider content is expected to change infrequently, this method can return some value so that the provider's content is not fetched every time the front page is drawn, thereby saving significant processing time.	id (optional) scope (optional)
getWidth	Returns an integer with the suggested width for the channel to the container of the channel as to how much screen real estate it requires. The values correspond to thick, thin, full top and full bottom.	id (optional) scope (optional) silentException (optional)
getEditType	Returns an integer that defines edit type either EDIT_SUBSET or EDIT_COMPLETE.	id (optional) scope (optional)
isEditable	Returns a Boolean that gives the editable status of the channel. Returns true if the channel is editable; otherwise false.	id (optional) scope (optional)

Table B-2 desktop.tld Normal Tags with Attributes *(Continued)*

Tag Name	Description	Attributes/Descriptions
isPresentable	Returns a Boolean that gives the presentable status for a channel. Returns <code>true</code> if the channel is presentable. Searches for the key HTML with the value <code>true</code> on the client data for the session's client type and returns <code>true</code> . If there is no such key, the method returns <code>true</code> if the session's client type is named HTML. In both cases, the content-type for the session's client type must equal <code>text/html</code> in order for the method to return <code>true</code> .	id (optional) scope (optional)
processEdit	Performs the provider's Edit page processing. Processes a form for this provider. This method is called to process form data associated with the provider. Typically, this method is called to process the Edit page generated from the <code>getEdit()</code> method. Usually, the client calling this method on a provider object is the desktop servlet. Form data that is passed into this method in the request has been decoded into Unicode.	id (optional) scope (optional)
getContainerClassName	Returns the class name of the container that backs the container.	id (optional) scope (optional)
getSelectedChannels	Returns a list of selected channel names. The list returned is a Collection of Strings. Each of the Strings is the name of a channel that has been selected.	id (required) scope (optional)
getAvailableChannels	Returns a list of available channel names. The list returned is a Collection of Strings. Each of the Strings is the name of a channel that is available.	id (required) scope (optional)
scontent	Returns the URL of the directory that has the static content (for example, images and style sheet). This utility tag can be used anywhere.	none

All of the normal tags in the `desktopProviderContext.tld` file have a `bodycontent` of empty.

Table B-3 desktopProviderContext.tld Normal Tags with Attributes

Tag Name	Description	Attributes/Descriptions
<code>getStringProperty</code>	Returns a string with the property value. This is an overloaded method that can return alternately the default or localized version of the property value.	<p><code>key</code> (required) - the name of the property</p> <p><code>localized</code> (optional)</p> <p><code>id</code> (optional)</p> <p><code>scope</code> (optional)</p> <p><code>pflist</code> (optional)</p>
<code>getBooleanProperty</code>	Returns a Boolean that gives the value of the property.	<p><code>key</code> (required) - the name of the property</p> <p><code>id</code> (optional)</p> <p><code>scope</code> (optional)</p> <p><code>pflist</code> (optional)</p>
<code>getCollectionProperty</code>	Returns a Java Map with the collection property. Here, a collection refers to a multi-value property. Depending on the context, it is either the analogue of Java Maps or Lists. For Lists, the returned Java Map object contains key-value pairs where the key equals the value. This is an overloaded method that can return alternately the default or localized version of the collection property.	<p><code>key</code> (required) - the name of the property</p> <p><code>id</code> (optional)</p> <p><code>scope</code> (optional)</p> <p><code>pflist</code> (optional)</p>
<code>getIntegerProperty</code>	Returns an integer with the integer property. This method returns a default value if the property does not exist.	<p><code>key</code> (required) - the name of the property</p> <p><code>id</code> (optional)</p> <p><code>scope</code> (optional)</p> <p><code>pflist</code> (optional)</p>
<code>getProperty</code>	Returns a Java Object with a property. The value returned from this method is a Java Object of type String, Integer, Boolean, or Map. If the property does not exist, then the default value is returned.	<p><code>key</code> (required) - the name of the property</p> <p><code>id</code> (optional)</p> <p><code>scope</code> (optional)</p>
<code>setStringProperty</code>	Sets a string property.	<p><code>key</code> (required) - the name of the property</p> <p><code>value</code> (required) - the value of the property to be set</p> <p><code>pflist</code> (optional)</p>

Table B-3 desktopProviderContext.tld Normal Tags with Attributes *(Continued)*

Tag Name	Description	Attributes/Descriptions
setBooleanProperty	Sets a Boolean property.	key (required) - the name of the property value (required) - the value of the property to be set pflist (optional)
setCollectionProperty	Sets a collection property.	key (required) - the name of the property value (required) - the value of the property to be set pflist (optional)
setIntegerProperty	Sets an integer property.	key (required) - the name of the property value (required) - the value of the property to be set pflist (optional)
getLocalePropertiesFilters		id (required) scope (optional)
getClientPropertiesFilters		id (required) scope (optional)
getClientAndLocalePropertiesFilters		id (required) scope (optional)
getClassName	Returns a string with the class name for the provider class that this object is providing an environment for. The class name returned must implement the provider interface. This method is used to construct the provider object. It is used by container channels.	id (optional) scope (optional)
getTemplate	Returns a string buffer with the desktop template. The actual template buffer returned is based on the Desktop type, locale, channel, client type, and the template name.	file (required) - name of template to return table (optional) - hashtable - tag table used for tag swapping id (optional) scope (optional)

Table B-3 desktopProviderContext.tld Normal Tags with Attributes (Continued)

Tag Name	Description	Attributes/Descriptions
getDesktopURL	Returns a string with the Desktop URL. The Desktop URL is the absolute URL used to access the Desktop application. For example: <code>http://server:port/portal/dt</code> . The request object parameter is included to facilitate implementations. It may be used to build the Desktop URL by supplying the server, port, and protocol of the request. It is not required that the request object be utilized to generate the Desktop URL.	querymap (optional) querystring (optional) pathinfo (optional) escape (optional) id (optional) scope (optional)
getDesktopType	Returns a string with the Desktop type. The Desktop type, also known as template type, is a string that is one of several indexes used to lookup Desktop templates and JSP files. The Desktop type is typically used to group Desktop customization files to provide different themes.	id (optional) scope (optional)
getLocaleString	Returns a string representation of the locale.	id (optional) scope (optional)
getLocale	Returns Java Locale object representation of the locale.	id (optional) scope (optional)
getLogoutURL	Returns a string with the logout URL. The result of making a connection to the logout URL is typically the termination of the user's session. What actually happens is dependent on the application receiving the URL connection. Providers may use this value to generate links that allow the user to end their session.	id (optional) scope (optional)
getStringAttribute	Returns a string with the value of the string attribute or null if the attribute is not found. Attributes are settings that are not channel-specific. An example of an attribute might be the user's first and last name. Channel-specific settings are called properties. Properties can be retrieved by calling the <code>get*Property()</code> methods. Whether a particular value is considered a property or an attribute depends on the underlying implementation of <code>ProviderContext</code> .	key (required) - the name of the attribute id (optional) scope (optional)
setStringAttribute	Sets a string attribute. Attributes are settings that are not channel-specific. An example of an attribute might be the user's first and last name. Channel-specific settings are called properties. Properties can be set by calling the <code>set*Property()</code> methods. Whether a particular value is considered a property or an attribute depends on the underlying implementation of <code>ProviderContext</code> .	key (required)- the name of the attribute value (required)- the value of the attribute

Table B-3 desktopProviderContext.tld Normal Tags with Attributes *(Continued)*

Tag Name	Description	Attributes/Descriptions
getClientTypeProperty		key (required) clientType (optional) id (optional) scope (optional)
getClientType	Returns a string with the client type. There is no requirement as to how the client type is determined. It may be hardcoded, derived from the session, or otherwise.	id (optional) scope (optional)
getDefaultClientType	Returns a string with the default client type.	id (optional) scope (optional)
getCharset	Returns a string with the character set. The character set is used for decoding input and encoding output.	id (optional) scope (optional)
getClientPath	Returns a string with the client path. The client path is one of several components used to lookup Desktop templates and JSPs. This allows the lookup to be client-specific.	id (optional) scope (optional)
getContentType	Returns a string with the content type. This value is used to determine if a provider is able to produce content for the client's device.	id (optional) scope (optional)
getSessionID	Returns a string with the unique session identifier. The format of the return value is implementation specific. The only guarantee is that it is unique (each user session has a unique session ID).	id (optional) scope (optional)
getUserID	Returns a string with the user identifier. The format of the return value is implementation specific. There is no guarantee that this value is unique (there may be multiple user sessions for a given user identifier).	id (optional) scope (optional)
setClientProperty	Sets a client property.	name (required) - the name of the property value (required) - the value of the property to be set
getClientProperty	Returns a string with the client property.	name (required) id (optional) scope (optional)
isLogMessageEnabled	Returns a Boolean; true if the log level is set to message or higher; otherwise false.	id (optional) scope (optional)

Table B-3 desktopProviderContext.tld Normal Tags with Attributes *(Continued)*

Tag Name	Description	Attributes/Descriptions
isLogWarningEnabled	Returns a Boolean; true if the log level is set to warning or higher; otherwise false.	id (optional) scope (optional)
logError	Logs a message (any Java Object) if the logging level is error. The location to store logging messages is implementation dependent.	value (required) - message to log throwable (optional)
logMessage	Logs a message (any Java Object) if the logging level is message or higher. The location to store logging messages is implementation dependent.	value (required) - message to log throwable (optional)
logWarning	Logs a message (any Java Object) if the logging level is warning or higher. The location to store logging messages is implementation dependent.	value (required) - message to log throwable (optional)
getDefaultChannelName	Returns a string with the default channel name.	id (optional) scope (optional)
getTopChannelName	Returns the top channel name for the current request.	id (optional) scope (optional)
getStaticContentPath	Gets the URI prefix to web server static content.	id (optional) scope (optional)
getProviderVersion	Get the version of the provider schema for the current channel.	id (optional) scope (optional)
encodeURLParameter	URL encodes a unicode string.	id (optional) scope (optional) key (required)
decodeURLParameter	Decodes the URL encoded Unicode string. This tag just returns back the original string passed in.	id (optional) scope (optional) key (required)
encodeURL	Encodes a URL. Rewrites the URL to include the session id.	url (required) id (optional) scope (optional)
escape	Escapes a String using an encoder class that encodes a specific type of markup. This tag is used to allow provider code to encode content in a device-unaware manner.	id (optional) scope (optional) unescaped (required)

The normal tag in the `desktopContainerProviderContext.tld` file has a bodycontent of empty.

Table B-4 `desktopContainerProviderContext.tld` Normal Tags with Attributes

Tag Name	Description	Attributes/Descriptions
<code>getContent</code>	Returns a string buffer with the content of the named channel. This method is provided for convenience. It gets the provider object for the named channel and calls <code>Provider.getContent()</code> .	<code>channel</code> (required) - the name of the channel <code>id</code> (optional) <code>scope</code> (optional)

All of the normal tags in the `desktopTab.tld` file have a bodycontent of empty.

Table B-5 `desktopTab.tld` Normal Tags with Attributes

Tag Name	Description	Attributes/Descriptions
<code>getAvailableTabs</code>	Returns a list of available tabs. The list returned is a Collection of Strings. Each of the Strings is the name of an Unmodifiable Tab that is available.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getSelectedTabs</code>	Returns the list of selected tabs. The list returned is a Collection of Strings. Each of the Strings is the name of an Unmodifiable Tab that is selected.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getSelectedTab</code>	Returns the selected tab, the current selected Unmodifiable Tab in the user's session.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getMakeTab</code>	Returns the make tab, the tab spec to be used for "Make My Own tab" creation by the user.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getStartTabName</code>	Returns a string with the start tab Name, the name of the tab to be displayed when the user logs in.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getSelectedTabName</code>	Returns a string with the selected tab Name, the current selected tab in the user's session.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getTabURL</code>	Returns the Tab URL. This method gets the tab URL used to switch the selected tab on the user's desktop.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getName</code>	Returns a string with the name of the tab.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getDesc</code>	Returns a string with the description of the tab.	<code>id</code> (optional) <code>scope</code> (optional)

Table B-5 desktopTab.tld Normal Tags with Attributes (Continued)

Tag Name	Description	Attributes/Descriptions
getDisplayname	Returns a string with the display name of the tab.	id (optional) scope (optional)
getEncodedName	Returns a string with the HTML encoded name of the tab.	id (optional) scope (optional)
isPredefined	Determines whether the tab is predefined or not and returns a boolean that gives the predefined status of a tab.	id (optional) scope (optional)
isRemovable	Returns a Boolean that gives the removable status of the tab. Returns <code>true</code> if the tab is removable; otherwise <code>false</code> .	id (optional) scope (optional)
isRenamable	Returns a Boolean that gives the renamable status of the tab. Returns <code>true</code> if the tab is renamable; otherwise <code>false</code> .	id (optional) scope (optional)

All of the normal tags in the `desktopTable.tld` file have a `bodycontent` of empty.

