



# Sun Java System Portal Server 7.1 Technical Reference



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

Part No: 819-7095-10  
March 2007

Copyright 2007 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

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

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

This distribution may include materials developed by third parties.

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

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

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

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

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

---

Copyright 2007 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

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

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

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

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

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

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

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

# Contents

---

<b>Preface</b> .....	29
<b>Part I Attributes</b> .....	37
<b>1 Desktop Service</b> .....	39
Introduction .....	39
Desktop Global Attributes .....	39
Desktop Dynamic Attributes .....	42
<b>2 Rewriter Service</b> .....	47
Rewriter Tab Tasks .....	47
▼ To Enable Rewriting of All URIs .....	47
▼ To Specify the URIs Not to Rewrite .....	48
▼ To Map a URI to a Ruleset .....	48
▼ To Specify MIME Mappings .....	48
▼ To Configure Rewriter Proxies .....	48
Rewriter Rulesets .....	49
▼ To Download a Ruleset .....	49
▼ To Upload a Ruleset .....	49
▼ To Delete a Ruleset .....	49
<b>3 Search Attributes: Server</b> .....	51
Introduction .....	51
<b>4 Search Attributes: Robot</b> .....	53
Status and Control .....	53

Sites .....	55
Filters .....	58
Properties .....	60
Indexing .....	66
Simulator .....	67
Site Probe .....	67
<b>5 Search Attributes: Database .....</b>	<b>69</b>
Management .....	69
Import Agents .....	70
Resource Descriptions .....	72
Schema .....	74
Analysis .....	76
<b>6 Search Attributes: Categories .....</b>	<b>79</b>
Overview .....	79
Manage Categories .....	79
Manage Classification Rules .....	80
<b>7 Search Attributes: Scheduling .....</b>	<b>83</b>
Scheduling Robot .....	83
Scheduling Import Agents .....	84
Scheduling Autoclassify .....	84
<b>8 Search Attributes: Reports .....</b>	<b>87</b>
Introduction .....	87
Excluded URLs .....	87
Advanced Robot Reports .....	88
Log Files .....	88
Popular Searches .....	89
Search Server Log Settings .....	89
<b>9 Subscriptions Attributes .....</b>	<b>91</b>
Root Level Attributes .....	91

---

Organization Attributes .....	92
User Attributes .....	94
Scheduling Profiler Attributes .....	98
<b>Part II Configuration Files .....</b>	<b>99</b>
<b>10 Desktop Configuration Properties File .....</b>	<b>101</b>
Overview .....	101
Parameters .....	101
<b>11 Search Configuration Properties File .....</b>	<b>105</b>
Overview .....	105
Parameters .....	105
<b>12 XML and Schema Files .....</b>	<b>109</b>
Service Management Services .....	109
<b>Part III Display Profile .....</b>	<b>111</b>
<b>13 Introduction to Display Profile .....</b>	<b>113</b>
What is Display Profile? .....	113
Administering the Display Profile .....	114
<b>14 Display Profile Document .....</b>	<b>115</b>
Document Structure .....	115
How are the Display Profile XML Documents Stored? .....	116
Types of Display Profile Documents .....	116
<b>15 Display Profile Objects .....</b>	<b>119</b>
Introduction .....	119
Channel Object .....	119
Provider Object .....	120

Property object .....	121
Object Lookup .....	122
▼ To Perform Object Lookup: .....	122
<b>16 Document Priorities .....</b>	<b>123</b>
Overview .....	123
Examples .....	124
Example 1 .....	124
Example 2 .....	125
Summary .....	126
<b>17 Merge Semantics .....</b>	<b>127</b>
Introduction .....	127
Process of Merging .....	129
▼ To Determine the User Display Profile: .....	129
Types of Merge .....	130
Overview .....	130
Examples .....	131
Merge Locking .....	134
Examples .....	134
<b>18 Display Profile Properties: Overview .....</b>	<b>137</b>
Introduction .....	137
Display Profile Properties .....	138
Global .....	138
Provider .....	138
Channel .....	139
Container .....	139
Display Profile Property Types .....	139
<b>19 Display Profile Properties: Global Properties .....</b>	<b>141</b>
Example for Global Properties Sample .....	141

---

<b>20</b>	<b>Display Profile Properties: Container Provider Properties</b> .....	143
	Introduction .....	143
	Available and Selected List .....	144
	Common Properties for Table Container .....	145
	Common Properties for Tab Container .....	147
	Other Container Properties .....	148
<b>21</b>	<b>Display Profile Properties: Leaf Building-Block Provider Properties</b> .....	149
	JSPPProvider .....	149
	URLScrapperProvider .....	150
	XMLProvider .....	151
<b>22</b>	<b>Display Profile Properties: Service Provider Properties</b> .....	155
	SearchProvider .....	155
	DiscussionsProvider .....	156
	SubscriptionsProvider .....	158
<b>23</b>	<b>Display Profile Properties: Content Provider Properties</b> .....	159
	AddressBookProvider, LotusNotesAddressBookProvider, and MSEExchangeAddressBookProvider .....	159
	AppProvider .....	160
	BookmarkProvider .....	161
	CalendarProvider, LotusNotesCalendarProvider, and MSEExchangeCalendarProvider ..	161
	IMProvider .....	162
	LoginProvider .....	163
	NotesProvider .....	164
	SimpleWebServiceProvider .....	164
	SimpleWebServiceConfigurableProvider .....	165
<b>24</b>	<b>Display Profile Common Properties for Leaf Providers</b> .....	167
	Introduction .....	167
	Common Properties .....	167
	Other Leaf Provider Display Profile Properties .....	169
	authlessState .....	169

encoderClassName .....	169
ConditionalProperties .....	170
<b>25 Display Profile Channel Properties .....</b>	<b>171</b>
Display Profile Channel Properties .....	171
<b>Part IV Desktop Java Server Pages .....</b>	<b>173</b>
<b>26 Overview of Desktop JavaServer Pages .....</b>	<b>175</b>
Introduction .....	175
Installation Location .....	175
The Desktop and JavaServer Pages .....	176
File Lookup Scenario .....	176
<b>27 JSPs in the default Directory .....</b>	<b>181</b>
AddressBookJSPProvider .....	182
CalendarJSPProvider .....	183
CommunityDiscussionProvider .....	183
CommunitySearchProvider .....	184
CommunitySubscriptionsProvider .....	185
DefaultChannel .....	186
DiscussionLite .....	186
DiscussionProvider .....	186
IFrameProvider .....	188
IMProvider .....	188
JSPContentContainer .....	189
JSPDynamicSingleContainer .....	189
JSPEditContainer .....	189
JSPFrameCustomTableContainerProvider .....	190
JSPLayoutContainer .....	191
JSPMenuContainerProvider .....	191
JSPNativeContainer .....	192
JSPProvider .....	192
JSPRenderingContainerProvider .....	192



---

JSPRenderingContentChannel .....	193
JSPRenderingEditContainer .....	193
JSPRenderingErrorChannel .....	193
JSPRenderingLayoutChannel .....	194
JSPRenderingProvider .....	194
JSPSingleContainerProvider .....	194
JSPSingleRenderingContainerProvider .....	195
JSPTabContainerProvider .....	195
JSPTabCustomTableContainerProvider .....	196
JSPTableContainerProvider .....	197
JSPs Within the default Directory .....	198
LotusNotesAddressBookJSPProvider .....	199
LotusNotesCalendarJSPProvider .....	200
LotusNotesMailJSPProvider .....	200
MAAddressBookJSPProvider .....	200
MAMailJSPProvider .....	201
MSExchangeAddressBookJSPProvider .....	202
MSExchangeCalendarJSPProvider .....	203
MSExchangeMailJSPProvider .....	203
MailJSPProvider .....	204
RenderingWrappingProvider .....	204
SampleSimpleWebService .....	204
SampleSimpleWebServiceConfigurable .....	205
SearchProvider .....	205
SimpleWebServiceConfigurableProvider .....	207
SimpleWebServiceProvider .....	207
SubscriptionsProvider .....	208
TabJSPEditContainer .....	208
UWCAddressBookJSPProvider .....	209
UWCCalendarJSPProvider .....	209
UWCMailJSPProvider .....	210
WirelessJSPClientConfigContainer .....	210
WirelessJSPContentChannel .....	210
WirelessJSPDesktopProvider .....	211
WirelessJSPLayoutChannel .....	212
vxml .....	213