Table B-6 desktopTable.tld Normal Tags with Attributes

Tag Name	Description	Attributes/Descriptions
getColumns	Returns a list of channel names that belong in that particular column.	column (required) id (required)
getColumnWidth	Returns the width of a column (a percentage with respect to the entire Desktop). Valid columns are left, center, and right.	column (required) id (optional)
getHasFrame	Returns a Boolean that gives the frame status of the channel. Returns <code>true</code> if the channel has a frame; otherwise <code>false</code> .	id (optional) scope (optional)
getIsMinimized	Returns a Boolean that gives the minimized status of the channel. Returns <code>true</code> if the channel is minimized; otherwise <code>false</code> .	id (optional) scope (optional)
getIsMovable	Determines whether the channel is movable or not and returns a Boolean that gives the movable status of a channel.	id (optional) scope (optional)

Table B-6 desktopTable.tld Normal Tags with Attributes *(Continued)*

Tag Name	Description	Attributes/Descriptions
getProviderCommand	Gets the HTML code needed to display the provider commands (minimize channel, help screen, edit channel, and so forth). The commands are put in a Map. The keys for the Map are <code>minMaximizedCommand</code> , <code>helpCommand</code> , <code>editCommand</code> , <code>detachAttachCommand</code> , and <code>removeCommand</code> .	id (optional) scope (optional)
getIsDetached	Returns a Boolean that gives the detached status of the channel. Returns <code>true</code> if the channel is detached; otherwise <code>false</code> .	id (optional) scope (optional)
getDetached	Returns the detached channels list. The list returned is a Collection of Strings. Each of the Strings is the name of a channel. The channels returned are not necessary channels that have been detached from the desktop. The tag <code>getIsDetached</code> should be used to verify that a channel has been detached.	id (required) scope (optional)
getWindowName	Returns a string with the window name for the detached window when a channel is detached.	id (optional) scope (optional)
isPopup	Determines whether the page is being drawn for a popup channel. Returns <code>true</code> if the table container action is a popup and returns <code>false</code> otherwise.	id (optional) scope (optional)
getPopupWindowWidth	Returns an integer with the popup window width for the detached window when a channel is detached.	id (optional) scope (optional)
getPopupWindowHeight	Returns an integer with the popup window height for the detached window when a channel is detached.	id (optional) scope (optional)
getContents	Returns a string buffer with the contents of all non-minimized and selected channels of the table container. The contents are put in a Map with the channel name as the key.	id (required) scope (optional)

All of the normal tags in the `desktopTheme.tld` file have a `bodycontent` of empty.

Table B-7 desktopTheme.tld Normal Tags with Attributes

Tag Name	Description	Attributes/Descriptions
getGlobalThemes	Returns the list of globally defined themes. The list returned is a Collection of Strings. Each of the Strings is the name of a globally defined theme.	id (required) scope (optional)

Table B-7 desktopTheme.tld Normal Tags with Attributes *(Continued)*

Tag Name	Description	Attributes/Descriptions
getSelectedName	Returns the name of the selected theme. The name returned can be the name of one of the globally defined themes or CustomTheme. If CustomTheme is returned, this means the user has defined and is using the custom defined theme.	id (optional) scope (optional)
setSelectedName	Sets a theme for the current user. The theme to be set can be one of the globally defined themes or a CustomTheme.	value (required) - the name of the theme set
getAttribute	Returns the value of a theme attribute. Get a specific attribute value of a specific theme.	name (required) - the name of the attribute. Possible values are: bgColor, borderColor, titleBarColor, fontColor, borderWidth and fontFace theme (optional) - the name of the theme. If this is not specified, the currently selected theme is used. requestOverride (optional) true or false - Whether to use value in the request to override the theme value. If this is not specified, false is assumed. This is useful in the preview case. id (optional) scope (optional) default - the default value. If getAttribute() of a theme returns null, and a default value is defined in the tag, then return the default value.
setCustomAttribute	Sets a customized value in the CustomTheme	name (required) - the name of the attribute. Possible values are: bgColor, borderColor, titleBarColor, fontColor, borderWidth and fontFace value (required) - the value to be set

Search Tags

The Search tag library contains tag wrappers for the SearchContext Java API. SearchContext is an extension of the Search API with convenient methods for advanced search and search result status. The Search tag library can be divided into various categories depending upon where the tags should be used.

Table [Table B-8 on page 453](#) through [Table B-12 on page 457](#) describe the normal tags in the Search tag library. Each table has three columns: the first column lists the tag name, the second a description of the tag's function, and the third column lists that tag's attributes with brief comments. The tables are:

- [Table B-8](#) searchContext tag
- [Table B-9](#) tags used before a search
- [Table B-10](#) executeSearch tag
- [Table B-11](#) tags related to Search results
- [Table B-12](#) miscellaneous tags

The `search.tld` does not have any context setup or validator tags. All of the normal tags in the `search.tld` file have a bodycontent of empty except `searchContext` and `SOIF`.

A number of attributes have a value of true for `rtexprvalue`, which means the value for this attribute can be obtained at run time or it can be hardcoded. These attributes are listed as dynamic attributes.

Table B-8 searchContext Tag

Tag Name	Description	Attributes/Descriptions
searchContext	Main outer tag that encloses all other tags.	bodycontent - JSP rdmServer (optional) - search server - http://.../portal/search or https://.../portal/search rdmType (optional) - rd-request (default), taxonomy-request, schema-request, server-request or status-request ql (optional) - query language query (optional) - dynamic attribute

The tags in [Table B-9](#) are normally used before executing a search. Before a successful search can be executed these tags must set a value:

- setRDMServer
- setRDMDType
- setQuery **OR** setCriteria

Table B-9 Pre-Search Tags in search.tld

Tag Name	Description	Attributes/Descriptions
setRDMServer	Sets the RDMServer variable. The server URL should be set explicitly. Format: http:// or https:// <i>hostname:port</i> /portal/search Value has to be set.	rdmServer (required) - server URL; can be an expression or a hardcoded string value.

Table B-9 Pre-Search Tags in `search.tld` (Continued)

Tag Name	Description	Attributes/Descriptions
<code>setRDMDType</code>	<p>Sets the RDM Request type. RDMDType - Can be one of:</p> <p><code>rd-request</code>: The default request type. Resource descriptions (documents).</p> <p><code>taxonomy-request</code>: Taxonomy.</p> <p><code>schema-request</code>: The schema.</p> <p><code>server-request</code>: Server information.</p> <p><code>status-request</code>: Server status information.</p> <p>Set value explicitly; the default set by the system may not return the expected results.</p>	<code>rdmType</code> (required) - string
<code>setViewAttributes</code>	<p>Sets the SOIF attributes that are returned for the search. This is an optional tag. It assumes the default values if not set explicitly.</p>	<code>viewAttributes</code> (required) - string: null (all) or comma delimited list of attributes.
<code>setSessionID</code>	<p>The Search Server needs to validate the user's identity for document level security. This tag is a wrapper for <code>setSessionID(String)</code> method in <code>SearchContext</code>. The JSP gets the portal access Token string and passes it along to the search server using this tag.</p>	<code>sessionID</code> (required) - string
<code>setViewHits</code>	<p>Sets the maximum number of hits returned.</p>	<code>viewHits</code> (required) - integer - dynamic attribute
<code>setViewOrder</code>	<p>Sets the sorting order for results.</p>	<code>viewOrder</code> (required) - string: null or comma delimited list of attributes each preceded with + for ascending order or - for descending order.
<code>setCategory</code>	<p>Sets the category id. It is mainly required for browsing and category searches. The category is appended to the query string based on the RDMDType and query language in the <code>executeSearch</code> tag.</p>	<code>category</code> (required) - string: current category level; if not set, the root of the taxonomy tree is used and the search is executed for all categories.
<code>setFirstHit</code>	<p><code>setFirstHit</code> takes a string input. Sets the starting hit for search results. In other words, start from 1, start from 11 and so forth. It corresponds to the <code>setFirstHit()</code> method in the <code>Search</code> API. Results are returned from this hit. This is an optional tag. Alternately you can use <code>setPage</code> and <code>setViewHits</code>.</p>	<code>fromHit</code> (required) - integer

Table B-9 Pre-Search Tags in `search.tld` (Continued)

Tag Name	Description	Attributes/Descriptions
<code>setQuery</code>	Query string. Either this value has to be set or else <code>setCriteria</code> has to be set.	<code>query</code> (required) - string - dynamic attribute
<code>setDatabase</code>	Has a string attribute. Default value is <code>'</code> , which means use <code>berkeley db</code> . Database should be specified if <code>berkeley db</code> should not be used.	<code>database</code> (required) - string - dynamic attribute
<code>setPage</code>	Sets the current page value. The page value is used to calculate the <code>firstHit</code> and <code>totalPages</code> .	<code>page</code> (required) - integer - dynamic attribute
<code>setSearchAll</code>	Set <code>searchAllCategories</code> or not. If the search is within a particular category then set it to <code>false</code> ; else set it to <code>true</code> . Default is <code>true</code> .	<code>searchAll</code> (required) - Boolean
<code>setQueryLanguage</code>	Sets the query language. <code>ql</code> - Can be one of: <code>search</code> : The default Search query language. Searches documents or the taxonomy. <code>taxonomy-basic</code> : Used for requesting branches or parts of the taxonomy. <code>schema-basic</code> : Queries the Compass schema. <code>url</code> : Retrieves RDs by URL (<code>scope=url</code>).	<code>ql</code> (required) - string
<code>setCriteria</code>	Sets the query string in a list format. Useful in advanced search. The list should have an operand, operator and a value. The list of valid operator's are defined in the <code>SearchContext</code> API. The operand can be a schema field. The user can bypass this tag and just use the <code>setQuery</code> tag directly by converting a complex query into the syntax that the search engine requires. The <code>setCriteria</code> tag is a wrapper for the <code>setScope(list)</code> method in the <code>searchContext</code> API. This method basically parses the list and converts it into a string. For example, <code>author CONTAINS xyz</code> . This produces a query. Either this value has to be set or else <code>setQuery</code> has to be set.	<code>criteria</code> (required) - string - dynamic attribute

The `executeSearch` tag in [Table B-10 on page 456](#) executes the search.

Table B-10 executeSearch Tag

Tag Name	Description	Attributes/Descriptions
executeSearch	Tag for the execute method in SearchContext API. Executes search after doing some validation of the search parameters and query string.	none

The tags in [Table B-11](#) are related to search results and are used after a search is executed. They provide various counts and help display the search results.

Table B-11 Post Search Tags in search.tld

Tag Name	Description	Attributes/Descriptions
SOIF	Wrapper for SOIF API.	bodycontent - JSP input (required) -SOIF document
getSoifResult	Returns SOIF type object.	id (optional) scope (optional)
getValue	Returns a string value of the attribute or returns a string value of a multivalue attribute with index. Wrapper for SOIF API. This tag must be used within the SOIF tag.	soifAttribute (required) - string - attribute name escape (optional) id (optional) scope (optional) truncate (optional) - specifies the number of characters to display in the title
getUrl	Returns a string with the SOIF URL. Wrapper for SOIF API. This tag must be used within the SOIF tag.	escape (optional) id (optional)
getHasNextPage	Returns true if there are more hits for the next page by considering the values of viewHits and the current page.	id (optional) scope (optional)
getHasPreviousPage	Returns true if there is a previous page. Value based on viewHits and current page value.	id (optional) scope (optional)
getNoHits	Returns true if no matching hits were found. This is a convenience tag.	id (optional) scope (optional)

Table B-11 Post Search Tags in `search.tld` (Continued)

Tag Name	Description	Attributes/Descriptions
<code>getHitCount</code>	Returns the total number of results that matched the query.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getToHit</code>	Returns the last hit being displayed on a page. The value is based on <code>firstHit</code> and <code>viewHits</code> .	<code>id</code> (optional) <code>scope</code> (optional)
<code>getPage</code>	Returns the current page. If the value is not set, calculates the page based on <code>viewHits</code> and <code>firstHit</code> .	<code>id</code> (optional) <code>scope</code> (optional)
<code>getTotalDocuments</code>	Returns the total number of documents in the database.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getTotalPages</code>	Returns the total number of pages of hits that are available. The value is calculated from <code>viewHits</code> and <code>hitCount</code> .	<code>id</code> (optional) <code>scope</code> (optional)
<code>getResultCount</code>	Returns the number of results returned by the search. Returns -1 for an error.	<code>id</code> (optional) <code>scope</code> (optional)

The tags in [Table B-12](#) are used for debugging.

Table B-12 Miscellaneous Tags in `search.tld`

Tag Name	Description	Attributes
<code>getSearchString</code>	Returns a string with a the search query being executed. Good for debugging.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getCategory</code>	Returns a string with the current category name or else set to root.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getFirstHit</code>	Returns the starting hit being displayed.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getSessionID</code>	Returns a string with a user's session id. It has no value unless previously set.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getQuery</code>	Returns the query string; returns an empty string if not set.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getViewHits</code>	Returns <code>viewHits</code> , an integer that defines the maximum number of hits returned (range 0-100). The default value is 8 if not set or set outside of range.	<code>id</code> (optional) <code>scope</code> (optional)

Search Exceptions

When you are developing JSPs or doing Search administration tasks, you may see the following exceptions:

Taglib Use Errors:

- Invalid Tag Sequence: Must be within a searchContext Tag
- Variable Undefined in Page Context
- Invalid nesting of context
- Error in creating search context

Search Request Errors:

- Search server is not defined
- View Hits cannot be 0
- RDM Type must be defined
- Query Language must be defined

TIP When debugging, if you receive unusual exceptions, check that the search server is set and its format is correct and has no typographical errors.

Example of a Search JSP

Code Example B-6 Sample Search JSP

```
<%--
  Copyright 2002 Sun Microsystems, Inc. All rights reserved.
  PROPRIETARY/CONFIDENTIAL. Use of this product is subject to license terms.
--%>

<!-- @(#)searchContent.jsp -->

<%@page language="java" import="com.sun.portal.search.providers.*, java.lang.*,
java.util.*, java.text.*" %>
<%@page errorPage="error.jsp" %>

<%@ taglib uri="/tld/search.tld" prefix="search" %>
<%@ taglib uri="/tld/desktop.tld" prefix="dt" %>
<%@ taglib uri="/tld/desktopProviderContext.tld" prefix="dtpc" %>

<B>Test 1: </B><BR>
```

Code Example B-6 Sample Search JSP (Continued)

```

<dt:obtainChannel channel="$JSPPProvider">
<dtpc:providerContext>

    <search:searchContext>
    <%--<search:setQueryLanguage ql= "search"/> --%>
    <search:setQuery query= "java"/>
    <%--<search:setRDMDType rdmType= "rd-request"/>--%>
    <search:setRDMServer rdmServer= "http://milongal.sesta.com:80/portal/search"/>
    <search:setViewAttributes viewAttributes=
"url,title,description,score,content-length,classification"/>
    <search:setViewHits viewHits= "10"/>
    <search:setPage page= "1"/>

    <dtpc:getSessionID id = "accessToken"/>
    <search:setSessionID sessionID="$accessToken"/>

    <search:executeSearch/>

    <UL>
    <LI><search:getSearchString/>
    <LI>Query = <search:getQuery/>
    <LI>resultCount = <search:getResultCount/>
    <LI>hitCount = <search:getHitCount/>
    <LI>total Documents = <search:getTotalDocuments/>
    <LI>total Pages = <search:getTotalPages/>
    <LI>hasnoHits= <search:getNoHits/></UL>

    </search:searchContext>
</dtpc:providerContext>
</dt:obtainChannel>

<B>Test 2: </B><BR>
<search:searchContext query="*" rdmType="rd-request" ql="search"
rdmServer="http://milongal.sesta.com:80/portal/search">
    <search:setViewHits viewHits= "10"/>
    <search:executeSearch/>
    <UL><LI>Query = <search:getQuery/>
    <LI>resultCount = <search:getResultCount/></UL>
</search:searchContext>

<B>Test 3: </B><BR>
<search:searchContext query="*" rdmType="rd-request" ql="search"
rdmServer="http://milongal.sesta.com:80/portal/search">
    <search:setFirstHit firstHit= "21"/>
    <search:setViewHits viewHits= "10"/>
    <search:executeSearch/>
    <UL><LI>Query = <search:getQuery/>
    <LI>Page = <search:getPage/>
    <LI>resultCount = <search:getResultCount/></UL>
</search:searchContext>

```

Instant Messaging Tags

The IMProvider content page uses a custom tag library defined in a file called `im.tld` which is installed into the `/etc/opt/SUNWps/desktop/default/tld` directory. The `im.tld` file defines the following tags:

Table B-13 Tags in `im.tld`

Tag Name	Description	Attributes
<code>getContactGroups</code>	Returns list of contact groups that the user has defined. The list is returned as a Collection of Strings where each String is the display name for a contact group.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getContactGroup</code>	Returns the list of contacts in the named contact group. The list is returned as a Collection of internal objects that can be passed to the <code>obtainContact</code> tag.	<code>group</code> (required) - The name of the contact group. <code>id</code> (optional) <code>scope</code> (optional)
<code>getUsername</code>	Returns the instant messaging username for the user.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getToken</code>	Returns the login token for the user (either an Sun ONE Identity Server software SSOToken or the user's password.)	<code>id</code> (optional) <code>scope</code> (optional)
<code>obtainContact</code>	A context setup tag that is used to obtain the presence information for the indicated contact. The remaining tags can be used inside this tag.	<code>contact</code> (required) - The internal identifier of the contact, typically obtained from the <code>getContactGroup</code> tag.
<code>getContactPresence</code>	Returns the current presence status for the contact. The status can be: AWAY, BUSY, CLOSED, FORWARDED, IDLE, OPEN, OTHER. These are from the PresenceSession class in the instant messaging API.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getContactName</code>	Return the common name for the contact.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getContactUsername</code>	Returns the instant messaging user name for the contact.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getDateTime</code>	Returns the update time.	<code>format</code> (required) <code>id</code> (optional) <code>scope</code> - (optional)
<code>isSecureMode</code>	Returns boolean indicating whether the channel is being accessed via the Secure Remote Access Gateway component and the Netlet is loaded.	<code>id</code> (optional) <code>scope</code> (optional)

Table B-13 Tags in `im.tld` (Continued)

Tag Name	Description	Attributes
<code>getCodebase</code>	Returns the codebase to use to download the applet. This takes into account whether the channel is being accessed via the Secure Remote Access Gateway component and the Netlet is loaded.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getIMServer</code>	Returns the <code>-server</code> argument to pass to the IM client. This takes into account whether the channel is being accessed via the Secure Remote Access Gateway component and the Netlet is loaded.	<code>id</code> (optional) <code>scope</code> (optional)

The entire interface to the Instant Messaging server APIs is in the `getContactGroup` tag. This tag will fetch all of the presence information and cache it in the request. The remaining tags will simply fetch the information out of the cache.

JavaServer Pages Reference

This appendix describes the various JavaServer Pages™ (JSP™) used by Sun™ ONE Portal Server software.

This appendix contains these sections:

- [Anonymous Desktop JSPs](#)
- [Desktop JSPs in the default Directory](#)
- [JSPs in the sampleportal Directory](#)

Before using this appendix, you should understand the order in which Sun ONE Portal Server software looks for JSPs. Some of the JSPs that are used by a particular channel are actually located in the directory named for the provider instead of the channel. In other cases, the JSPs are located in the `/etc/opt/SUNWps/desktop/default` directory. Also, if the Desktop type, for example, is `sampleportal`, the lookup mechanism searches in the `default` directory if the file is not found in the particular Desktop type subdirectory.

See [“JSP and Template Files Lookup Scenario”](#) on page 38 for more information.

Anonymous Desktop JSPs

Anonymous Desktop JSPs are located under the `/etc/opt/SUNWps/desktop/anonymous/*` directory. This section includes:

- [FrameTabContainer JSPs](#)
- [JSPDynamicSingleContainer JSPs](#)
- [JSPTabContainer JSPs](#)
- [JSPTableContainer JSPs](#)

- [PredefinedFrontPageFramePanelContainerProvider JSPs](#)
- [PredefinedFrontPageTabPagePanelContainerProvider JSPs](#)
- [PredefinedSamplesFramePanelContainerProvider JSPs](#)
- [PredefinedSamplesTabPagePanelContainerProvider JSPs](#)

FrameTabContainer JSPs

Anonymous Desktop JSPs for the FrameTabContainer are located in the `/etc/opt/SUNWps/desktop/anonymous/FrameTabContainer` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>frametab.jsp</code>	This JSP is the anonymous version of the <code>frametab.jsp</code> for the frame tab container with the links for customization deactivated.
<code>header.jsp</code>	This JSP is the anonymous version of the <code>header.jsp</code> for the frame tab container with the links for customization deactivated.
<code>menubar.jsp</code>	This JSP is the anonymous version of the <code>menubar.jsp</code> for frame tab container with the links for customization deactivated.

JSPDynamicSingleContainer JSPs

Anonymous Desktop JSPs for the JSPDynamicSingleContainer are located in the `/etc/opt/SUNWps/desktop/anonymous/JSPDynamicSingleContainer` directory.

JSPTabContainer JSPs

Anonymous Desktop JSPs for the JSPTabContainer are located in the `/etc/opt/SUNWps/desktop/anonymous/JSPTabContainer` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>header.jsp</code>	This JSP is the anonymous version of the <code>header.jsp</code> for the tab container with the links for customization deactivated.
<code>menubar.jsp</code>	This JSP is the anonymous version of the <code>menubar.jsp</code> for the tab container with the links for customization deactivated.

JSPTableContainer JSPs

Anonymous Desktop JSPs for the JSPTableContainer are located in the `/etc/opt/SUNWps/desktop/anonymous/JSPTableContainer` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>header.jsp</code>	This JSP is the anonymous version of the <code>header.jsp</code> for the table container with the links for customization deactivated.
<code>menubar.jsp</code>	This JSP is the anonymous version of the <code>menubar.jsp</code> for the table container with the links for customization deactivated.
<code>table.jsp</code>	Displays the table container's content view, which is used by the table container inside a frame tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>tabtable.jsp</code>	Displays the table container's content view, which is used by the table container inside a tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>toptable.jsp</code>	Displays the table container's content view, which is used by the table container that is the top most container in the Desktop. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)

PredefinedFrontPageFramePanelContainerProvider JSPs

Anonymous Desktop JSPs for the `PredefinedFrontPageFramePanelContainerProvider` are located in the `/etc/opt/SUNWps/desktop/anonymous/PredefinedFrontPageFramePanelContainerProvider` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>header.jsp</code>	This JSP is the anonymous version of the <code>header.jsp</code> for the table container with the links for customization deactivated.
<code>menubar.jsp</code>	This JSP is the anonymous version of the <code>menubar.jsp</code> for the table container with the links for customization deactivated.
<code>searchbox.jsp</code>	Displays the search box that are used in the Desktop header area.
<code>table.jsp</code>	Displays the table container's content view, which is used by the table container inside a frame tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>tabtable.jsp</code>	Displays the table container's content view, which is used by the table container inside a tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>toptable.jsp</code>	Displays the table container's content view, which is used by the table container that is the top most container in the Desktop. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)

PredefinedFrontPageTabPanelContainerProvider JSPs

Anonymous Desktop JSPs for the `PredefinedFrontPageTabPanelContainerProvider` are located in the `/etc/opt/SUNWps/desktop/anonymous/PredefinedFrontPageTabPanelContainerProvider` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>header.jsp</code>	This JSP is the anonymous version of the <code>header.jsp</code> for the tab container with the links for customization deactivated.
<code>menubar.jsp</code>	This JSP is the anonymous version of the <code>menubar.jsp</code> for the tab container with the links for customization deactivated.
<code>searchbox.jsp</code>	Displays the search box that are used in the Desktop header area.
<code>table.jsp</code>	Displays the table container's content view, which is used by the table container inside a frame tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>tabtable.jsp</code>	Displays the table container's content view, which is used by the table container inside a tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>toptable.jsp</code>	Displays the table container's content view, which is used by the table container that is the top most container in the Desktop. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)

PredefinedSamplesFramePanelContainerProvider JSPs

Anonymous Desktop JSPs for the `PredefinedSamplesFramePanelContainerProvider` are located in the `/etc/opt/SUNWps/desktop/anonymous/PredefinedSamplesFramePanelContainerProvider` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>header.jsp</code>	This JSP is the anonymous version of the <code>header.jsp</code> for the table container with the links for customization deactivated.
<code>menubar.jsp</code>	This JSP is the anonymous version of the <code>menubar.jsp</code> for the table container with the links for customization deactivated.
<code>searchbox.jsp</code>	Displays the search box that are used in the Desktop header area.
<code>table.jsp</code>	Displays the table container's content view, which is used by the table container inside a frame tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>tabtable.jsp</code>	Displays the table container's content view, which is used by the table container inside a tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>toptable.jsp</code>	Displays the table container's content view, which is used by the table container that is the top most container in the Desktop. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)

PredefinedSamplesTabPanelContainerProvider JSPs

Anonymous Desktop JSPs for the PredefinedSamplesTabPanelContainerProvider are located in the

`/etc/opt/SUNWps/desktop/anonymous/PredefinedSamplesTabPanelContainerProvider` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>header.jsp</code>	This JSP is the anonymous version of the <code>header.jsp</code> for the tab container with the links for customization deactivated.
<code>menubar.jsp</code>	This JSP is the anonymous version of the <code>menubar.jsp</code> for the tab container with the links for customization deactivated.
<code>searchbox.jsp</code>	Displays the search box that are used in the Desktop header area.
<code>table.jsp</code>	Displays the table container's content view, which is used by the table container inside a frame tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>tabtable.jsp</code>	Displays the table container's content view, which is used by the table container inside a tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>toptable.jsp</code>	Displays the table container's content view, which is used by the table container that is the top most container in the Desktop. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)

Desktop JSPs in the default Directory

This section provides a listing with description of the JSPs in the `/etc/opt/SUNWps/desktop/default` directory. It contains:

- [DiscussionLite JSPs, Discussions JSP, and DiscussionsProvider JSPs](#)
- [DummyChannel JSPs](#)
- [IMProvider JSPs](#)
- [JSPContentContainer JSPs](#)
- [JSPDynamicSingleContainer](#)
- [JSPEditContainer JSPs](#)
- [JSPFrameCustomTableContainerProvider JSPs](#)
- [JSPLayoutContainer JSPs](#)
- [JSPProvider JSPs](#)
- [JSPSingleContainerProvider JSPs](#)
- [JSPTabContainerProvider JSPs](#)
- [JSPTabCustomTableContainerProvider JSPs](#)
- [JSPTableContainerProvider JSPs](#)
- [Miscellaneous JSPs](#)
- [SampleSimpleWebService JSPs and SampleSimpleWebServiceConfigurable JSPs](#)
- [Search JSPs and SearchProvider JSPs](#)
- [SimpleWebServiceConfigurableProvider JSPs and SimpleWebServiceProvider JSPs](#)
- [Subscriptions JSPs and SubscriptionsProvider JSPs](#)
- [TabJSPEditContainer JSPs](#)

DiscussionLite JSPs

The DiscussionLite channel JSPs are located in `/etc/opt/SUNWps/desktop/default/DiscussionLite` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>discussionLiteContent.jsp</code>	Content JSP. This JSP gets portal display profile properties and invokes search using <code>query.jsp</code> and most of the user interface is in <code>display.jsp</code> .
<code>display.jsp</code>	Displays results mostly in HTML.
<code>error.jsp</code>	Error page.
<code>query.jsp</code>	Sets search parameters and executes search.