---

<b>28</b>	<b>JSPs in the community_sample Directory</b> .....	215
	AccountPreferencesContainer .....	215
	SearchProvider .....	216
	DiscussionProvider .....	217
	PagePreferencesContainer .....	217
	JSPTableContainerProvider .....	218
	JSPEditContainer .....	219
	JSPMenuContainerProvider .....	219
	JSPs Within the community_sample Directory .....	220
<b>29</b>	<b>JSPs in the enterprise_sample Directory</b> .....	223
	ASCTabContainer .....	223
	AccountPreferencesContainer .....	224
	FullPageContainer .....	225
	JSPEditContainer .....	225
	JSPTableContainerProvider .....	225
	OnlineHelpContainer .....	226
	PagePreferencesContainer .....	226
	JSPs Within the enterprise_sample Directory .....	227
<b>30</b>	<b>JSPs in the developer_sample Directory</b> .....	229
	FrameTabContainer .....	230
	JSPContentContainer .....	230
	JSPCreateChannelContainer .....	231
	JSPCustomThemeContainer .....	231
	JSPDynamicSingleContainer .....	232
	JSPEditContainer .....	232
	JSPFrameCustomTableContainerProvider .....	232
	JSPLayoutContainer .....	233
	JSPPopupContainer .....	233
	JSPPresetThemeContainer .....	233
	JSPSingleContainer .....	234
	JSPTabContainer .....	234
	JSPTabCustomTableContainerProvider .....	234
	JSPTableContainerProvider .....	235

---

JSPs Within the developer_sample Directory .....	236
PredefinedFrontPageFramePanelContainerProvider .....	237
PredefinedFrontPageTabPanelContainerProvider .....	238
PredefinedSamplesFramePanelContainerProvider .....	239
PredefinedSamplesTabPanelContainerProvider .....	240
SampleJSP .....	241
SampleSimpleWebService .....	242
<b>31 JSPs in the developer_anonymous_sample Directory .....</b>	<b>243</b>
FrameTabContainer .....	243
JSPDynamicSingleContainer .....	244
JSPTabContainer .....	244
JSPTableContainerProvider .....	244
PredefinedFrontPageFramePanelContainerProvider .....	245
PredefinedFrontPageTabPanelContainerProvider .....	246
PredefinedSamplesFramePanelContainerProvider .....	246
PredefinedSamplesTabPanelContainerProvider .....	247
<b>32 JSPs in the welcome_sample Directory .....</b>	<b>249</b>
RedirectChannel .....	249
<b>Part V Rewriter .....</b>	<b>251</b>
<b>33 Overview of Rewriter .....</b>	<b>253</b>
Expanding Relative URLs to Absolute URLs .....	253
URLScrapperProvider Limitations .....	254
Prefixing the Gateway URL to an Existing URL .....	254
Attributes .....	255
<b>34 Supported URLs .....</b>	<b>257</b>
Supported URLs .....	257

---

<b>35</b>	<b>Defining Rewriter Rules and Rulesets</b> .....	259
	Overview .....	259
	Rules for HTML Content .....	260
	Attribute Rules for HTML Content .....	260
	JavaScript Token Rules for HTML Content .....	261
	Form Rules for HTML Content .....	262
	Applet Rules for HTML Content .....	262
	Rules for JavaScript Content .....	263
	JavaScript Variables .....	263
	JavaScript Function Parameters .....	265
	Rules for XML Content .....	266
	Tag Text in XML .....	266
	Attributes in XML .....	267
<b>Part VI</b>	<b>Sample Portal</b> .....	269
<b>36</b>	<b>Understanding the Developer Sample</b> .....	271
	Overview .....	271
	Developer Sample Installation Directories .....	272
	<i>PortalServer-base/samples/desktop</i> .....	273
	Sample Desktops .....	273
<b>37</b>	<b>JSP-Based Desktop</b> .....	275
	JSPTabContainer .....	275
	Sample Desktop .....	275
	JSPTabContainer Architecture .....	278
	JSP Files Used by JSPTabContainer .....	279
	JSPTableContainer .....	280
	Sample Desktop .....	280
	JSPTableContainer Architecture .....	282
	JSP Files Used by JSPTableContainer .....	283
<b>38</b>	<b>Frame-Based Desktop</b> .....	285
	Sample Desktop .....	285

---

Default Layout .....	285
Default Actions .....	286
Default Display Profile Settings .....	287
FrameTabContainer Architecture .....	288
JSP Files Used by FrameTabContainer .....	289
<b>39 Internally Used Containers .....</b>	<b>291</b>
Internally Used Containers .....	291
<b>40 Global Themes .....</b>	<b>293</b>
What is a Theme? .....	293
GlobalThemes Display Profile Definition .....	293
Theme Properties .....	294
Glossary of Terms .....	296
<b>Part VII Search Engine Robot .....</b>	<b>299</b>
<b>41 Overview of Search Engine Robot .....</b>	<b>301</b>
Overview .....	301
<b>42 Process Parameters .....</b>	<b>303</b>
User Modifiable Parameters in robot.conf File .....	303
<b>43 The Filtering Process .....</b>	<b>309</b>
Overview .....	309
Stages in the Filter Process .....	310
Filter Syntax .....	311
Filter Directives .....	312
Writing or Modifying a Filter .....	312
<b>44 Robot Application Functions - Sources and Destinations .....</b>	<b>315</b>
Introduction .....	315
Setup Stage .....	315

MetaData Filtering Stage .....	316
Data Stage .....	316
Enumeration, Generation, and Shutdown Stages .....	317
<b>45 Robot Application Functions - Enable Parameter .....</b>	<b>319</b>
Example .....	319
<b>46 Robot Application Functions - Setup Functions .....</b>	<b>321</b>
filterrules-setup .....	321
Parameters .....	321
Example .....	321
setup-regex-cache .....	322
Parameters .....	322
Example .....	322
setup-type-by-extension .....	322
Parameters .....	322
Example .....	322
<b>47 Robot Application Functions - Filtering Functions .....</b>	<b>323</b>
Introduction .....	323
filter-by-exact .....	323
Parameters .....	324
Example .....	324
filter-by-max .....	324
Parameters .....	324
Example .....	324
filter-by-md5 .....	325
Parameters .....	325
Example .....	325
filter-by-prefix .....	325
Parameters .....	325
Example .....	326
filter-by-regex .....	326
Parameters .....	326

---

Example .....	326
filterrules-process .....	326
Parameters .....	326
Example .....	326
<b>48 Robot Application Functions - Filtering Support Functions .....</b>	<b>327</b>
Introduction .....	327
assign-source .....	327
Parameters .....	327
Example .....	328
assign-type-by-extension .....	328
Parameters .....	328
Example .....	328
clear-source .....	328
Parameters .....	328
Example .....	329
convert-to-html .....	329
Parameters .....	329
Example .....	329
copy-attribute .....	329
Parameters .....	330
Example .....	330
generate-by-exact .....	330
Parameters .....	330
Example .....	330
generate-by-prefix .....	331
Parameters .....	331
Example .....	331
generate-by-regex .....	331
Parameters .....	331
Example .....	332
generate-md5 .....	332
Parameters .....	332
Example .....	332
generate-rd-expires .....	332

Parameters .....	332
Example .....	333
generate-rd-last-modified .....	333
Parameters .....	333
Example .....	333
rename-attribute .....	333
Parameters .....	333
Example .....	333
<b>49 Robot Application Functions - Enumeration Functions .....</b>	<b>335</b>
Introduction .....	335
enumerate-urls .....	335
Parameters .....	335
Example .....	336
enumerate-urls-from-text .....	336
Parameters .....	336
Example .....	336
<b>50 Robot Application Functions - Generation Functions .....</b>	<b>337</b>
Introduction .....	337
extract-full-text .....	337
Parameters .....	337
Example .....	338
extract-html-meta .....	338
Parameters .....	338
Example .....	338
extract-html-text .....	338
Parameters .....	338
Example .....	339
extract-html-toc .....	339
Parameters .....	339
Example .....	339
extract-source .....	340
Parameters .....	340
Example .....	340



---

harvest-summarizer .....	340
Parameters .....	340
Example .....	340
<b>51 Robot Application Functions - Shutdown Functions .....</b>	<b>341</b>
Introduction .....	341
filterrules-shutdown .....	341
Parameters .....	341
Example .....	341
<b>Part VIII Desktop Tag Library .....</b>	<b>343</b>
<b>52 Overview of Desktop Tag Library .....</b>	<b>345</b>
Introduction .....	345
Types of Tags .....	345
Desktop Template Tags .....	345
JavaServer Pages Tags .....	346
Search Tags .....	349
Desktop Tag Library Hierarchy .....	350
Tag Library Descriptors .....	350
<b>53 Context Setup Tags .....</b>	<b>353</b>
Context Setup Tags .....	353
<b>54 Validator Tags .....</b>	<b>355</b>
Validator Tags .....	355
<b>55 desktop.tld Tags .....</b>	<b>357</b>
desktop.tld Tags With Attributes .....	357
<b>56 desktopProviderContext.tld Tags .....</b>	<b>361</b>
desktopProviderContext.tld Normal Tags with Attributes .....	361