Discussions JSP

The Discussions channel JSPs are located in the `/etc/opt/SUNWps/desktop/default/Discussions` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>declare.jsp</code>	Declares all portal properties.
<code>discussionContent.jsp</code>	Content page. This JSP routes the request to: <ul style="list-style-type: none"> <code>fullDiscussion.jsp</code> if display mode is set to full (or <code>dmode=full</code>) <code>viewDiscussion.jsp</code> if <code>dmode=vl</code> <code>feedback.jsp</code> if <code>dmode=cmt</code> <code>ratingProcess.jsp</code> if <code>dmode=rtg</code>
<code>discussionDoEdit.jsp</code>	Process edit page.
<code>discussionEdit.jsp</code>	Edit page.
<code>error.jsp</code>	Error page.
<code>feedback.jsp</code>	Request is handled by <code>feedback.jsp</code> when the <code>dmode</code> (discussions mode) value is equal to <code>cmt</code>. Routes the request to <code>feedbackForm.jsp</code>.
<code>feedbackDisplay.jsp</code>	Displays the header information above the feedback form.
<code>feedbackForm.jsp</code>	Displays the 'post reply' and 'start a new discussion' form.

<code>feedbackProcess.jsp</code>	Comment submission is handled by this JSP. The JSP retrieves all the input parameters, builds the SOIF and submits the SOIF to the search database. This JSP consists of scriptlets.
<code>fullDiscussion.jsp</code>	Request is handled by <code>fullDiscussion.jsp</code> when <code>dmode</code> is <code>full</code> and routes the request to <code>fullDiscussionDisplay.jsp</code> . Controls the list view of discussions.
<code>fullDiscussionDisplay.jsp</code>	<code>fullDiscussionDisplay.jsp</code> displays the list of results requested by <code>fullDiscussion.jsp</code> .
<code>pageFooter.jsp</code>	Displays pagination on list discussions page.
<code>portal.jsp</code>	General portal Desktop page. Retrieves all portal provider properties.
<code>query.jsp</code>	Executes search. Used by all pages to execute a search.
<code>rating.jsp</code>	Displays the selection menu for ratings. Included in <code>viewDiscussionDisplay.jsp</code> and <code>viewDiscussionHeader.jsp</code> .
<code>ratingProcess.jsp</code>	Request is handled by <code>ratingProcess.jsp</code> when <code>dmode=rtg</code> . Handles rating submission. Consists mostly of scriptlets.
<code>searchUI.jsp</code>	Displays the search box on the list discussions page.
<code>viewDiscussion.jsp</code>	Request is handled by <code>viewDiscussion.jsp</code> when <code>dmode=v1</code> . Controls the View A Discussion subtree page.
<code>viewDiscussionBar.jsp</code>	Displays the separator bar with the filter, threshold, view menus and the search discussion text field.
<code>viewDiscussionDisplay.jsp</code>	Displays the discussion subtree below the separator bar.
<code>viewDiscussionHeader.jsp</code>	Displays the detailed view of the discussion. Displayed above the separator bar.
<code>viewDiscussionNavigation.jsp</code>	Displays the Navigation links shown above and below the discussion header. Navigation links consist of links for 'All Discussions', 'To parent', 'To Discussion', 'Reference', 'Post Reply'.

DiscussionsProvider JSPs

The DiscussionsProvider JSPs are located in `/etc/opt/SUNWps/desktop/default/DiscussionsProvider` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>declare.jsp</code>	Declares all portal properties.
<code>discussionContent.jsp</code>	Content page. This JSP routes the request to <code>full</code> if <code>display mode</code> is set to <code>full</code> (or <code>dmode=full</code>), or if <code>vl</code> is set to <code>viewDiscussion.jsp</code> , or <code>cmt</code> is set to <code>feedback.jsp</code> , or if <code>rtg</code> is set to <code>ratingProcess.jsp</code> .
<code>discussionDoEdit.jsp</code>	Process edit page.
<code>discussionEdit.jsp</code>	Edit page.
<code>error.jsp</code>	Error page.
<code>feedback.jsp</code>	Request is handled by <code>feedback.jsp</code> when the <code>dmode</code> (discussions mode) value is equal to <code>cmt</code> . Routes the request to <code>feedbackForm.jsp</code> .
<code>feedbackDisplay.jsp</code>	Displays feedback.
<code>feedbackForm.jsp</code>	Displays the 'post reply' and 'start a new discussion' form.
<code>feedbackProcess.jsp</code>	Comment submission is handled by this JSP. The JSP retrieves all the input parameters, builds the SOIF and submits the SOIF to the search database. This JSP consists of scriptlets.
<code>fullDiscussion.jsp</code>	Sets search parameters, executes search, and displays results.
<code>fullDiscussionDisplay.jsp</code>	Displays main discussions with description inline based on <code>showDesc</code> property.
<code>pageFooter.jsp</code>	Displays pagination on list discussions page.
<code>portal.jsp</code>	General portal Desktop page. Retrieves all portal provider properties.
<code>query.jsp</code>	Executes search. Used by all pages to execute a search.

<code>rating.jsp</code>	Displays the selection menu for ratings. Included in <code>viewDiscussionDisplay.jsp</code> and <code>viewDiscussionHeader.jsp</code> .
<code>ratingProcess.jsp</code>	Request is handled by <code>ratingProcess.jsp</code> when <code>dmode=rtg</code> . Handles rating submission. Consists mostly of scriptlets.
<code>searchUI.jsp</code>	Displays the search box on the list discussions page.
<code>viewDiscussion.jsp</code>	Request is handled by <code>viewDiscussion.jsp</code> when <code>dmode=v1</code> . Controls the View A Discussion subtree page.
<code>viewDiscussionBar.jsp</code>	Displays the separator bar with the filter, threshold, view menus and the search discussion text field.
<code>viewDiscussionDisplay.jsp</code>	Displays the discussion subtree below the separator bar.
<code>viewDiscussionHeader.jsp</code>	Displays the detailed view of the discussion. Displayed above the separator bar.
<code>viewDiscussionNavigation.jsp</code>	Displays the Navigation links shown above and below the discussion header. Navigation links consist of links for 'All Discussions', 'To parent', 'To Discussion', 'Reference', 'Post Reply'.

DummyChannel JSPs

The DummyChannel JSPs are located in the `/etc/opt/SUNWps/desktop/default/DummyChannel` directory. These JSPs are used in the situation when there is no valid default channel name specified for the Desktop. For example, when a new organization is created, the default channel name will be set to DummyChannel by default. If the system administrator does not specify a valid default channel name, then the DummyChannel will be displayed with a warning message to remind user that the default channel name needs to be configured.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>dummy.jsp</code>	Displays the warning message for the user.
<code>header.jsp</code>	The header for the Dummy Channel.

`menubar.jsp`

The menubar for the Dummy Channel.

IMProvider JSPs

The IMProvider JSPs are located in the `/etc/opt/SUNWps/desktop/default/IMProvider` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>IMArchiveDisplay.jsp</code>	Controls searching through archived instant messaging content.
<code>IMContent.jsp</code>	Controls the content in the channel.
<code>IMEdit.jsp</code>	Controls the content for the edit page of the channel.
<code>invite.jsp</code>	Controls the content in the popup window that is displayed when a user is invited to a conference is an instant messaging client that is already running.
<code>jnlpLaunch.jsp</code>	Controls the messages that are in the Java Web Start window that is used to start the instant messaging client.
<code>pluginLaunch.jsp</code>	Controls the content of the popup window that is used to run the IM client.

JSPContentContainer JSPs

The JSPContentContainer JSPs are located in the `/etc/opt/SUNWps/desktop/default/JSPContentContainer` directory. These JSPs are used for the content view when the Content link is selected in a JSP-based table container.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>contentdoedit.jsp</code>	Processes the result from the Content Edit page.
<code>contentedit.jsp</code>	Displays the content Edit page.

JSPDynamicSingleContainer

The JSPDynamicSingleContainer JSPs are located in the `/etc/opt/SUNWps/desktop/default/JSPDynamicSingleContainer` directory. This container is used by the search form in the header on the Desktop front page.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>dynamicSingle.jsp</code>	Used by the DynamicSingleContainer to display the channel specified in the request parameter.
--------------------------------	---

JSPEditContainer JSPs

JSPEditContainer JSPs are located in the `/etc/opt/SUNWps/desktop/default/JSPEditContainer` directory. These JSPs are used when the Edit icon is selected in a channel title bar inside a JSP- based container. Channels that have the `editType` defined as `EDIT_SUBSET` use these JSPs.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>doedit.jsp</code>	Processes the result from the Edit page.
<code>edit.jsp</code>	Displays the Edit view of a channel. Also provides a wrapper around the actual Edit view for a given channel.

JSPFrameCustomTableContainerProvider JSPs

JSPFrameCustomTableContainerProvider JSPs are located in the `/etc/opt/SUNWps/desktop/default/JSPFrameCustomTableContainerProvider` directory.

JSPFrameCustomTableContainerProvider JSPs are used when the user creates a new page from Scratch in the sections page.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>framecustomtable.jsp</code>	Displays the content for the newly created page (table container) from the Sections page.
<code>launchPopup.jsp</code>	This JSP is used to determine the channels that are in detached mode and invoke the detached windows for these channels.
<code>leafWrapper.jsp</code>	Displays the channel title bar and border.
<code>popup.jsp</code>	This JSP is used to draw the content in the detached window for the channel in the detached mode.
<code>popupMenubar.jsp</code>	This JSP is used to draw the menubar in the detached window for a detached channel.
<code>providerCommands.jsp</code>	Displays the channel command buttons and links, such as Remove, Minimize/Maximize, Edit, and Help.
<code>providerWrapper.jsp</code>	This JSP is used to draw the wrapper containing the attach, edit, and help links around the channel displayed in the detached window.
<code>tablecolumn.jsp</code>	Handles the left, center and right columns of a table. (Dynamically included.)
<code>tablemaximized.jsp</code>	Handles the channel in the maximized state on the Desktop front page. This JSP is used to draw the HTML around the channel displayed in the maximized state.
<code>tabletopbottom.jsp</code>	Handles the top and bottom channels of a table. (Dynamically included.)

JSPLayoutContainer JSPs

JSPLayoutContainer JSPs are located in the `/etc/opt/SUNWps/desktop/default/JSPLayoutContainer` directory. These JSPs are used to display the Layout view when the Layout link is selected in a JSP-based table container.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>layout1.jsp</code>	Displays the thin-wide layout view.
<code>layout2.jsp</code>	Displays the wide-thin layout view.
<code>layout3.jsp</code>	Displays the thin-wide-thin layout view.
<code>layoutdoedit.jsp</code>	Processes the result from the Layout Edit page.
<code>layoutedit.jsp</code>	Displays the Layout Edit page.
<code>selectLayout.jsp</code>	Displays the three layout images and the select radio buttons.

JSPProvider JSPs

JSPProvider JSPs are located in the `/etc/opt/SUNWps/desktop/default/JSPProvider` directory. This directory contains default set of JSPs that are used by the JSP channels.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>samplecontent.jsp</code>	Displays the contents of the JSP channels.
<code>sampledoedit.jsp</code>	Invoked when the user completes processing the Edit page of the JSP channels.
<code>sampleedit.jsp</code>	Invoked when the user clicks the Edit button of the JSP channels.

JSPSingleContainerProvider JSPs

JSPSingleContainerProvider JSPs are located in the `/etc/opt/SUNWps/desktop/default/JSPSingleContainerProvider` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>header.jsp</code>	Displays the product banner that is used by the Single Container page.
-------------------------	--

<code>menubar.jsp</code>	Displays the menubar that has the Home, Theme, Help, and Logout links.
<code>single.jsp</code>	Displays the content for <code>JSPSingleContainerProvider</code> .

JSPTabContainerProvider JSPs

JSPTabContainerProvider JSPs are located in the `/etc/opt/SUNWps/desktop/default/JSPTabContainerProvider` directory. These JSPs are used as the default set of JSPs for a new channel based on `JSPTabContainerProvider`.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>availableTabs.jsp</code>	Displays the tabs and the URLs associated with each tab for activating them on the front page.
<code>header.jsp</code>	Displays the header bar for the Tab Container page. (Dynamically included.)
<code>makeNewTab.jsp</code>	Provides the content for the Make New Tab page of the tab container.
<code>makeTopic.jsp</code>	Provides the content for each of the tab topics in the Make New Tab page.
<code>menubar.jsp</code>	Displays the menubar that has the Home, Tabs, Theme, Help, and Logout links. (Dynamically included.)
<code>remove.jsp</code>	Displays the remove, rename, and start links for each of the selected pages of the JSP tab container in the Current Tab Settings page.
<code>removeRenameTab.jsp</code>	Displays the remove and rename part of the Edit page for the tab container.
<code>selectedTab.jsp</code>	Displays the tab image for the current selected tab in the tab container.
<code>tab.jsp</code>	Is the main JSP for the tab container. It draws the content page for the tab container. (Dynamically includes <code>header.jsp</code> and <code>menubar.jsp</code> .)

<code>tabedit.jsp</code>	Displays the Edit page for the tab container where new pages can be added, removed, or renamed.
<code>tabs.jsp</code>	Displays the available tabs and the links for them to be activated on the Desktop.

JSPTabCustomTableContainerProvider JSPs

JSPTabCustomTableContainerProvider JSPs are located in the `/etc/opt/SUNWps/desktop/default/JSPTabCustomTableContainerProvider` directory. JSPTabCustomTableContainerProvider JSPs are used when the user creates a new tab from scratch in the tabs page.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>launchPopup.jsp</code>	Displays the windows that are detached from the table container. (Dynamically included.)
<code>leafWrapper.jsp</code>	Displays the content view for each channel inside the table container.
<code>popup.jsp</code>	This JSP is used to draw the content in the detached window for the channel in the detached mode.
<code>popupMenubar.jsp</code>	This JSP is used to draw the menubar in the detached window for a detached channel.
<code>providerCommands.jsp</code>	Displays the provider command bar for each channel inside the table container.
<code>providerWrapper.jsp</code>	This JSP is used to draw the wrapper containing the attach, edit, and help links around the channel displayed in the detached window.
<code>tabcustomtable.jsp</code>	Displays the table container's content view.
<code>tablecolumn.jsp</code>	Handles the left, center and right columns of a table. (Dynamically included.)
<code>tablemaximized.jsp</code>	Handles the channel in the maximized state on the Desktop front page. This JSP is used to draw the HTML around the channel displayed in the maximized state.

`tabletopbottom.jsp` Handles the top and bottom channels of a table.
(Dynamically included.)

JSPTableContainerProvider JSPs

JSPTableContainer JSPs are located in the `/etc/opt/SUNWps/desktop/default/JSPTableContainerProvider` directory. JSPTableContainerProvider JSPs are the default JSPs that are used by the JSPTableContainerProvider channels.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>header.jsp</code>	Displays the product banner that includes the user reference links for the table container.
<code>launchPopup.jsp</code>	Displays the detached windows that are detached from the table container. (Dynamically included.)
<code>leafWrapper.jsp</code>	Displays the content view for each channel inside the table container.
<code>menubar.jsp</code>	Displays the menubar that has the Home, Tabs, Theme, Help, and Logout links.
<code>popup.jsp</code>	This JSP is used to draw the content in the detached window for the channel in the detached mode.
<code>popupMenubar.jsp</code>	This JSP is used to draw the menubar in the detached window for a detached channel.
<code>providerCommands.jsp</code>	Displays the provider command bar for each channel inside the table container.
<code>providerWrapper.jsp</code>	This JSP is used to draw the wrapper containing the attach, edit, and help links around the channel displayed in the detached window.
<code>table.jsp</code>	Displays the table container's content view, which is used by the table container inside a frame tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)

<code>tablecolumn.jsp</code>	Handles the left, center and right columns of a table. (Dynamically included.)
<code>tablemaximized.jsp</code>	Handles the channel in the maximized state on the Desktop front page. This JSP is used to draw the HTML around the channel displayed in the maximized state.
<code>tabletopbottom.jsp</code>	Handles the top and bottom channels of a table. (Dynamically included.)
<code>tabtable.jsp</code>	Displays the table container's content view, which is used by the table container inside a tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>toptable.jsp</code>	Displays the table container's content view, which is used by the table container that is the top most container in the Desktop. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)

Miscellaneous JSPs

Miscellaneous JSPs are located in the `/etc/opt/SUNWps/desktop/default` directory. These JSPs are used by more than one channel, and are also used as a default if the named JSP is not found in the provider or channel subdirectory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>PortletBanner.jsp</code>	This JSP is used to draw the banner on the edit page of a JSR 168 portlet when the edit button is clicked on the portlet.
<code>PortletEdit.jsp</code>	This JSP is used to draw the edit page of a JSR 168 portlet when the edit button is clicked on the portlet.
<code>PortletHelp.jsp</code>	This JSP is used as a wrapper for the portlet's help content on the help page of a JSR 168 portlet when the help button is clicked on the portlet.

<code>PortletMenubar.jsp</code>	This JSP is used to draw the menubar on the edit page of a JSR 168 portlet when the edit button is clicked on the portlet.
<code>defaultHeader.jsp</code>	Displays the default product banner that includes the user reference links.
<code>defaultMenubar.jsp</code>	Displays the default menubar that includes the user reference links.
<code>launchPopup.jsp</code>	Displays the detached windows that are detached from the table containers. (Dynamically included.)
<code>providerCommands.jsp</code>	Displays the minimize, maximize, help, edit, detach, remove links in the channel title bar.
<code>searchbox.jsp</code>	Displays the search box that are used in the desktop header area.
<code>singlePreferenceHeader.jsp</code>	Displays the product banner that is used by the single containers.
<code>singlePreferenceMenubar.jsp</code>	Displays the menubar that is used by the single containers.
<code>tabPreferenceHeader.jsp</code>	Displays the product banner that is used by the tab containers.
<code>tabPreferenceMenubar.jsp</code>	Displays the menubar that is used by the tab containers.
<code>tabPreferenceHeader.jsp</code>	Displays the product banner that is used by the table containers.
<code>tablePreferenceHeader.jsp</code>	Displays the product banner that is used by the table containers.
<code>tablePreferenceMenubar.jsp</code>	Displays the menubar that is used by the table containers.

SampleSimpleWebService JSPs

SampleSimpleWebService JSPs are located in the `/etc/opt/SUNWps/desktop/default/SampleSimpleWebService` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

CurrencyExchangeService.wsd1	Displays the WSDL file for the current exchange web service.
webserviceContent.jsp	Displays the Content view of the simple web service channels.
webserviceInputEdit.jsp	Displays the Edit view of the simple web service channels.

SampleSimpleWebServiceConfigurable JSPs

SampleSimpleWebService JSPs are located in the `/etc/opt/SUNWps/desktop/default/SampleSimpleWebServiceConfigurable` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

CurrencyExchangeService.wsd1	Displays the WSDL file for the current exchange web service.
webserviceContent.jsp	Displays the Content view of the simple web service configurable channels.
webserviceWsd1Edit.jsp	Displays the Edit view of the simple web service configurable channels.

Search JSPs

Search JSPs are located in the `/etc/opt/SUNWps/desktop/default/Search` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

advQuery.jsp	Converts the advanced query to the list format. This JSP consists mostly of Java™ code, which is exposed so that the advanced search query list can be customized according to the schema changes.
--------------	--

<code>advancedSearch.jsp</code>	<p>Displays the interface to perform an advanced search, including the description menu. This JSP uses the <code>SearchRequestBean</code> to store request parameters and display the form values. The bean reduces Java™ scriptlets in the JSP.</p> <p>You use this JSP to make changes to the advanced search interface, removal for search fields, and addition of fields.</p>
<code>basicSearch.jsp</code>	Displays the interface to perform a basic search.
<code>browseHeader.jsp</code>	Contains the browse related code that shows up in the browse interface. Includes the category tree Home link and the Search in all categories, and Search within a category radio buttons.
<code>browseOnly.jsp</code>	Sets and executes the parameters for category browsing using the search tag library. This JSP sets all the parameters required to browse and executes the search and includes the <code>browseResults.jsp</code> page.
<code>browseResults.jsp</code>	Displays category tree in the browse section. (It consists of many Java scriptlets, so modifying this JSP requires good Java proficiency.)
<code>browseSearch.jsp</code>	Sets and executes the parameters for searching and browsing within categories using the Search tag library. The JSP sets the <code>RDMType</code> to <code>rd-request</code> and query language to search, and sets other search parameters. It includes the <code>browseSearchResults.jsp</code> page to display the category matches.
<code>browseSearchResults.jsp</code>	Displays the number of category matches found and the links to matching categories. (It consists of many Java scriptlets, so modifying this JSP requires good Java proficiency.)
<code>descMenu.jsp</code>	Contains the description menu, that is, the Full, Brief, and Title menus. This menu is included in the basic, advanced, and browse interfaces.
<code>error.jsp</code>	Displays error messages.
<code>pageFooter.jsp</code>	Displays the list of pages, Next, and Previous links.

<code>psSearch.jsp</code>	Is the portal server related JSP file. The user profile property values are retrieved from the portal server. The customer can substitute this file if the values can be retrieved from other data stores.
<code>results.jsp</code>	Specifies the number of matches found and displays document results, score, title, description, and so on, for each document. (Consists of some Java scriptlets.)
<code>score.jsp</code>	Computes the scale that displays the document match relevance.
<code>searchContent.jsp</code>	<p>Displays the Content view of the Search channel, and delegates the request to other search JSPs, based on the request type. The basic search, advanced search, or browse interfaces are displayed based on the requested mode. The search results are displayed based on the request type.</p> <p>This JSP includes the <code>advancedSearch.jsp</code> or <code>basicSearch.jsp</code> based on user selection. <code>browseSearch.jsp</code> and <code>searchOnly.jsp</code> are included only if the user has specified a query; otherwise the category tree (no search) is displayed from <code>browseOnly.jsp</code>. The <code>pageFooter.jsp</code> is included to display the pagination bar in the Search channel.</p>
<code>searchDoEdit.jsp</code>	Invoked when the user completes processing the Edit page of the Search channel.
<code>searchEdit.jsp</code>	Invoked when the user clicks the Edit button of the Search channel.
<code>searchMenu.jsp</code>	Contains the HTML ribbon for Basic, Advanced, and Browse links.
<code>searchOnly.jsp</code>	Sets and executes the parameters for search using search the tag library. The <code>advQuery.jsp</code> is included if its an advanced search. This JSP includes the <code>results.jsp</code> to display the document matches.

SearchProvider JSPs

SearchProvider JSPs are located in the

`/etc/opt/SUNWps/desktop/default/SearchProvider` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>advQuery.jsp</code>	Converts the advanced query to the list format. This JSP consists mostly of Java code, which is exposed so that the advanced search query list can be customized according to the schema changes.
<code>advancedSearch.jsp</code>	Displays the interface to perform an advanced search, including the description menu. This JSP uses the <code>SearchRequestBean</code> to store request parameters and display the form values. The bean reduces Java scriptlets in the JSP. You use this JSP to make changes to the advanced search interface, removal for search fields, and addition of fields.
<code>browseHeader.jsp</code>	Contains the browse-related code that shows up in the browse interface. Includes the category tree Home link and the Search in all categories, and Search within a category radio buttons.
<code>basicSearch.jsp</code>	Displays the interface to perform a basic search.
<code>browseOnly.jsp</code>	Sets and executes the parameters for category browsing using the search tag library. This JSP sets all the parameters required to browse and executes the search and includes the <code>browseResults.jsp</code> page.
<code>browseResults.jsp</code>	Displays category tree in the browse section. (It consists of many Java scriptlets, so modifying this JSP requires good Java proficiency.)

<code>browseSearch.jsp</code>	Sets and executes the parameters for searching and browsing within categories using the Search tag library. The JSP sets the RDMType to rd-request and query language to search, and sets other search parameters. It includes the <code>browseSearchResults.jsp</code> page to display the category matches.
<code>browseSearchResults.jsp</code>	Displays the number of category matches found and the links to matching categories. (It consists of many Java scriptlets, so modifying this JSP requires good Java proficiency.)
<code>descMenu.jsp</code>	Contains the description menu, that is, the Full, Brief, and Title menus. This menu is included in the basic, advanced, and browse interfaces.
<code>error.jsp</code>	Displays error messages.
<code>pageFooter.jsp</code>	Displays the list of pages, Next, and Previous links.
<code>psSearch.jsp</code>	Is the portal server related JSP file. The user profile property values are retrieved from the portal server. The customer can substitute this file if the values can be retrieved from other data stores.
<code>results.jsp</code>	Specifies the number of matches found and displays document results, score, title, description for each document. (Consists of some Java scriptlets.)
<code>score.jsp</code>	Computes the scale that displays the document match relevance.

<code>searchContent.jsp</code>	<p>Displays the Content view of the Search channel, and delegates the request to other search JSPs, based on the request type. The basic search, advanced search, or browse interfaces are displayed based on the requested mode. The search results are displayed based on the request type.</p> <p>This JSP includes the <code>advancedSearch.jsp</code> or <code>basicSearch.jsp</code> based on user selection. <code>browseSearch.jsp</code> and <code>searchOnly.jsp</code> are included only if the user has specified a query; otherwise the category tree (no search) is displayed from <code>browseOnly.jsp</code>. The <code>pageFooter.jsp</code> is included to display the pagination bar in the Search channel.</p>
<code>searchDoEdit.jsp</code>	<p>Invoked when the user completes processing the Edit page of the Search channel.</p>
<code>searchEdit.jsp</code>	<p>Invoked when the user clicks the Edit button of the Search channel.</p>
<code>searchMenu.jsp</code>	<p>Contains the HTML ribbon for Basic, Advanced, and Browse links.</p>
<code>searchOnly.jsp</code>	<p>Sets and executes the parameters for search using search the tag library. The <code>advQuery.jsp</code> is included if its an advanced search. This JSP includes the <code>results.jsp</code> to display the document matches.</p>

SimpleWebServiceConfigurableProvider JSPs

SimpleWebServiceConfigurableProvider JSPs are located in the `/etc/opt/SUNWps/desktop/default/SimpleWebServiceConfigurableProvider` directory. These JSPs are the default JSPs that are used by the SimpleWebServiceConfigurableProvider channels.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>webserviceContent.jsp</code>	<p>Displays the Content view of the simple web service configurable channels.</p>
------------------------------------	---

<code>webserviceWsdlEdit.jsp</code>	Displays the Edit view of the simple web service configurable channels.
-------------------------------------	---

SimpleWebServiceProvider JSPs

SimpleWebServiceProvider JSPs are located in the `/etc/opt/SUNWps/desktop/default/SimpleWebServiceProvider` directory. These JSPs are default JSPs that are used by the SimpleWebServiceProvider channels.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>webserviceContent.jsp</code>	Displays the Content view of the simple web service configurable channels.
<code>webserviceInputEdit.jsp</code>	Displays the Edit view of the simple web service channels.