<b>57</b>	<b>desktopContainerProviderContext.tld Tags</b> .....	369
	desktopContainerProviderContext.tld Tags with Attributes .....	369
<b>58</b>	<b>desktopTab.tld Tags</b> .....	371
	desktopTab.tld Normal Tags with Attributes .....	371
<b>59</b>	<b>desktopTable.tld Tags</b> .....	373
	desktopTable.tld Normal Tags with Attributes .....	373
<b>60</b>	<b>desktopTheme.tld Tags</b> .....	375
	desktopTheme.tld Normal Tags with Attributes .....	375
<b>61</b>	<b>searchContext tag in search.tld</b> .....	379
	search.tld Normal Tags with Attributes for the searchContext .....	379
<b>62</b>	<b>Pre-Search Tags in search.tld</b> .....	381
	search.tld Tags for Pre-Search .....	381
<b>63</b>	<b>Execute Search tag in search.tld</b> .....	385
	desktop.tld Normal Tags with Attributes .....	385
<b>64</b>	<b>Post Search Tags in search.tld</b> .....	387
	desktop.tld Normal Tags with Attributes .....	387
<b>65</b>	<b>Miscellaneous Tags in search.tld</b> .....	391
	search.tld Normal Tags with Attributes .....	391
<b>66</b>	<b>Desktop Template Common Tags</b> .....	393
	Desktop Template Common Tags .....	393

<b>67</b>	<b>Provider-Specific Desktop Template Tags</b> .....	399
	AddressBookProvider .....	399
	AppProvider .....	400
	CalendarProvider .....	401
	LoginProvider .....	410
<b>68</b>	<b>Instant Messaging Tags</b> .....	411
	Tags in im.tld .....	411
<b>Part IX</b>	<b>Desktop Templates</b> .....	413
<b>69</b>	<b>Overview of Desktop Templates</b> .....	415
	Introduction .....	415
	Installation Location .....	415
	The Desktop and Template Files .....	416
	File Lookup Scenario .....	416
<b>70</b>	<b>Desktop Templates in the default Directory</b> .....	421
	AddressBookProvider .....	421
	AppProvider .....	422
	BookmarkProvider .....	422
	CalendarProvider .....	423
	error .....	425
	LoginProvider .....	426
	default .....	426
<b>71</b>	<b>Desktop Templates in the Developer Sample Directory</b> .....	429
	MyFrontPageTemplatePanelContainer .....	429
	PredefinedFrontPageTemplatePanelContainerProvider .....	429
	PredefinedSamplesTemplatePanelContainerProvider .....	431
	SamplesTemplatePanelContainer .....	432
	ToolsTemplatePanelContainer .....	433
	developersample .....	433

<b>72 Portal Server : Java Enterprise System Monitoring Framework Integration</b> .....	435
JESMF .....	435
JESMC .....	436
Common Agent Container .....	437
Portal Server:JESMF Detailed Mapping .....	437
CMM_InstalledProduct:PortalServer .....	438
CMM_ApplicationSystem:PortalServerInstance .....	440
Portal Server:Desktop Detailed Mapping .....	441
CMM_LogicalComponent:Desktop .....	441
CMM_Service:Desktop .....	442
CMM_ServiceAccessURI:DesktopServiceAccessURI .....	443
CMM_LogicalComponent:Provider .....	444
CMM_Service:- Content, Edit, Process, Logout, PreLogin, LocalAuth, Exception .....	444
CMM_Service:- Content, Edit, Process .....	445
CMM_ThreadPool:- GetterPool, CallerPool .....	446
JESMF Extensions for Portal Server .....	447
CMM_PSServiceStats .....	447
CMM_PSServiceSetting .....	447
Installing and Configuring Portal Server:JESMF Integration .....	448
Enabling or Disabling Portal Server:JESMF Integration .....	449
▼ To Enable Portal Server:JESMF Integration: .....	449
▼ To Disable Portal Server:JESMF Integration: .....	449
<b>Index</b> .....	451

# Figures

---

FIGURE 17-1	Example of Merge Process .....	129
FIGURE 36-1	Desktop Components .....	272
FIGURE 37-1	Sample Desktop Based on JSPTabContainer .....	276
FIGURE 37-2	JSPTabContainer Architecture .....	279
FIGURE 37-3	Sample Desktop Based on JSPTableContainer .....	280
FIGURE 37-4	JSPTableContainer Architecture .....	282
FIGURE 38-1	Sample Desktop Based on FrameTabContainer .....	286
FIGURE 38-2	FrameTabContainer Architecture .....	288
FIGURE 41-1	How the Robot Works .....	302



# Tables

---

TABLE 1-1	Desktop Service - Global Attributes .....	40
TABLE 1-2	Desktop Service - Dynamic Attributes .....	43
TABLE 3-1	Server Settings Attributes .....	51
TABLE 4-1	Robot Overview Attributes .....	54
TABLE 4-2	Manage Sites Attributes .....	55
TABLE 4-3	New Robot Site Attributes .....	56
TABLE 4-4	Edit a Site Attributes .....	56
TABLE 4-5	Filter Definition and Behavior Attributes .....	59
TABLE 4-6	Edit a Filter Attributes .....	59
TABLE 4-7	Manage Properties Attributes .....	60
TABLE 4-8	Robot Simulator Attributes .....	67
TABLE 4-9	Robot Site Probe Attributes .....	67
TABLE 5-1	Database Management Attributes .....	70
TABLE 5-2	Database Import Agent Attributes .....	70
TABLE 5-3	Resource Descriptions Attributes .....	72
TABLE 5-4	Database RD Editable Attributes .....	73
TABLE 5-5	Search Schema Attributes .....	74
TABLE 5-6	Edit Search Schema Attribute Attributes .....	75
TABLE 5-7	Database Analysis Attributes .....	76
TABLE 6-1	Manage Categories Attributes .....	80
TABLE 6-2	Edit Category Attributes .....	80
TABLE 6-3	Classification Rules Attributes .....	81
TABLE 7-1	Robot Schedule Attributes .....	83
TABLE 7-2	Database Import Schedule Attributes .....	84
TABLE 7-3	Scheduling Autoclassify Attributes .....	85
TABLE 8-1	Reports Excluded URLs Attributes .....	88
TABLE 8-2	Robot Report Files Attributes .....	88
TABLE 8-3	Reports —> Log Files Attributes .....	89

---

TABLE 8-4	Popular Searches Attributes .....	89
TABLE 8-5	Search Server Log Settings Attributes .....	89
TABLE 9-1	Subscriptions Service - Root Level Attributes .....	91
TABLE 9-2	Subscriptions Service — Organization Attributes .....	92
TABLE 9-3	Subscriptions Service - User Attributes .....	94
TABLE 9-4	Attributes for Start Profiler .....	98
TABLE 9-5	Attributes for Stop Profiler .....	98
TABLE 18-1	Display Profile General Property Types .....	139
TABLE 20-1	Table Container Properties .....	145
TABLE 36-1	Sun Java System Portal Server Sample Desktop Containers .....	273
TABLE 39-1	Internally Used Containers .....	291
TABLE 40-1	Glossary of Terms .....	296
TABLE 42-1	User Modifiable Parameters in robot . conf File .....	303
TABLE 43-1	Common Metadata Types .....	310
TABLE 44-1	Sources Available to the RAFs at the MetaData Phase .....	316
TABLE 44-2	Sources Available to the RAFs at the Data Phase .....	317
TABLE 53-1	Context Setup Tags .....	353
TABLE 55-1	desktop.tld Tags with Attributes .....	357
TABLE 56-1	desktopProviderContext.tld Tags with Attributes .....	361
TABLE 57-1	desktopContainerProviderContext.tld Tags with Attributes .....	369
TABLE 58-1	desktopTab.tld Tags with Attributes .....	371
TABLE 59-1	desktopTable.tld Tags with Attributes .....	373
TABLE 60-1	desktopTheme.tld Tags with Attributes .....	375
TABLE 61-1	search.tld Tags with Attributes for the searchContext .....	379
TABLE 62-1	search.tld Tags for Pre-Search .....	381
TABLE 63-1	desktop.tld Normal Tags with Attributes .....	385
TABLE 64-1	desktop.tld Normal Tags with Attributes .....	387
TABLE 65-1	search.tld Normal Tags with Attributes .....	391
TABLE 68-1	Tags in im.tld .....	411
TABLE 72-1	CMM_InstalledProduct:PortalServer .....	438
TABLE 72-2	CMM_ApplicationSystem:PortalServerInstance .....	440
TABLE 72-3	CMM_LogicalComponent:Desktop .....	441
TABLE 72-4	CMM_Service:Desktop .....	442
TABLE 72-5	CMM_ServiceAccessURI:DesktopServiceAccessURI .....	443
TABLE 72-6	CMM_LogicalComponent:Provider .....	444
TABLE 72-7	CMM_Service:- Content, Edit, Process, Logout, PreLogin, LocalAuth,	



---

	Exception .....	444
TABLE 72-8	CMM_Service:- Content, Edit, Process .....	445
TABLE 72-9	CMM_ThreadPool:- GetterPool, CallerPool .....	446
TABLE 72-10	CMM_PSRollingAvgServiceStats .....	447
TABLE 72-11	CMM_PSRollingAvgServiceStats .....	447