Subscriptions JSPs

The Subscriptions channel JSPs are located in the `/etc/opt/SUNWps/desktop/default/Subscriptions` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>subscontent.jsp</code>	Displays the list of subscriptions per type of subscriptions (such as category subscriptions, discussions subscriptions, saved search subscriptions). For each, the list of user's subscriptions labels and associated document hit link is displayed. The hit link, points the end user to display a detailed view of the information matching their subscription.
------------------------------	---

<code>subsdooedit.jsp</code>	<p>Used to manage (such as delete/update) existing subscriptions, or adding new subscriptions when the user clicks on “subscribe to” links from the search or discussion channel.</p> <p>This JSP segments the subscriptions in to three types: category subscriptions, discussions subscriptions, saved search subscriptions.</p>
<code>subsededit.jsp</code>	<p>This JSP is triggered to handle the subscriptions changes made by the end user in the page presented by <code>subsededit.jsp</code>. The role of this JSP, is to update the Identity Server subscriptions service attributes holding the subscription information.</p>

SubscriptionsProvider JSPs

The SubscriptionsProvider JSPs are located in the `/etc/opt/SUNWps/desktop/default/SubscriptionsProvider` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>subcontentent.jsp</code>	<p>Displays the list of subscriptions per type of subscriptions (such as category subscriptions, discussions subscriptions, saved search subscriptions). For each, the list of user’s subscriptions labels and associated document hit link is displayed. The hit link, points the end user to display a detailed view of the information matching their subscription.</p>
<code>subsdooedit.jsp</code>	<p>Used to manage (such as delete/update) existing subscriptions, or adding new subscriptions when the user clicks on “subscribe to” links from the search or discussion channel.</p> <p>This JSP segments the subscriptions in to three types: category subscriptions, discussions subscriptions, saved search subscriptions.</p>

`subseddit.jsp`

This JSP is triggered to handle the subscriptions changes made by the end user in the page presented by `subseddit.jsp`. The role of this JSP, is to update the Identity Server subscriptions service attributes holding the subscription information.

TabJSPEditContainer JSPs

TabJSPEditContainer JSPs are located in the `/etc/opt/SUNWps/desktop/default/TabJSPEditContainer` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

`contentLayoutBar.jsp`

Displays the Edit page for channels in JSPTabContainer.

`doedit.jsp`

Is the process page for TabJSPEditContainer.

`tabedit.jsp`

Displays the Content and Layout links and the current selected tab on the Edit page.

JSPs in the sampleportal Directory

This section provides a listing with description of the JSPs in the `/etc/opt/SUNWps/desktop/sampleportal/*` directory. It contains

- [FrameTabContainer JSPs](#)
- [JSPContentContainer JSPs](#)
- [JSPCreateChannelContainer JSPs](#)
- [JSPCustomThemeContainer JSPs](#)
- [JSPDynamicSingleContainer JSPs](#)
- [JSPEditContainer JSPs](#)
- [JSPFrameCustomTableContainerProvider JSPs](#)
- [JSPLayoutContainer JSPs](#)
- [JSPPopupContainer JSPs](#)

- [JSPPresetThemeContainer JSPs](#)
- [JSPSingleContainer JSPs](#)
- [JSPTabContainer JSPs](#)
- [JSPTabCustomTableContainerProvider JSPs](#)
- [JSPTableContainerProvider JSPs](#)
- [Miscellaneous JSPs](#)
- [PredefinedFrontPageFramePanelContainerProvider and PredefinedFrontPageTabPanelContainerProvider](#)
- [PredefinedSamplesFramePanelContainerProvider and PredefinedSamplesTabPanelContainerProvider](#)
- [SampleJSP JSPs](#)
- [SampleSimpleWebService JSPs](#)

NOTE All the JSPs in sampleportal directory inherit the new Themes.

FrameTabContainer JSPs

FrameTabContainer JSPs are located in the `/etc/opt/SUNWps/desktop/sampleportal/FrameTabContainer` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>banner.jsp</code>	Contains the required JavaScript™ and the body tag for the right frame of the frame tab container.
<code>frameset.jsp</code>	Contains the HTML source for frames of the frameset container.
<code>frametab.jsp</code>	Is the main JSP for the frame tab container and throws out the requested content for each frame based on the request parameters from the frameset JSP.
<code>frametabedit.jsp</code>	Displays the Edit page for the frame tab container where new pages can be added, removed, or renamed.

<code>frametabmenu.jsp</code>	Displays the left frame for the frame tab container that has the list of available pages and links to them.
<code>header.jsp</code>	Displays the product banner that is used by the frame tab container.
<code>makeNewTab.jsp</code>	Provides the content for the Make New Page part of the Sections page in the frame tab container.
<code>makeTopic.jsp</code>	Provides the content for each of the page topics in the Make New Page.
<code>menubar.jsp</code>	Displays the menubar that has the Home, Tabs, Theme, Help, and Logout links.
<code>remove.jsp</code>	Displays the remove, rename, and start links for each of the selected pages of the frame tab container in the Current Tab Settings page.
<code>removeRenameTab.jsp</code>	Displays the Start Page, Tab, and Actions part of the Current Tab Settings page for the frame tab container.
<code>selectedTab.jsp</code>	Displays the content for the right frame of the frame tab container.

JSPContentContainer JSPs

The JSPContentContainer JSPs are located in the `/etc/opt/SUNWps/desktop/sampleportal/JSPContentContainer` directory. These JSPs are used for the content view when the Content link is selected in a JSP-based table container.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>contentedit.jsp</code>	Displays the content Edit page.
------------------------------	---------------------------------

JSPCreateChannelContainer JSPs

The JSPCreateChannelContainer JSPs are located in the `/etc/opt/SUNWps/desktop/sampleportal/JSPCreateChannelContainer` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>createchannel.jsp</code>	These JSPs are used for the User Defined Channel page on the Desktop.
<code>createchannelcontent.jsp</code>	
<code>createchanneldoedit.jsp</code>	
<code>createchanneledit.jsp</code>	
<code>createchannelui.jsp</code>	
<code>deletchannel.jsp</code>	
<code>deletchannelui.jsp</code>	

JSPCustomThemeContainer JSPs

JSPCustomThemeContainer JSPs are located in the `/etc/opt/SUNWps/desktop/sampleportal/JSPCustomThemeContainer` directory. These JSPs are used when the Custom Theme link is selected in the Themes page.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>customthemedoedit.jsp</code>	Processes the result from the Custom Theme page.
<code>customthemeedit.jsp</code>	Displays the Custom Theme Edit page.
<code>themepreview.jsp</code>	Displays the preview view of the Custom Theme page.

JSPDynamicSingleContainer JSPs

The JSPDynamicSingleContainer JSPs are located in the `/etc/opt/SUNWps/desktop/sampleportal/JSPDynamicSingleContainer` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

`dynamicSingle.jsp`

Used by the `DynamicSingleContainer` to display the channel specified in the request parameter. This JSP uses the Desktop theme.

JSPEditContainer JSPs

The `JSPEditContainer` JSP is located in the `/etc/opt/SUNWps/desktop/sampleportal/JSPEditContainer` directory. These JSPs are used when the Edit icon is selected in a channel title bar inside a JSP-based container. Channels that have the `editType` defined as `EDIT_SUBSET` use these JSPs. The difference between this JSP and the one from `default` directory is that this includes the Desktop theme style.

The following is a two column table: column one lists the file name; column two provides a brief description.

`edit.jsp`

Displays the Edit view of a channel. Also provides a wrapper around the actual Edit view for a given channel.

JSPFrameCustomTableContainerProvider JSPs

The `JSPFrameCustomTableContainerProvider` JSP is located in the `/etc/opt/SUNWps/desktop/sampleportal/JSPFrameCustomTableContainerProvider` directory.

`JSPFrameCustomTableContainerProvider` JSPs are mainly used by the user created pages in the frame tab container. When the user creates a new page from the Sections page in the frame tab container, a new table container is created dynamically. The following JSPs are used by the user created table containers. The difference between this JSP and the one from `default` directory is that this includes the Desktop theme style.

The following is a two column table: column one lists the file name; column two provides a brief description.

`framecustomtable.jsp`

Displays the content for the newly created page (table container) from the Sections page.

JSPLayoutContainer JSPs

The JSPLayoutContainer JSPs are located in the `/etc/opt/SUNWps/desktop/sampleportal/JSPLayoutContainer` directory. These JSPs are used to display the Layout view when the Layout link is selected in a JSP-based table container.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>layoutedit.jsp</code>	Displays the Layout Edit page.
-----------------------------	--------------------------------

JSPPopupContainer JSPs

The JSPPopupContainer JSPs are located in the `/etc/opt/SUNWps/desktop/sampleportal/JSPPopupContainer` directory. These JSPs were used to detach a channel from a JSP-based Desktop. However, this container is no longer used by the sample portal. The detached windows in sample portal are now drawn by the `popup.jsp`, `popupmenubar.jsp`, and `providerwrapper.jsp` in JSPTableContainer. These JSPs are present for backward compatibility only.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>popup.jsp</code>	Displays the contents of the channel inside the detached window.
<code>popupMenubar.jsp</code>	Displays the Update, Close, and Logout links inside the popup window.
<code>providerWrapper.jsp</code>	Combines the above JSPs.

JSPPresetThemeContainer JSPs

The JSPPresetThemeContainer JSPs are located in the `/etc/opt/SUNWps/desktop/sampleportal/JSPPresetThemeContainer` directory. These JSPs are used when the Preset Themes link is selected in the Themes page.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>themedoedit.jsp</code>	Processes the result from the Preset Theme page.
<code>themededit.jsp</code>	Displays the Preset Theme Edit page.
<code>themepreview.jsp</code>	This file exists for backward compatibility and shows that the theme changes apply to a channel only.

JSPSingleContainer JSPs

The JSPSingleContainer JSPs are located in the `/etc/opt/SUNWps/desktop/sampleportal/JSPSingleContainer` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>header.jsp</code>	Displays the product banner that is used by the Single Container page.
<code>menubar.jsp</code>	Displays the menubar that has the Home, Tabs, Theme, Help, and Logout links.
<code>single.jsp</code>	Displays the content for JSPSingleContainerProvider.

JSPTabContainer JSPs

The JSPTabContainer JSPs are located in the `/etc/opt/SUNWps/desktop/sampleportal/JSPTabContainer` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>header.jsp</code>	Displays the product banner that is used by the Tab Container page. (Dynamically included.)
-------------------------	---

<code>menubar.jsp</code>	Displays the menubar that has the Home, Tabs, Theme, Help, and Logout links. (Dynamically included.)
<code>selectedTab.jsp</code>	Displays the tab image for the current selected tab in the tab container.
<code>tab.jsp</code>	Is the main JSP for the tab container. It draws the content page for the tab container. (Dynamically includes <code>header.jsp</code> and <code>menubar.jsp</code> .)
<code>tabedit.jsp</code>	Displays the Edit page for the tab container where new pages can be added, removed, or renamed.
<code>tabs.jsp</code>	Generates the available tabs and the links for them to be activated on the Desktop.

NOTE The `header.jsp` and `menubar.jsp` have the new Themes look.

JSPTabCustomTableContainerProvider JSPs

The JSPTabCustomTableContainer JSPs is located in the `/etc/opt/SUNWps/desktop/sampleportal/JSPTabCustomTableContainerProvider` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>tabcustomtable.jsp</code>	Displays the table container's content view.
---------------------------------	--

JSPTableContainerProvider JSPs

The JSPTableContainer JSPs are located in the `/etc/opt/SUNWps/desktop/sampleportal/JSPTableContainerProvider` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>header.jsp</code>	Displays the product banner that includes the user reference links for the table container.
<code>leafWrapper.jsp</code>	Displays the content view for each channel inside the table container.
<code>menubar.jsp</code>	Displays the menubar that has the Home, Theme, Help, and Logout links.
<code>popup.jsp</code>	This JSP is used to draw the content in the detached window for the channel in the detached mode.
<code>popupMenubar.jsp</code>	This JSP is used to draw the menubar in the detached window for a detached channel.
<code>providerWrapper.jsp</code>	This JSP is used to draw the wrapper containing the attach, edit, and help links around the channel displayed in the detached window.
<code>table.jsp</code>	Displays the table container's content view, which is used by the table container inside a frame tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>tabtable.jsp</code>	Displays the table container's content view, which is used by the table container inside a tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>toptable.jsp</code>	Displays the table container's content view, which is used by the table container that is the top most container in the Desktop. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)

Miscellaneous JSPs

Miscellaneous JSPs are located in the `/etc/opt/SUNWps/desktop/sampleportal` directory. These JSPs are used by more than one channel, and are also used as a default if the named JSP is not found in the provider or channel subdirectory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>PortletBanner.jsp</code>	This JSP is used to draw the banner on the edit page of a JSR 168 portlet when the edit button is clicked on the portlet.
<code>PortletMenubar.jsp</code>	This JSP is used to draw the menubar on the edit page of a JSR 168 portlet when the edit button is clicked on the portlet.
<code>defaultHeader.jsp</code>	Displays the default product banner that includes the user reference links.
<code>defaultMenubar.jsp</code>	Displays the default menubar that includes the user reference links.
<code>framePreferenceHeader.jsp</code>	Displays the product banner that is used by the frame tab containers.
<code>framePreferenceMenubar.jsp</code>	Displays the menubar that is used by the frame tab containers.
<code>searchbox.jsp</code>	Displays the search box that are used in the desktop header area.
<code>singlePreferenceHeader.jsp</code>	Displays the product banner that is used by the single containers.
<code>singlePreferenceMenubar.jsp</code>	Displays the menubar that is used by the single containers.
<code>tabPreferenceHeader.jsp</code>	Displays the product banner that is used by the tab containers.
<code>tabPreferenceMenubar.jsp</code>	Displays the menubar that is used by the tab containers.
<code>tablePreferenceHeader.jsp</code>	Displays the product banner that is used by the table containers.
<code>tablePreferenceMenubar.jsp</code>	Displays the menubar that is used by the table containers.

PredefinedFrontPageFramePanelContainerProvider

The `PredefinedFrontPageFramePanelContainerProvider` JSPs are located in the `/etc/opt/SUNWps/desktop/sampleportal/PredefinedFrontPageFramePanelContainerProvider` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>header.jsp</code>	Displays the product banner that includes the user reference links for the table container.
<code>launchPopup.jsp</code>	Displays the detached windows that are detached from the table container. (Dynamically included.)
<code>leafWrapper.jsp</code>	Displays the content view for each channel inside the table container.
<code>menubar.jsp</code>	Displays the menubar that has the Home, Theme, Help, and Logout links.
<code>popup.jsp</code>	This JSP is used to draw the content in the detached window for the channel in the detached mode.
<code>popupMenubar.jsp</code>	This JSP is used to draw the menubar in the detached window for a detached channel.
<code>providerCommands.jsp</code>	Displays the channel command buttons and links, such as Remove, Minimize/Maximize, Edit, and Help.
<code>providerWrapper.jsp</code>	This JSP is used to draw the wrapper containing the attach, edit, and help links around the channel displayed in the detached window.
<code>table.jsp</code>	Displays the table container's content view, which is used by the table container inside a frame tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>tablecolumn.jsp</code>	Handles the left, center and right columns of a table. (Dynamically included.)

<code>tablemaximized.jsp</code>	Handles the channel in the maximized state on the Desktop front page. This JSP is used to draw the HTML around the channel displayed in the maximized state.
<code>tabletopbottom.jsp</code>	Handles the top and bottom channels of a table. (Dynamically included.)
<code>tabtable.jsp</code>	Displays the table container's content view, which is used by the table container inside a tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>toptable.jsp</code>	Displays the table container's content view, which is used by the table container that is the top most container in the Desktop. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)

PredefinedFrontPageTabPanelContainerProvider

The `PredefinedFrontPageTabPanelContainerProvider` JSPs are located in the `/etc/opt/SUNWps/desktop/sampleportal/PredefinedFrontPageTabPanelContainerProvider` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>header.jsp</code>	Displays the product banner that includes the user reference links for the table container.
<code>launchPopup.jsp</code>	Displays the detached windows that are detached from the table container. (Dynamically included.)
<code>leafWrapper.jsp</code>	Displays the content view for each channel inside the table container.
<code>menubar.jsp</code>	Displays the menubar that has the Home, Theme, Help, and Logout links.
<code>popup.jsp</code>	This JSP is used to draw the content in the detached window for the channel in the detached mode.

<code>popupMenubar.jsp</code>	This JSP is used to draw the menubar in the detached window for a detached channel.
<code>providerCommands.jsp</code>	Displays the channel command buttons and links, such as Remove, Minimize/Maximize, Edit, and Help.
<code>providerWrapper.jsp</code>	This JSP is used to draw the wrapper containing the attach, edit, and help links around the channel displayed in the detached window.
<code>table.jsp</code>	Displays the table container's content view, which is used by the table container inside a frame tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>tablecolumn.jsp</code>	Handles the left, center and right columns of a table. (Dynamically included.)
<code>tablemaximized.jsp</code>	Handles the channel in the maximized state on the Desktop front page. This JSP is used to draw the HTML around the channel displayed in the maximized state.
<code>tabletopbottom.jsp</code>	Handles the top and bottom channels of a table. (Dynamically included.)
<code>tabtable.jsp</code>	Displays the table container's content view, which is used by the table container inside a tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>toptable.jsp</code>	Displays the table container's content view, which is used by the table container that is the top most container in the Desktop. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)

PredefinedSamplesFramePanelContainerProvider

The `PredefinedSamplesFramePanelContainerProvider` JSPs are located in the `/etc/opt/SUNWps/desktop/sampleportal/PredefinedSamplesFramePanelContainerProvider` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>header.jsp</code>	Displays the product banner that includes the user reference links for the table container.
<code>launchPopup.jsp</code>	Displays the detached windows that are detached from the table container. (Dynamically included.)
<code>leafWrapper.jsp</code>	Displays the content view for each channel inside the table container.
<code>menubar.jsp</code>	Displays the menubar that has the Home, Theme, Help, and Logout links.
<code>popup.jsp</code>	This JSP is used to draw the content in the detached window for the channel in the detached mode.
<code>popupMenubar.jsp</code>	This JSP is used to draw the menubar in the detached window for a detached channel.
<code>providerCommands.jsp</code>	Displays the channel command buttons and links, such as Remove, Minimize/Maximize, Edit, and Help.
<code>providerWrapper.jsp</code>	This JSP is used to draw the wrapper containing the attach, edit, and help links around the channel displayed in the detached window.
<code>table.jsp</code>	Displays the table container's content view, which is used by the table container inside a frame tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>tablecolumn.jsp</code>	Handles the left, center and right columns of a table. (Dynamically included.)
<code>tablemaximized.jsp</code>	Handles the channel in the maximized state on the Desktop front page. This JSP is used to draw the HTML around the channel displayed in the maximized state.
<code>tabletopbottom.jsp</code>	Handles the top and bottom channels of a table. (Dynamically included.)

<code>tabtable.jsp</code>	Displays the table container's content view, which is used by the table container inside a tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>toptable.jsp</code>	Displays the table container's content view, which is used by the table container that is the top most container in the Desktop. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)

PredefinedSamplesTabPageContainerProvider

The `PredefinedSamplesTabPageContainerProvider` JSPs are located in the `/etc/opt/SUNWps/desktop/sampleportal/PredefinedSamplesTabPageContainerProvider` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>header.jsp</code>	Displays the product banner that includes the user reference links for the table container.
<code>launchPopup.jsp</code>	Displays the detached windows that are detached from the table container. (Dynamically included.)
<code>leafWrapper.jsp</code>	Displays the content view for each channel inside the table container.
<code>menubar.jsp</code>	Displays the menubar that has the Home, Theme, Help, and Logout links.
<code>popup.jsp</code>	This JSP is used to draw the content in the detached window for the channel in the detached mode.
<code>popupMenubar.jsp</code>	This JSP is used to draw the menubar in the detached window for a detached channel.
<code>providerCommands.jsp</code>	Displays the channel command buttons and links, such as Remove, Minimize/Maximize, Edit, and Help.
<code>providerWrapper.jsp</code>	This JSP is used to draw the wrapper containing the attach, edit, and help links around the channel displayed in the detached window.

<code>table.jsp</code>	Displays the table container's content view, which is used by the table container inside a frame tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>tablecolumn.jsp</code>	Handles the left, center and right columns of a table. (Dynamically included.)
<code>tablemaximized.jsp</code>	Handles the channel in the maximized state on the Desktop front page. This JSP is used to draw the HTML around the channel displayed in the maximized state.
<code>tabletopbottom.jsp</code>	Handles the top and bottom channels of a table. (Dynamically included.)
<code>tabtable.jsp</code>	Displays the table container's content view, which is used by the table container inside a tab container. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)
<code>toptable.jsp</code>	Displays the table container's content view, which is used by the table container that is the top most container in the Desktop. (Dynamically includes <code>launchPopup.jsp</code> , <code>tablecolumn.jsp</code> , and <code>tabletopbottom.jsp</code> .)

SampleJSP JSPs

The SampleJSP JSPs are located in the

`/etc/opt/SUNWps/desktop/sampleportal/SampleJSP` directory. These JSPs are used by the SampleJSP channel.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>samplecontent.jsp</code>	Displays the contents of the Sample JSP channel. This JSP uses of the Desktop theme, and is used as an example of the <code>channelHighlightColor</code> attribute.
--------------------------------	---

SampleSimpleWebService JSPs

The SampleSimpleWebService JSP is located in the `/etc/opt/SUNWps/desktop/sampleportal/SampleSimpleWebService` directory.

The following is a two column table: column one lists the file name; column two provides a brief description.

<code>webserviceContent.jsp</code>	Displays the Content view of the simple web service channels.
------------------------------------	---

A

- AddressBook channel
 - display profile definition 165
 - display profile properties 165
- AddressBookProvider 92, 93, 309
- AddressBookProvider display profile properties
 - applicationHelperEdit 93
 - applicationHelperURL 93
 - displayEntries 93
 - dpEditAttributes 93
 - introduction 93
 - maxEntries 93
 - numEntries 93
 - ssoAdapter 93
 - ssoEditAttributes 93
 - sunPortalABSortBy 93
 - sunPortalABSortBySelectOptions 93
 - sunPortalABSortOrder 93
 - sunPortalABSortOrderSelectOptions 94
- applications list channel
 - display profile definition 166
 - display profile properties 166
- AppProvider 94, 95, 311
- AppProvider display profile properties
 - introduction 95
 - targets 95
 - userApps 95
 - windowPref 95

B

- Bookmark channel
 - display profile definition 166
 - display profile properties 167
- BookmarkProvider 95, 96, 311
- BookmarkProvider display profile properties
 - introduction 96
 - targets 96
 - windowPref 96

C

- Calendar channel
 - display profile definition 167
 - display profile properties 168
- CalendarProvider 96, 98, 313
- CalendarProvider display profile properties
 - applicationHelperEdit 98
 - applicationHelperURL 98
 - calendar 98
 - calendarSelectOptions 98
 - disableTaskEventURLs 98
 - dpEditAttributes 98
 - introduction 98
 - loadSubscribedCalendars 98
 - ssoAdapter 98
 - ssoEditAttributes 98
 - view 98
 - viewSelectOptions 98

- Channel display profile properties
 - introduction [42](#)
- Channel object [22, 24](#)
- CollaborationTabPanelContainer [194](#)
- Container object [24](#)
- Container provider display profile properties [77](#)
 - customThemeChannel [77](#)
 - editContainerName [78](#)
 - presetThemeChannel [77](#)

D

Desktop

- Administrator Customizations [19](#)
- Developer Customizations [18](#)
- End-User Customizations [18](#)
- Types of Customizations [17](#)
- Using the Display Profile [20](#)

DiscussionLite channel

- display profile definition [169](#)
- introduction [168](#)

DiscussionProvider [88, 89](#)

DiscussionProvider display profile properties

- anonymousAuthor [90](#)
- dbname [89](#)
- defaultDiscussionDisplay [89](#)
- defaultExpansionThreshold [89](#)
- defaultFilter [89](#)
- displaySearch [90](#)
- introduction [89](#)
- ratingText [90](#)
- searchServer [89](#)
- showDescription [90](#)
- viewDiscussionWindow [90](#)
- viewHits [89](#)

Discussions [471](#)

Discussions channel

- display profile definition [169](#)
- display profile properties [170](#)
- introduction [168](#)

DiscussionsProvider [473](#)

Display Profile

- Channel object [22, 24](#)

- Channel properties [42](#)
- Container object [24](#)
- Container properties [73](#)
- Definitions [20, 47, 133](#)
- Document [20, 21, 22](#)
 - structure [23](#)
- Global properties [42, 44](#)
- Hierarchical Structuring [20](#)
- Introduction [20](#)
 - definitions [20](#)
 - documents [20, 21, 22](#)
 - hierarchical structuring [20](#)
 - objects [22](#)
- Leaf provider properties [79](#)
- Objects [22](#)
 - channel [22, 24](#)
 - container [24](#)
 - property [23](#)
 - provider [22](#)
- properties [41](#)
 - Channel properties [42](#)
 - Container properties [73](#)
 - Global properties [42, 44](#)
 - Leaf provider properties [79](#)
 - Provider properties [42](#)
 - types [43](#)
- Property object [23](#)
- Provider object [22](#)
- Provider properties [42, 80, 127](#)
- Document Type Definition [20](#)
- dp-anon.xml [80, 192](#)
- dp-org.xml [79, 191](#)
- dp-org-final.xml [79](#)
- dp-providers.xml [79](#)
- DTD [20](#)
- DummyChannel [474](#)

E

- error [328](#)

F

FrameCustomTableContainer 163, 164
 FrameTabContainer 50, 60, 62, 141, 152, 154, 161,
 162, 163, 190, 204, 206, 208, 209, 214, 464, 493

G

Global display profile properties
 introduction 42, 44

I

IMProvider 98, 99, 100, 475
 IMProvider display profile properties
 authMethod 100
 authUsernameAttr 100
 clientRunMode 100
 codebase 100
 contactGroup 100
 introduction 99
 mux 100
 muxport 100
 netletRule 100
 password 100
 port 99
 server 99
 username 100
 Instant Messaging channel
 display profile definition 171
 display profile properties 171

J

JavaServer Pages 305, 463
 JavaSever Pages 427
 JSP channel
 display profile definition 172

JSPContentContainer 51, 196, 198, 202, 207, 209, 475,
 494
 JSPCreateChannelContainer 51, 143, 144, 494
 JSPCustomThemeContainer 51, 145, 146, 198, 495
 JSPDynamicSingleContainer 51, 154, 155, 464, 476,
 495
 JSPEditContainer 51, 476, 496
 JSPFrameCustomTableContainerProvider 56, 208,
 476, 496
 JSPLayoutContainer 51, 196, 202, 207, 477, 497
 JSPPopupContainer 51, 146, 147, 497
 JSPPresetThemeContainer 51, 144, 145, 198, 497
 JSPProvider 24, 80, 81, 88, 478
 JSPProvider display profile properties
 contentPage 81
 editPage 81
 introduction 81
 processPage 81
 showExceptions 81
 JSRs
 availableTabs.jsp 194
 banner.jsp 205
 frameset.jsp 205
 frametab.jsp 205, 208
 frametabedit.jsp 208, 209
 frametabmenu.jsp 205
 header.jsp 205
 menubar.jsp 194, 205
 selectedTab.jsp 194, 205
 tab.jsp 194, 199
 tabs.jsp 194
 toptable.jsp 204
 JSPSingleContainer 49, 498
 JSPSingleContainerProvider 50, 154, 196, 202, 207,
 478
 JSPTabContainer 49, 50, 69, 70, 137, 139, 141, 143,
 155, 156, 157, 190, 192, 194, 195, 196, 197, 198, 199,
 214, 223, 464, 498
 JSPTabContainerProvider 190, 197, 208, 220, 479
 JSPTabCustomTableContainer 157, 158, 499
 JSPTabCustomTableContainerProvider 54, 55, 480
 JSPTabContainer 49, 134, 136, 162, 190, 196, 199,
 202, 203, 223, 465, 481, 499
 JSPTabContainer display profile properties