# Examples

---

EXAMPLE 15-1	Example Channel Object XML Syntax .....	119
EXAMPLE 15-2	Example Container Object XML Syntax .....	120
EXAMPLE 15-3	Example Provider Object XML Syntax .....	121
EXAMPLE 16-1	Display Profile Document for the Organization (dc=acme.com) .....	124
EXAMPLE 16-2	Global Display Profile Document .....	125
EXAMPLE 17-1	Display Profile for the Admin Role .....	133
EXAMPLE 19-1	Global Properties Sample in the Display Profile dp-org.xml File .....	141
EXAMPLE 24-1	<ConditionalProperties> Tag Usage Sample .....	170
EXAMPLE 26-1	JSP Lookup Scenario .....	176
EXAMPLE 26-2	JSP Lookup Scenario (no clientPath) .....	178
EXAMPLE 43-1	Enumeration File Syntax .....	311
EXAMPLE 69-1	Template File Lookup Scenario .....	416
EXAMPLE 69-2	Template File Lookup Scenario (no clientPath) .....	418



# Preface

---

The Sun™ Java™ System Portal Server 7.1 Technical Reference guide provides detailed information on the Sun Java System Portal Server 7.1 technical concepts (such as Display Profile, Rewriter), command line utilities, tag libraries (in the software), and files (such as templates and JSPs). Use this book in conjunction with the other books in the Portal Server 7.1 documentation set (see “Books in This Documentation Set” on page 33)

## Who Should Use This Book

The *Sun Java System Portal Server 7.1 Technical Reference* is intended for use by administrators and individuals responsible for:

- Administering and configuring the Portal Server 7.1 software
- Customizing pieces of the software

Readers should have admin-level permissions to the software. This guide is not intended for end users.

Administrators and individuals using this guide must be familiar with basic Solaris™ administrative procedures and must understand the following technologies:

- Lightweight Directory Access Protocol (LDAP)
- Java technology
- JavaServer Pages (JSP) technology
- Hypertext Transfer Protocol (HTTP)
- Hypertext Markup Language (HTML)
- Extensible Markup Language (XML)

## Before You Read This Book

Portal Server 7.1 is a component of Sun Java Enterprise System, a software infrastructure that supports enterprise applications distributed across a network or Internet environment. You should be familiar with the documentation provided with Sun Java Enterprise System, which can be accessed online at <http://docs.sun.com><sup>SM</sup>.

Because the Portal Server 7.1 software components work together with Sun Java System Sun Java System Access Manager for user, service, and policy management and authentication,

single sign-on, and logging services, you should be familiar with the documentation provided with that product. Sun Java System Access Manager software documentation can be accessed online at <http://docs.sun.com/>.

Because Sun Java System Sun Java System Directory Server is used as the data store for primary configuration information and user profile data in a Portal Server 7.1 deployment, you should be familiar with the documentation provided with that product. Sun Java System Directory Server software documentation can be accessed online at <http://docs.sun.com>.

## How This Book Is Organized

The first chapter of this book provides an overview of the entire Portal Server 7.1 product. The rest of the book is divided into parts; each part includes multiple chapters detailing a concept. The following table summarizes various parts of this book.

TABLE P-1 How This Book Is Organized

Part Title	Part Description
Part I	Describes the various attributes in the Portal Server 7.1 software.
Part II	Describes properties in the properties file for the Desktop and Search service.
Part III	Describes the display profile document, objects, priorities, merge semantics, and properties.
Part IV	Describes the various JSPs installed in the default and sampleportal directories.
Part V	Describes the supported URLs and provide background information on defining rules and rulesets.
Part VI	Describes the software Sample Portal and its associated containers and themes. The sample portal is an authentication-less desktop that consists of sample containers, channels, portlets, services, and templates which demonstrates the Portal Server 7.1 software's capabilities.
Part VII	Describes the Search engine robot and its application functions. The Search Engine robot is included to discover, convert, and summarize document resources. Users can then locate or browse content contained in the database.
Part VIII	Describes the tag libraries used by the Desktop and Search service.
Part IX	Describes the Desktop templates installed in the default and sampleportal directories.

## Conventions Used in This Book

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

### Typographic Conventions

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

TABLE P-2 Typographic Conventions

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

### Symbols

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

TABLE P-3 Symbol Conventions

Symbol	Description	Example	Meaning
[ ]	Contains optional command options.	<code>ls [-l]</code>	The <code>-l</code> option is not required.
{   }	Contains a set of choices for a required command option.	<code>-d {y n}</code>	The <code>-d</code> option requires that you use either the <code>y</code> argument or the <code>n</code> argument.
-	Joins simultaneous multiple keystrokes.	Control-A	Press the Control key while you press the A key.

TABLE P-3 Symbol Conventions (Continued)

Symbol	Description	Example	Meaning
+	Joins consecutive multiple keystrokes.	Ctrl+A+N	Press the Control key, release it, and then press the subsequent keys.
>	Indicates menu item selection in a graphical user interface.	File > New > Templates	From the File menu, choose New. From the New submenu, choose Templates.

## Default Paths and File Names

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

TABLE P-4 Default Paths and File Names

Term	Description
<i>PortalServer-base</i>	Represents the base installation directory for Sun Java System Portal Server 7.1 software. The Portal Server 7.1 software default base installation and product directory depends on your specific platform:  Solaris systems: <code>/opt/SUNWportal</code>  Linux systems: <code>/opt/sun/porta1</code>
<i>AccessManager-base</i>	Represents the base installation directory for Sun Java System Access Manager software. The Sun Java System Access Manager software default base installation and product directory depends on your specific platform:  Solaris systems: <code>/opt/SUNWam</code>
<i>DirectoryServer-base</i>	Represents the base installation directory for Sun Java System Directory Server software. The Sun Java System Directory Server software default base installation is <code>/opt/sun/ds6</code> .
<i>ApplicationServer-base</i>	Represents the base installation directory for Sun One Application Server software. The Sun One Application Server software default base installation is <code>/opt/SUNWappserver</code> .
<i>WebServer-base</i>	Represents the base installation directory for Sun Java System Web Server software. The Sun Java System Web Server software default base installation is <code>/opt/SUNWbsvr7</code> .
<i>PortalServer-ContentFiles</i>	Represents the Portal Server Data directory where JSPs, templates and property files, and tag libraries are installed. By default, this is <code>/var/opt/sun/porta1</code> for Linux.
<i>PortalServer-SEdb</i>	Represents the Portal Server 7.1 software Search Engine database. By default, this is <code>/var/opt/SUNWporta1/searchservers/search1/db</code> .



## Shell Prompts

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

TABLE P-5 Shell Prompts

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

## Related Documentation

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

## Books in This Documentation Set

The following table summarizes the books included in the Portal Server 7.1 core application documentation set.

TABLE P-6 Books in This Documentation Set

Book Title	Description
<i>Sun Java System Portal Server 7.1 Administration Guide</i>	Describes how to administer the Portal Server 7.1 software using the administration console and the command line.
<i>Sun Java System Portal Server Secure Remote Access 7.1 Administration Guide</i>	Describes how to administer Portal Server 7.1 Secure Remote Access software.
<i>Sun Java System Portal Server 7.1 Configuration Guide</i>	Describes how to configure the Portal Server 7.1 software Desktop.
<i>Sun Java System Portal Server 7.1 Developer's Guide</i>	Describes how to extend the Portal Server 7.1 software APIs.
<i>Sun Java System Portal Server 7.1 Deployment Planning Guide</i>	Describes how to plan for and deploy Portal Server 7.1 software.

## Other Server Documentation

Other server documentation collections are:

- Sun Java System Directory Server documentation: <http://docs.sun.com>
- Sun Java System Web Server documentation: <http://docs.sun.com>
- Sun One Application Server documentation: <http://docs.sun.com>
- Sun Java System Access Manager documentation: <http://docs.sun.com>

## Accessing Sun Resources Online

Access these URLs for product downloads, professional services, patches and support, and additional developer information.

- Download Center <http://www.sun.com>
- Professional Services: <http://www.sun.com>
- Sun Enterprise Services, Solaris Patches, and Support: <http://www.sun.com>
- Developer Information: <http://developers.sun.com>

## Contacting Sun Technical Support

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

## Related Third-Party Web Site References

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

## Sun Welcomes Your Comments

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

To share your comments, go to <http://docs.sun.com> and click Send Comments. In the online form, provide the document title and part number. The part number is a seven-digit or nine-digit number that can be found on the title page of the book or at the top of the document. For example, the title of this book is *Sun Java System Portal Server 7.1 Technical Reference* and the part number is 819-7095.



PART I

Attributes

- Chapter 1
- Chapter 2
- Chapter 3
- Chapter 4
- Chapter 5
- Chapter 6
- Chapter 7
- Chapter 8
- Chapter 9



# Desktop Service

---

This chapter contains the following sections:

- [“Introduction” on page 39](#)
- [“Desktop Global Attributes” on page 39](#)
- [“Desktop Dynamic Attributes” on page 42](#)

## Introduction

The Desktop Service consists of global and dynamic attributes. The values applied to the global attributes are applied across the Sun Java System Sun Java System Access Manager configuration and are inherited by every configured organization. They cannot be applied directly to roles or organizations as the goal of global attributes is to customize the Sun Java System Access Manager application. Values applied to the dynamic attributes are assigned to a role or organization. When the role is assigned to a user or a user is created in an organization, the dynamic attribute then becomes a characteristic of the user.