- channelsColumn [136](#)
- channelsIsDetached [136](#)
- channelsIsMinimized [136](#)
- channelsRow [136](#)
- Selected [136](#)
- title [136](#)

JSPTableContainerProvider [48, 190, 203, 481](#)

L

Leaf provider display profile properties [79](#)

Login channel display profile definition [172](#)

LoginProvider [100, 101, 102, 331](#)

LoginProvider display profile properties

- federationEnabled [101](#)
- introduction [101](#)
- persistentCookie [101](#)
- preLoginURL [101](#)

Lotus Notes Address Book channel

- display profile definition [173](#)
- display profile properties [173](#)

Lotus Notes Calendar channel

- display profile definition [174](#)
- display profile properties [174](#)

Lotus Notes Mail channel

- display profile definition [174](#)
- display profile properties [175](#)

LotusNotesAddressBookProvider [102, 103, 104, 331](#)

LotusNotesAddressBookProvider display profile

properties

- applicationHelperEdit [103](#)
- applicationHelperURL [103](#)
- displayEntries [103](#)
- dpEditAttributes [103](#)
- introduction [103](#)
- maxEntries [103](#)
- numEntries [103](#)
- ssoAdapter [103](#)
- ssoEditAttributes [103](#)
- sunPortalABSORTBy [103](#)
- sunPortalABSORTBySelectOptions [104](#)
- sunPortalABSORTOrder [103](#)
- sunPortalABSORTOrderSelectOptions [104](#)

LotusNotesCalendarProvider [104, 105, 106, 334](#)

LotusNotesCalendarProvider display profile

properties

- applicationHelperEdit [105](#)
- applicationHelperURL [105](#)
- calendar [105](#)
- calendarSelectOptions [105](#)
- disableTaskEventURLs [105](#)
- dpEditAttributes [106](#)
- introduction [105](#)
- loadSubscribedCalendars [105](#)
- ssoAdapter [105](#)
- ssoEditAttributes [105](#)
- view [105](#)
- viewSelectOptions [106](#)

LotusNotesMailProvider [106, 107, 108, 349](#)

LotusNotesMailProvider display profile properties

- applicationHelperEdit [107](#)
- applicationHelperURL [107](#)
- displayHeaders [107](#)
- dpEditAttributes [108](#)
- introduction [107](#)
- numberHeaders [107](#)
- sentFolderCopy [107](#)
- sortOrder [107](#)
- sortOrderSelectOptions [108](#)
- ssoAdapter [107](#)
- ssoEditAttributes [108](#)

M

Mail channel

- display profile definition [175](#)
- display profile properties [176](#)

Mail Check channel

- display profile properties [176](#)

Mail check channel

- display profile definition [176](#)

MailCheckProvider [108, 109, 110, 351](#)

MailCheckProvider display profile properties

- defaultConfigParameters [109](#)
- IMAPPASSWORD [109](#)
- IMAPSERVERNAME [109](#)

- IMAPUserID [109](#)
 - introduction [109](#)
 - targets [109](#)
 - userApps [109](#)
 - MailProvider** [110, 111, 112, 352](#)
 - MailProvider display profile properties**
 - applicationHelperEdit [111](#)
 - applicationHelperURL [111](#)
 - displayHeaders [111](#)
 - dpEditAttributes [111](#)
 - introduction [111](#)
 - numberHeaders [111](#)
 - sentFolderCopy [111](#)
 - sortOrder [111](#)
 - sortOrderSelectOptions [111](#)
 - ssoAdapter [111](#)
 - ssoEditAttributes [111](#)
 - MS Exchange Address Book channel**
 - display profile definition [177](#)
 - display profile properties [177](#)
 - MS Exchange Calendar channel**
 - display profile definition [178](#)
 - display profile properties [178](#)
 - MS Exchange Mail channel**
 - display profile definition [178](#)
 - display profile properties [179](#)
 - MSEXchangeAddressBookProvider** [112, 113, 114, 355](#)
 - MSEXchangeAddressBookProvider display profile properties**
 - applicationHelperEdit [113](#)
 - applicationHelperURL [113](#)
 - displayEntries [113](#)
 - dpEditAttributes [113](#)
 - introduction [113](#)
 - maxEntries [113](#)
 - numEntries [113](#)
 - ssoAdapter [113](#)
 - ssoEditAttributes [113](#)
 - sunPortalABSorderBy [113](#)
 - sunPortalABSorderBySelectOptions [114](#)
 - sunPortalABSorderBy [113](#)
 - sunPortalABSorderBySelectOptions [114](#)
 - MSEXchangeCalendarProvider** [114, 115, 116, 357](#)
 - MSEXchangeCalendarProvider display profile properties**
 - applicationHelperEdit [115](#)
 - applicationHelperURL [115](#)
 - calendar [115](#)
 - calendarSelectOptions [115](#)
 - disableTaskEventURLs [115](#)
 - dpEditAttributes [116](#)
 - introduction [115](#)
 - loadSubscribedCalendars [115](#)
 - ssoAdapter [115](#)
 - ssoEditAttributes [116](#)
 - view [115](#)
 - viewSelectOptions [116](#)
 - MSEXchangeMailProvider** [116, 117, 118, 372](#)
 - MSEXchangeMailProvider display profile properties**
 - applicationHelperEdit [117](#)
 - applicationHelperURL [118](#)
 - displayHeaders [117](#)
 - dpEditAttributes [118](#)
 - introduction [117](#)
 - numberHeaders [117](#)
 - sentFolderCopy [117](#)
 - sortOrder [117](#)
 - sortOrderSelectOptions [118](#)
 - ssoAdapter [117](#)
 - ssoEditAttributes [118](#)
 - MyFrontPageFramePanelContainer** [136, 137, 206, 208, 209, 214](#)
 - MyFrontPageTabPanelContainer** [139, 140, 194, 197, 198, 214](#)
 - MyFrontPageTemplatePanelContainer** [150, 151, 214, 395](#)
- ## N
- Notes channel**
 - display profile definition [179](#)
 - display profile properties [180](#)
 - NotesProvider** [118, 119, 120](#)
 - NotesProvider display profile properties**
 - introduction [119](#)
 - lines [119](#)

location 119
 maxLines 119
 timeout 119

P

PersonalNoteProvider 120
 PortletSamplesTabPanelContainer 194
 PredefinedFrontPageFramePanelContainer 161
 PredefinedFrontPageFramePanelContainerProvider 60, 466, 502
 PredefinedFrontPageTabPanelContainer 155, 156, 198, 214
 PredefinedFrontPageTabPanelContainerProvider 70, 467, 503
 PredefinedFrontPageTemplatePanelContainer 158, 159
 PredefinedFrontPageTemplatePanelContainerProvider 66, 396
 PredefinedNewTabPanelContainerProvider 219
 PredefinedSamplesFramePanelContainer 162, 163
 PredefinedSamplesFramePanelContainerProvider 62, 468, 504
 PredefinedSamplesTabPanelContainer 156, 157, 198, 214
 PredefinedSamplesTabPanelContainerProvider 69, 469, 506
 PredefinedSamplesTemplatePanelContainer 159
 PredefinedSamplesTemplatePanelContainerProvider 67, 408
 PredefinedTabPanelContainer 190
 PredefinedTemplatePanelContainer 190
 PredefinedToolsTemplatePanelContainer 160
 PredefinedToolsTemplatePanelContainerProvider 64
 Property object 23
 property type
 Boolean 43
 Collection 43
 Integer 44
 Reference 44
 String 44

Provider display profile properties 42, 80, 127
 Provider object 22

R

RSS channel display profile definition 184
 RSSProvider 85

S

SamplesFramePanelContainer 141, 206, 214
 SampleSimpleWebService 483, 484, 508
 SamplesTabPanelContainer 137, 138, 194, 198, 214
 SamplesTemplatePanelContainer 151, 420
 Search 484
 Search channel
 display profile definition 181
 display profile properties 181
 SearchProvider 86, 87, 487
 SearchProvider display profile properties
 basicSearchDefault 87
 categorySearch 88
 contentPage 87
 defaultMode 87
 editPage 87
 introduction 87
 processPage 87
 searchServer 87
 showExceptions 87
 viewHits 87
 SearchTabContainer 194
 SearchTabPanelContainer 49, 138, 139, 198, 214
 Simple Web Service channel display profile definition 182
 Simple Web Service Configurable channel display profile properties 183
 Simple Web Sservice Configurable channel display profile definition 183
 SimpleWebServiceConfigurableProvider 122, 123, 489

SimpleWebServiceConfigurableProvider display

profile properties

[contentPage 124](#)
[defaultInput 124](#)
[editPage 124](#)
[introduction 123](#)
[isDefaultAvailable 124](#)
[isDefaultShowOutput 123](#)
[methodName 123](#)
[showExceptions 124](#)
[wsdlURL 123](#)

SimpleWebServiceProvider 120, 121, 122, 490**SimpleWebServiceProvider display profile**

properties

[contentPage 122](#)
[defaultInput 122](#)
[editPage 122](#)
[introduction 121](#)
[isDefaultAvailable 122](#)
[isDefaultShowOutput 122](#)
[methodName 122](#)
[showExceptions 122](#)
[wsdlURL 122](#)

Subscriptions 490**Subscriptions channel display profile definition 184****SubscriptionsProvider 90, 491****T****Tab Container**

display profile properties 77

[channelNumber 77](#)
[contentChannel 77](#)
[makeTabChannel 77](#)
[makeTabProvider 77](#)
[maxTabs 77](#)
[startTab 77](#)
[TabProperties 77](#)

TabContainerProvider 74**TabJSPEditContainer 51, 492****Table container**

[display profile properties 74](#)
[borderlessChannels 76](#)

[channelsColumn 76](#)
[channelsHasFrame 76](#)
[channelsIsDetachable 76](#)
[channelsIsDetached 76](#)
[channelsIsMaximizable 76](#)
[channelsIsMinimized 76](#)
[channelsIsMovable 76](#)
[channelsIsRemovable 76](#)
[channelsRow 76](#)
[defaultBorderlessChannel 76](#)
[defaultChannelColumn 76](#)
[defaultChannelHasFrame 75](#)
[defaultChannelIsDetachable 75](#)
[defaultChannelIsDetached 75](#)
[defaultChannelIsMinimizable 75](#)
[defaultChannelIsMinimized 75](#)
[defaultChannelIsMovable 76](#)
[defaultChannelIsRemovable 75](#)
[defaultChannelRow 76](#)
[defaultchannelsIsMaximizable 75](#)
[fullwidth_popup_height 75](#)
[fullwidth_popup_width 75](#)
[parentTabContainer 74](#)
[refreshParentContainerOnly 74](#)
[thick_popup_height 75](#)
[thick_popup_width 75](#)
[thin_popup_height 75](#)
[thin_popup_width 75](#)

Tag Library

[Descriptors 428](#)
[exceptions 437](#)
[heirarchy 429](#)
[introduction 427](#)

template

[footer.html 194, 205](#)

TemplateEditContainerProvider 58

[TemplateTabContainer 58, 64, 66, 67, 148, 150, 151, 152, 158, 159, 160, 190, 210, 214](#)

TemplateTabContainerProvider 53, 190, 374, 420**TemplateTabCustomTableContainer 160, 161**

[TemplateTabCustomTableContainerProvider 59, 421](#)

TemplateTableContainer 58, 147, 148, 190, 210, 422

[TemplateTableContainerProvider 52, 190, 377](#)

[Theme page 77](#)

ToolsTemplatePanelContainer 152, 423
Types of Desktop Customizations 17
Types of Display Profile Properties 43

U

URL scraper channel display profile definition 186
URLScrapperProvider 25, 82, 83
URLScrapperProvider display profile properties
 cookiesToForwardAll 83
 cookiesToForwardList 83
 inputEncoding 83
 introduction 82
 isEditable 83
 timeout 83
 url 82
 urlScrapperRulesetID 82
User Information channel
 display profile definition 185
 display profile properties 185
UserInfoProvider 124, 125, 389
UserInfoProvider display profile properties
 attribute 126
 authTypes 127
 cn 126
 currentDate 126
 datetime 127
 dp 126
 givenname 126
 greeting 126
 imappw 127
 introduction 125
 locale 126
 localeList 126
 localelist 127
 maxIdle 126
 Membership 127
 membershipConfirmPassword 126
 membershipNewPassword 126
 membershipOriginalPassword 126
 membershiappw 127
 netmailServiceName 127
 preferredlocale 126

 preferredtimezone 126
 sn 126
 sunPortalNetmailIMAPPASSWORD 126
 sunPortalNetmailIMAPServerName 126
 sunPortalNetmailIMAPUserid 126
 sunPortalNetmailSMTPServerName 126
 tagModules 126
 tags 125
 timeLeft 126
 timezone 127
 timezoneList 126
 uid 126

W

Web Services Description Language
 see WSDL 120
WSDL
 document 120
 parsing 121
 specification 121

X

XML channel display profile definition 187
XMLProvider 25, 83, 84, 85
XMLProvider display profile properties
 cookiesToForwardAll 85
 cookiesToForwardList 85
 inputEncoding 85
 introduction 84
 timeout 85
 url 84
 urlScrapperRulesetID 84
 xslFileName 84