## Desktop Global Attributes

This section describes the global attributes for the Desktop Service.

The table below describes the desktop service global attributes and their description.

TABLE 1-1 Desktop Service - Global Attributes

Attribute	Default Value	Description
XML Parsing Validation	Enable (selected)	Specifies whether to enforce validation while parsing the display profile XML document. Not selecting this attribute can improve system performance. However, this can potentially introduce corruption in the display profile document because the resulting XML document might include some fragments that do not conform to the DTD (Document Type Definition).
Federation	Disable (not selected)	Enables Identity Federation so that a user can associate, connect or bind multiple internet service providers' local identities, enabling them to have one network identity.
Hosted Provider ID	None	Specifies the unique identifier of the host that provides the network identity of a user.
Session Reap Interval	1800	Defines in seconds the time interval between checks for removing inactive client sessions.
Session Idle Time	3600	Specifies the maximum number of seconds a client session can be idle before it is considered inactive. If a session is idle for more than this value, it is made a candidate for session reaping and can be removed the next time the client session times out.
Maximum number of Sessions	1000000	Number of authless users can be supported by the Portal Server deployment.



TABLE 1-1 Desktop Service - Global Attributes (Continued)

Attribute	Default Value	Description
DisplayProfile Priority	<p>The default value depends on the type of installation performed. If the sample portal is installed, the DisplayProfile contains the definitions for the built-in providers (the basic providers of Portal Server), such as bookmark and notes. If the sample portal was not installed, the global DisplayProfile value is blank or zero.</p>	<p>Displays several controls for manipulating the global display profile, an XML document that defines the container management, channel attributes, and provider definitions for the organization. The controls include:</p> <ul style="list-style-type: none"> <li>■ <b>Disable Authentication-less Access for Federated Users</b>--Prevents a user with a federated network identity to access the portal without entering a user name and password.</li> <li>■ <b>Upload XML</b>--Allows you to upload an XML file containing display profile information to the Portal Server.</li> <li>■ <b>Download XML</b>--Allows you to download the display profile to your local drive.</li> <li>■ <b>Parent and Edit Container Management</b>--Provides a graphical user interface to manage container channels and channels without the need to edit the XML file. These links are not attributes. Selecting these links allows you to manipulate the display profile. Display profile elements defined in the global display profile are inherited by all users on the system, regardless of the organization or role to which they belong.</li> </ul>

TABLE 1-1 Desktop Service - Global Attributes (Continued)

Attribute	Default Value	Description
Anonymous Desktop	Enable (selected)	<p>If this value is Enabled, it lists the valid UID's and their passwords for the Anonymous Desktop. Displays several controls for configuring authentication-less configuration of the portal desktop. The controls are:</p> <ul style="list-style-type: none"> <li>■ Add-Click the Add button to add a User DN (Distinguished Name) to the existing list.</li> <li>■ Delete-From the list of available DNs, select the User DN. Click the Delete button to delete the selected User DN from the list.</li> <li>■ Set as Default-Select a User DN from the list by selecting the corresponding checkbox. Click the set as Default button to make the selected User DN as a default.</li> </ul>

## Desktop Dynamic Attributes

This section describes the dynamic attributes for the Desktop Service.

The table below describes the desktop service dynamic attributes and their description.

TABLE 1-2 Desktop Service - Dynamic Attributes

Attribute	Default Value	Description
COS Priority	Highest	<p>Sets the conflict resolution level for the Desktop service template used to resolve conflicts when multiple Desktop templates are merged. There are seven conflict resolution settings available ranging from Highest to Lowest.</p> <p>Do not confuse this setting with the display profile document priority. The display profile document priority is a numeric value that is set in the XML file with the priority= syntax tag. When a merge occurs, it starts with the lowest display profile priority document (lowest number) and proceeds in increasing priority number, until it arrives at the user (base), the highest priority display profile.</p> <p>When an attribute conflict occurs, the attribute on the template set with the highest conflict resolution level is returned.</p>
Parent Container	DefaultChannel	Identifies which default channel is rendered when the Desktop is called with an unspecified provider.
Edit Container	JSPEditContainer	<p>Specifies which default edit channel to use to wrap the content when one is not specified in the URL. When a channel is edited, an “Edit” request URL is sent to the Desktop Servlet. The URL generated for the “Edit” of each of the channels inside a container depends on the property “editContainerName” defined in the display profile.</p> <p>If you have migrated containers from iPlanet™ Portal Server 3.0, you must specify the default edit channel with which to wrap the content using this attribute because the URL format has changed.</p>

TABLE 1-2 Desktop Service - Dynamic Attributes (Continued)

Attribute	Default Value	Description
Desktop Type	default	<p>Retrieves template files for the specified Desktop type when different Desktop configurations are needed and when different sets of templates and JSPs are required for those configurations.</p> <p>The Desktop type attribute of the Desktop service is a comma-separated string type, that the Portal Desktop uses as an ordered 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 subdirectory 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.</p> <p>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 <code>sampleportal</code>, the target template will be searched in the <code>sampleportal</code> subdirectory, then the default subdirectory. By default, if the sample portal is installed, then the Desktop type attribute, <code>sunPortalDesktopType</code>, is set to <code>sampleportal</code>. If the sample portal is not installed, then the Desktop type attribute value is set to default.</p> <p>Most sites will not use the default Desktop type, as they will have different channels, different logo, different look and feel, and the like.</p>

TABLE 1-2 Desktop Service - Dynamic Attributes (Continued)

Attribute	Default Value	Description
Desktop Attributes	Show (selected)	<p>Specifies whether the Desktop Service attributes are displayed to the users associated with the role. This dynamic attribute is mainly used for role-based delegated administration. Values applied to this attribute are only in effect for a role.</p> <p>When the role is assigned to a user and the value of this attribute is false, users (usually delegated administrators) cannot see any Desktop Service attributes except the Channel and Container Management link when they navigate into all the roles within the organization.</p>
DisplayProfile Priority	<p>The default value depends on the type of installation performed. If the sample portal was installed, a sample display profile document is installed at the organization level that contains channels that display the built-in providers defined in the global display profile.</p>	<p>Displays several links for manipulating the display profile, an XML document that defines the container management, channel attributes, and provider definitions for this specific node (role, organization, suborganization). Links are:</p> <ul style="list-style-type: none"> <li>■ Edit XML--Allows you to edit the entire display profile XML file.</li> <li>■ Upload XML--Allows you to upload an XML file containing display profile information to the Portal Server.</li> <li>■ Download XML--Allows you to download the display profile to your local drive.</li> </ul> <p>These links are not attributes. Selecting these links allows you to manipulate the display profile.</p>



## Rewriter Service

---

The Rewriter service provides a Java class library for rewriting URI (Uniform Resource Identifier) references in Web languages such as HTML, JavaScript, and WML (Wireless Markup Language), and in HTTP Location headers (redirections). For example, the Rewriter rewrites relative URIs of image tags so that the browser requests the images from the appropriate server. The Rewriter also functions in the Portal Gateway. For example, the Rewriter rewrites HTML links to provide access to an internal server. The Rewriter service is available as a Rewriter Rulesets tab listed in the Secure Remote Access (SRA) feature in the Portal Server Administration console.

The topics in this chapter that describe the Rewriter Service are:

- [“Rewriter Tab Tasks” on page 47](#)
- [“Rewriter Rulesets” on page 49](#)

### Rewriter Tab Tasks

You can perform the following tasks from the Rewriter tab in the Gateway Profile menu bar:

#### ▼ To Enable Rewriting of All URIs

- 1 Select the Enable option in the Rewriting of All URIs attribute to enable the Gateway and to rewrite all URIs.
- 2 Click Save to record the change.

## ▼ To Specify the URIs Not to Rewrite

**Before You Begin** The prerequisite to map a URI to a Ruleset, to specify MIME (Multi Purpose Internet Mail Extensions) mappings, to specify the URIs not to rewrite, and to configure Rewriter proxies, specify one or more configuration attributes. See the <http://docs.sun.com/app/docs/coll/1303.1> for more information on configuration attributes.

- 1 Navigate to the URIs Not to Rewrite attribute, and specify the URI in the text box.
- 2 Click Add to add the URI in the URIs Not to Rewrite list.
- 3 Click Save to record the change.

## ▼ To Map a URI to a Ruleset

- 1 Navigate to the Map URIs to RuleSets attribute and map a URI to the appropriate RuleSet.
- 2 Click Save to record the change.

## ▼ To Specify MIME Mappings

- 1 Navigate to the Map Parser to MIME Types field, and map a Parser to the appropriate MIME type.
- 2 Click Save to record the change.

## ▼ To Configure Rewriter Proxies

- 1 Select the Enable option in the Rewriter Proxy attribute to enable the Rewriter proxy.
- 2 Specify the Rewriter Proxy value in the Rewriter Proxy List attribute text box.
- 3 Click Add to add the value in the Rewriter Proxy List.

---

**Note** – The Rewriter Proxy attribute is available in the Rewriter Proxy and Netlet Proxy option in the Deployment tab of Gateway Profile.

---

- 4 Click Save to record the change.



## Rewriter Rulesets

Rewriter Rulesets is a service, which is available as the Rewriter Rulesets tab in the Secure Remote Access tab of Portal Server Administration console. When you click the tab, the Rewriter Rulesets page is displayed. The Rewriter Rulesets page lists the available rulesets in the Gateway. Each ruleset is a file that defines how the contents in a Web page should be rewritten by the Rewriter attributes. You can download an existing ruleset, upload a new ruleset to the list, and delete an available ruleset from the list. You can perform these tasks as follows:

### ▼ To Download a Ruleset

**Before You Begin** The prerequisite to perform these tasks is to specify one or more configuration attributes. See the <http://docs.sun.com/app/docs/coll/1303.1> for more information on configuration attributes.

- 1 **Select a ruleset from the Gateway Rulesets list.**
- 2 **Click Download.**

You can save the ruleset or open and view in the default Web browser.

### ▼ To Upload a Ruleset

- 1 **Click Upload New.**  
The Upload New Ruleset page displays.
- 2 **Click Browse to choose the ruleset to upload.**
- 3 **Click Upload.**

### ▼ To Delete a Ruleset

- 1 **Select a ruleset from the Gateway Rulesets list.**
- 2 **Click Delete.**



# Search Attributes: Server

---

This chapter explains the attributes listed in the Search Servers page. The following section introduces the Search Server attributes and describes the attributes in Search Server Settings.

- [“Introduction” on page 51](#)

## Introduction

When you select Search Servers tab in the Portal Server console, the Search Servers page appears. This page lists the existing Search Servers. To view an existing Search Server, click on the search server name. The Search Server Settings page for the selected Search Server is displayed.

TABLE 3-1 Server Settings Attributes

Attribute	Default Value	Description
Root	<code>/var/opt/SUNWportal/ searchservers/search1</code>	Houses the configuration, log, database, and robot information files. Also it is the root directory for all of the search files that are generated and updated when conducting a search. This is not configurable.

TABLE 3-1 Server Settings Attributes (Continued)

Attribute	Default Value	Description
Document-Level Security	off	Controls who can access documents. When this setting is changed, the server must be restarted. Values: <ul style="list-style-type: none"><li>■ off (default) — all users have access to the RDs (Resource Description).</li><li>■ on — the ReadACL field in an RD is selected to see if the user asking for the RD has permission because the user is in an acceptable organization or role, or is an acceptable individual user. The ReadACL field is set in the Edit page for a Resource Description.</li></ul>
ID	ID of the selected search server	Displays the ID of the search server that you selected.
URL	<code>http://HOST:PORT/search server name</code>	Displays the URL of the search server that you selected.

# Search Attributes: Robot

---

This chapter explains about the attributes available in the Search Robot. The properties for the robot are quite complex. You can select the sites to be searched, check to see if a site is valid, define what types of documents should be picked up, and schedule when the searches take place.

This chapter contains the following sections:

- “Status and Control” on page 53
- “Sites” on page 55
- “Filters” on page 58
- “Properties” on page 60
- “Indexing” on page 66
- “Simulator” on page 67
- “Site Probe” on page 67

## Status and Control

The Robot Overview panel is where you can see what the robot is doing. If it is Off, Idle, Running, or Paused; and if it is Running, what progress it is making in the search since the panel is refreshed about every 30 seconds. The refresh rate is defined using the `robot - refresh` parameter in the `search.conf` file.

If the robot is Off, the buttons are Start and Clear Robot Database . The Start button is at the top and the Clear Robot Database button is at the bottom of the panel. If the robot is Running or Idle, the two buttons are Stop and Pause. If it is Paused, the two buttons are Stop and Resume. By selecting on any of the Attributes, you go to the Reports section where you can get a detailed up-to-the-minute report of that Attribute.

The table below lists the Robot Overview attributes and their description.

TABLE 4-1 Robot Overview Attributes

Attribute	Default Value	Description
The Robot is	Current activity	The Robot's state. Value can be Idle, Running, Paused, or Off
Last Updated at	Date and time last refreshed.	This page is refreshed to keep you aware of what progress the robot is making.
Starting Points	Number defined	Displays the sites that the robot crawls to generate resource descriptions. The robot does not index resources from disabled sites.
URL Pool	Number URLs waiting	Number of URLs (Uniform Resource Locator) yet to be investigated. When you begin a search, the starting point URLs are entered into the URL pool. As the search progresses, the robot discovers links to other URLs. These URLs get added to the pool. After all the URLs in the pool have been processed, the URL pool is empty and the robot is idle.
Extracting	Number connections per second	Number of resources looked at in a second.  Extracting is the process of discovering or locating resources, documents or hyperlinks to be included in the database and filtering out unwanted items.
Filtering	Number URLs rejected	Total number of URLs that are excluded.
Indexing	Number URLs per second	Number of resources or documents turned into a resource description in a second.  Indexing is the phase when all the information that has been gathered on a document is turned into a resource description for inclusion in the search database.
Excluded URLs	Number URLs excluded by filters	Number of URLs that did not meet the filtering criteria.

TABLE 4-1 Robot Overview Attributes (Continued)

Attribute	Default Value	Description
	Number URLs excluded by errors	Number of URLs where the robot encountered errors as file not found.
Resource Descriptions	Number RDs contributed	Number of resource descriptions added to the database.
	Number Bytes of RDs contributed	Number of bytes added to the database.
General Stats	Number URLs retrieved	Number of URLs retrieved during run.
	Number Bytes average size of RDs	Average number of bytes per resource description.
	Time in days, hours, minutes, and seconds running	The amount of time the robot has been running.

## Sites

When you click the Sites tab, the Manage Sites page is displayed. This page displays the list of Site Names and the Status of each site (enabled or disabled) that the robot crawls to generate resource descriptions. When you select the checkbox, the Delete, Enable, and Disable buttons become active. Select the Delete button to delete a selected site. You can enable or disable the selected site by clicking the Enable or Disable buttons. A disabled site is not searched when the robot is run.

The table below provides the attributes and their description in the Manage Sites page.

TABLE 4-2 Manage Sites Attributes

Attribute	Default Value	Description
Lock or cluster graphic	Status of site	Lock open means that the URL is accessible. The closed lock means that the site is a secure web server and uses SSL. The cluster means that the site is a domain.
Enabled/Disabled	Enabled	Choose to search this site or not when the robot is run.

You can create a new site, by clicking the New button. When you click the New button, the New Robot Site page appears. This page allows you to set up a new Robot site. The table below provides the attributes available in the New Robot Site page and their description.

TABLE 4-3 New Robot Site Attributes

Attribute	Default Value	Description
Type	URL	Select URL or Domain from the list box.
Site	Blank	If you have selected the Type as URL, enter the URL of the site you want to create. The URL format is: http://www.sesta.com  If you have selected the Type as Domain, enter the domain of the site you want to create. The Domain format is: *.sesta.com
Depth	10	You have a choice of 1 for this URL only, 2 for this URL and first links, 3 - 10, 100 or unlimited. The default value is set in the Robot —> Manage Properties page.
Destination Database	Use Internal Default	Select the database that you want to use from the list box showing the available databases.

Click on the Site name to navigate to the Edit a Site page. You can use this page to define the search site more completely. You can specify what type of server it is, redefine the depth of the search, and select what type of files to add to the search database. The attributes for URL and Domain sites are mostly the same. The additional column in this table shows which attributes are shared and which are unique.

You can verify the server name for the search site you entered. In the Server Group section, click the New button to add more servers to the server group. In the Starting Points section, click the New button to add more starting points. In the Filter Definition section, you can add or delete, exclude or include certain types of files as well as change the order the filters for these files are applied.

The table below provides the attributes and their description in the Edit a Site page.

TABLE 4-4 Edit a Site Attributes

Attribute	URL/Domain	Default Value	Description
Site Name	URL/D	Site entered - www.sesta.com	Name of the web site



TABLE 4-4 Edit a Site Attributes (Continued)

Attribute	URL/ Domain	Default Value	Description
Server Group - Name	URL	URL - www.sesta.com	Is either a single server or a part of a single server. The entry must include the full host name. If you specify just a host name, the site is limited to that host. If you provide directory information in addition to the host name, the site is defined as only that directory and any of its subdirectories.
Checkbox to select Server Group for deletion or verification	URL	Unselected	Unselected — Does not delete or verify the Server Group  Selected — Deletes or verifies the Server Group
Port	URL/D	80 for URL; blank for Domain	If the site you are searching uses a different port, enter it here.
Type	URL	Web Server	Web Server, File Server, FTP Server, Secure Web Server
Allowed Protocols	D	All selected	Checkboxes for http, file, ftp, https
Starting Points- Checkbox to select site for deletion	URL/D	Unselected	Unselected  Selected
Starting Points- URL	URL/D	http://URL:80	URL or domain
Starting Points - Depth	URL/D	10	1 - this URL only  2 - this URL and first links  3-10  100  unlimited
Filter Definition - Checkbox to select file type for deletion	URL/D	Unselected	Unselected  Selected

TABLE 4-4 Edit a Site Attributes (Continued)

Attribute	URL/Domain	Default Value	Description
Filter Definitions	URL/D	In this order, the defaults are Archive Files; Audio Files; Backup Files; Binary Files; CGI Files; Image Files; Java, JavaScript, Style Sheet Files; Log Files; Revision Control Files; Source Code Files; Temporary Files; Video Files.	The possible choices are Archive Files; Audio Files; Backup Files; Binary Files; CGI Files; Image Files; Java, JavaScript, Style Sheet Files; Log Files; Power Point Files; Revision Control Files; Source Code Files; Temporary Files; Video Files; Spreadsheet Files; Plug-in Files; Lotus Domino Documents; Lotus Domino OpenViews; System Directories (UNIX); System Directories (NT).
DNS Translation	URL/D	Blank	The DNS translation modifies the URL and the way it is crawled by replacing a domain name or alias with a cname. Select the available databases that you want to use from the box.
Description	URL/D	Blank	Description for the site that you had created.
Destination Database	URL/D	Use Internal Default	Select the available databases that you want to use from the list box.
Domain Group — Name	D	Domain entered. For example, *.sests.com	Name of the domain.
Checkbox to select Domain Group for deletion	D	Unselected	Unselected Selected

## Filters

Under the Filters tab, there is a Manage Filters page, which lists all the defined Filter Rules, Status of each Filter Rule, Default value for New Site, and Used in Sites. Each Filter Rule is preceded by a checkbox. To delete a Filter Rule, you need to select the corresponding checkbox and click the Delete button. To create a new filter:

1. Click the New button.  
The New Robot Filter Wizard appears. As a first step, the Specify Filter Name and Description page is displayed.
2. Enter the filter name in the Filter Name text box.
3. Enter the description for the filter in the Filter Description text box.
4. Click the Next button.

The Specify Filter Definition and Behavior page appears. This page provides the Filter Definition — Matching Rules section. The table below lists the attributes and their description provided in the Filter Definition and Behavior section.

5. Click the Finish button.

TABLE 4-5 Filter Definition and Behavior Attributes

Attribute	Default Value	Description
Filter Source	URL	Choose an option from the list box to specify the source of the filter. The available values are: URL, protocol, host, path, and MIME type.
Filter By	is	Choose an option from the list box to specify the how you want to filter the source. The available values are: is, contains, begins with, ends with, and regular expression.
Filter String	Blank	You can enter the string to define the filter.
Filter Default	Selected	Assign this filter to new sites when they are created.
Filter Behavior	Exclude documents that match this filter when Robot runs	The default option excludes documents that match this filter when robot runs. The other unselected option includes documents that match this filter when Robot runs.

Click on the Filter Rule to navigate to the Edit a Filter page. The table below lists the attributes and their description in the Edit a Filter page. The default value for these attributes are same as provided in the previous table.

TABLE 4-6 Edit a Filter Attributes

Attribute	Description
Filter Name	A descriptive name that reflects the type of file the filter applies to.
Drop down list of Filter Sources	URL, protocol, host, path, MIME type
Drop down list of positions	is, contains, begins with, ends with, regular expression

TABLE 4-6 Edit a Filter Attributes (Continued)

Attribute	Description
Text box for type (directory, protocol, file extensions) specifics	In this text box, list what you want to match. What would match in this example - <code>http://docs.sesta.com/manual.html</code> protocol is http; host contains sesta; file ends with html.
Filter Description	Describe the filter rule for yourself. The robot does not use it.
Filter Default	Use this as one of the default filters when creating new sites. If you do not check this, you can still add this filter to a new site by editing the site on the Robot, Sites page.
Filter Behavior	This attribute provides two options: Exclude documents that match this filter when Robot runs.  Include documents that match this filter when Robot runs.  By default, the first option is selected.

## Properties

Click the Robot —> Properties tab. The Manage Properties page appears. The settings on this page control the robot's operational parameters and defaults. It is divided into these sections: Crawling Speed, Completion Actions, Logfile Settings, Standard Compliance, Authentication Parameters, Proxy Settings, Link Following, Advanced Settings, and Indexing Settings.

The table below lists the attributes and their description in the Manage Properties page.

TABLE 4-7 Manage Properties Attributes

Attribute	Default Value	Description
Server Delay	No Delay	No Delay (default), 1 second, 2 seconds, 5 seconds, 10 seconds, 30 seconds, 1 minute, 5 minutes.
Maximum Connections - Max concurrent retrieval URLs	8	1, 2, 4, 8 (default), 10, 12, 16, 20.
Maximum Connections per Site	2	(no limit), 1, 2, 4, 8, 10, 12, 16, 20.
Send RDs to Indexing every	30 minutes	3 minutes, 5 minutes, 10 minutes, 15 minutes, 30 minutes (default), 1 hour, 2 hours, 4 hours, 8 hours.

TABLE 4-7 Manage Properties Attributes (Continued)

Attribute	Default Value	Description
Script to Launch	nothing	nothing (default). For sample files, see the cmdHook files in the /opt/SUNWportal/samples/robot directory (for the default installation).
After Processing all URLs	go idle	go idle (default), shut down, start over.
Contact Email	Blank	Enter your own.
Log Level	1 Generation	0 Errors only; 1 Generation (default); 2 Enumeration, Conversion; 3 Filtering; 4 Spawning; 5 Retrieval
User Agent	SunONERobot/6.2	Version of the search server.
Ignore robots.txt protocol	No	Some servers have a robot.txt file that says robots do not come here. If your search robot encounters this file on a site and this attribute is false, it does not search the site. If this attribute is true, the robot ignores the file and searches the site.
Perform Authentication?	Yes	Yes No
Robot Username	Blank	Robot uses the anonymous user name to gain access to a site.
Password	Blank	Frequently a site that allows anonymous users requires a email address as a password. This address is in plain text.
Proxy Username	Blank	Robot uses the anonymous user name to gain access to a site.
Password	Blank	Frequently a site that allows anonymous users requires an email address as a password. This address is in plain text.
Proxy Connection Type	Proxy — Manual Configuration	Direct Internet Connection, Proxy--Auto Configuration, Proxy--Manual Configuration
Auto Proxy Configuration Type	Local Proxy File	Local Proxy File, Remote Proxy File

TABLE 4-7 Manage Properties Attributes (Continued)

Attribute	Default Value	Description
Auto Proxy Configuration Location	Blank	The auto proxy has a file that lists all the proxy information needed.  An example of a local proxy file is <code>robot.pac</code> . An example of a remote proxy file is <code>http://proxy.sesta.com:8080/proxy.pac</code>
Manual Proxy Configuration HTTP Proxy	Host Name:Port	Format: <code>server1.sesta.com:8080</code> . These three manual configuration values are put in the <code>robot.pac</code> file in the  <code>/var/opt/SUNWportal/searchservers/search1/config</code> directory.
Manual Proxy Configuration HTTPS Proxy	Host Name:Port	This manual configuration value is put in the <code>robot.pac</code> file.  Format: <code>server1.sesta.com:8080</code>
Manual Proxy Configuration FTP Proxy	Host Name:Port	This manual configuration value is put in the <code>robot.pac</code> file.  Format: <code>server1.sesta.com:8080</code>
Follow Links in HTML	Yes	Extract hyperlinks from HTML
maximum links	1024	Limits the number of links the robot can extract from any one HTML resource. As the robot searches sites and discovers links to other resources, it could conceivably end up following huge numbers of links a great distance from its original starting point.
Follow Links in Plain Text	No	Extract hyperlinks from plain text.
maximum links	1024	Limits the number of links the robot can extract from any one text resource.

TABLE 4-7 Manage Properties Attributes (Continued)

Attribute	Default Value	Description
Use Cookies	No	If checked, the robot uses cookies when it crawls. Some sites require the use of cookies in order for them to be navigated correctly. The robot keeps its cookies in a file called <code>cookies.txt</code> in the robot state directory. The format of <code>cookies.txt</code> is the same format as used by the Netscape™ Communicator browser.
Use IP as Source	Yes	In most cases, the robot operates only on the domain name of a resource. In some cases, you might want to be able to filter or classify resources based on subnets by Internet Protocol (IP) address. In that case, you must explicitly allow the robot to retrieve the IP address in addition to the domain name. Retrieving IP addresses requires an extra DNS lookup, which can slow the operation of the robot. If you do not need this option, you can turn it off to improve performance.
Enable Smart Host Heuristics	No	<p>If checked, the robot converts common alternate host names used by a server to a single name. This is most useful in cases where a site has a number of servers all aliased to the same address, such as <code>www.sesta.com</code>, which often have names such as <code>www1.sesta.com</code>, <code>www2.sesta.com</code>, and so on.</p> <p>When you select this option, the robot will internally translate all host names starting with <code>wwwn</code> to <code>www</code>, where <code>n</code> is any integer. This attribute only operates on host names starting with <code>wwwn</code>.</p> <p>This attribute cannot be used when CNAME resolution is OFF (No).</p>

TABLE 4-7 Manage Properties Attributes (Continued)

Attribute	Default Value	Description
Resolve Host Names to CNAMEs	No	<p>If checked, the robot validates and resolves any host name it encounters into a canonical host name. This allows the robot to accurately track unique RDs. If unchecked, the robot validates host names without converting them to the canonical form. So you may get duplicate RDs listed with the different host names found by the robot.</p> <p>For example, <code>devedge.sesta.com</code> is an alias for <code>developer.sesta.com</code>. With CNAME resolution on, a URL referenced as <code>devedge.sesta.com</code> is listed as being found on <code>developer.sesta.com</code>. With CNAME resolution off, the RD retains the original reference to <code>devedge.sesta.com</code>.</p> <p>Smart Host Heuristics cannot be enabled when CNAME resolution is OFF (No).</p>
Accepts Commands from any Host	No	<p>Most robot control functions operate through a TCP/IP port. This attribute controls whether commands to the robot must come from the local host system (No), or whether they can come from anywhere on the network (Yes).</p> <p>It is recommended that you restrict direct robot control to the local host (No). You can still administer the robot remotely through the Administration Console.</p>
Default Starting Point Depth	10	<p>1- starting points only, 2- bookmark style, 3-10, unlimited.</p> <p>Default value for the levels of hyperlinks the robot traverses from any starting point. You can set the depth for any given starting point by editing the site on the Robot, Sites page.</p>



TABLE 4-7 Manage Properties Attributes (Continued)

Attribute	Default Value	Description
Work Directory	<code>/var/opt/SUNWportal/ searchservers/search1/tmp</code>	Full pathname of a temporary working directory the robot can use to store data. The robot retrieves the entire contents of documents into this directory, often many at a time, so this space should be large enough to handle all of those at once.
State Directory	<code>/var/opt/SUNWportal/ searchservers/search1/robot</code>	Full pathname of a temporary directory the robot uses to store its state information, including the list of URLs it has visited, the URL pool, and so on. This database can be quite large, so you might want to locate it in a separate partition from the Work Directory.
Page Extraction Index	Partial Text	Full Text uses the complete document in the resource description. Partial text only uses the specified number of bytes in the resource description.
extract first # bytes	4096	Enter the number of bytes.
Extract Table Of Contents	Yes	Yes includes the Table of Contents in the resource description.
Extract data in META tags	Yes	Yes includes the META tags in the resource description.
Allow No Existing Classifications	Yes	Yes allows none of the existing classifications in the resource description.

TABLE 4-7 Manage Properties Attributes (Continued)

Attribute	Default Value	Description
Document Converters	All selected; if unselected, that type of document cannot be indexed.	Adobe PDF Corel Presentations Corel Quattro Pro FrameMaker Lotus Ami Pro Lotus Freelance Lotus Word Pro Lotus 1-2-3 Microsoft Excel Microsoft Powerpoint Microsoft RTF Microsoft Word Microsoft Works Microsoft Write WordPerfect StarOffice™ Calc StarOffice Impress StarOffice Writer XyWrite
Converter Timeout	600	Time in seconds allowed for one document to be converted to HTML. If this time is exceeded, the URL is excluded.

## Indexing

The robot searches sites and collects documents based on the filters you have selected. The documents collected are in many different formats. To make them uniform and easily readable they need to be in one format, which is HTML. This page controls some of the parts that go into each resource description.

## Simulator

You can find the simulator attributes in the Robot Utilities page under the Utilities tab. The Robot Utilities page is a debugging tool that performs a partial simulation of robot filtering on a URL. You can type in a new URL to check. It checks the URL, DNS translations (including Smart Host Heuristics), and site redirections. It does not check the contents of the document specified by the URL, so it does not detect duplications, MIME types, network errors, permissions, and the like. The simulator indicates whether the listed sites would be accepted by the robot (ACCEPTED) or not (WARNING).

The table below provides the attributes and their description in the Simulator section in the Robot Utilities page.

TABLE 4-8 Robot Simulator Attributes

Attribute	Default Value	Description
Run Simulator on	URLs you have already defined and one blank text box.	You can check access to a new site by typing its URL in the blank text box. This checks to see if the new site accepts crawling.  Format <code>http://www.sesta.com:80/</code>
Show advanced DNS information	Unselected	Selected displays more information about the site.
Check for server redirects	Selected	Selected checks for any server redirects.

## Site Probe

The site probe attributes are also available in the Robot Utilities page. This page is a debugging tool that checks for DNS aliases, server redirects, and virtual servers. This tool returns information about site but does not test its acceptance of crawling.

The table below provides the site Probe attributes and their description.

TABLE 4-9 Robot Site Probe Attributes

Attribute	Default Value	Description
Run Site Probe on	Blank	Type in URL in format <code>http://www.sesta.com:80</code>
Show advanced DNS information	Unselected	Selected displays more information about the site including IP addresses.



## Search Attributes: Database

---

This chapter explains the attributes provided for the search database. The Database attributes are divided as follows:

- “Management” on page 69
- “Import Agents” on page 70
- “Resource Descriptions” on page 72
- “Schema” on page 74
- “Analysis” on page 76

### Management

Before knowing about the Search database, you need to know how to partition the database. To partition the database, use the `run-cs-cli rdmgr -G` command, because stopping the search server is required.

The initial Manage Databases page lists the available databases. You can select a database by selecting the checkbox preceding to it. Click the New, Reindex, Purge, Analyze, Manage, or Expire resource descriptions button to perform the necessary action on the selected database.

You should reindex the database if you have edited the schema to add or remove an indexed field (as author), or if a disk error has corrupted the index. You need to restart the server after you change the schema.

Because the time required to reindex the database is proportional to the number of RDs in the database, a large database should be reindexed when the server is not in high demand.

When you purge the contents of the database, disk space used for indexes will be recovered, but disk space used by the main database will not be recovered; instead, it is reused as new data is added to the database.

Expiring a database deletes all RDs that are deemed out-of-date. It does not decrease the size of the database. By default, an RD is scheduled to expire in 90 days from the time of creation.

The table below lists the Database Management attributes and their description.

TABLE 5-1 Database Management Attributes

Attribute	Default Value	Description
Name	True or False	Name for the database used by Search.
Federated	True or False	For a Federated database, this value is True. Otherwise, the value is False.

## Import Agents

Import agents are the processes that bring resource descriptions from other servers or databases and merge them into your search database.

The initial Manage Import Agents page lists the available import agents. You can select an import agent by selecting the checkbox preceding to it. Click the New, Enable, Disable, Delete, or Run All Enabled Import Agents to perform the necessary action on the selected import agent. To schedule the import agents, select Scheduling on the lower menu bar.

If you choose to create a new import agent or edit or modify an existing import agent, the following database import agent attributes are displayed.

The table below lists the Database Import Agent attributes and their description.

TABLE 5-2 Database Import Agent Attributes

Attribute	Default Value	Description
Import agent source	Local File	Select either Local File or Remote Server (if one is enabled).
Local File Path	Blank for new	Gives the full path name of local file that contains valid resource descriptions in search result (Summary Object Interchange Format). This can be a file on another server, as long as the path is addressable as if it were locally mounted.
Destination Database	Blank	Name of the destination database.
Remote Server Host	Blank for new	Gives the URL of the search server to retrieve resource descriptions from; the format is: <code>www.sesta.com</code>

TABLE 5-2 Database Import Agent Attributes (Continued)

Attribute	Default Value	Description
Remote Server Port	Blank for new	Specify the port number for the given remote server host. For example, <b>8080</b>
Search URI	Blank for new	Enter full path and file names. Use <code>/search1/search</code> .
Enable SSL	False	If this is a server-to-server transaction, select if the servers should use the SSL (Secure Sockets Layer) protocol.
User	Blank for new or none	If you selected Use User/Password, enter a user.
Password	Blank for new or none	If you selected Use User/Password, enter a password (shown as *).
Content Transfer	All	<p>By default, an import agent asks for all resource descriptions added or changed since its last import from the same source.</p> <p>The search query specifies that the import agent should request only certain resource descriptions from the source. This is much the same way that users request listings of resources from the search database.</p> <p>Use Scope, View-Attributes and View-Hits fields to specify the query.</p>
Scope	Blank for new	The text of the query. The query syntax is identical to that used for end-user queries from the server.
View-Attributes	Blank for new	Lists which fields (not case sensitive) you want to import in each resource description. For example, title and author. The default is all.
View-Hits	Blank for new	The maximum number of matching resource descriptions to import. If no hits are specified, it defaults to 20.
Network Timeout in seconds	Blank for new	Specifies the number of seconds the import agent will allow before timing out the connection over the network. You can adjust this to allow for varying network traffic and quality.

TABLE 5-2 Database Import Agent Attributes (Continued)

Attribute	Default Value	Description
Title	Blank for new	Title of the import agent.
Remote Database	Blank	Name of the database on the remote server.

## Resource Descriptions

The initial Resource Descriptions page allows you to search the Resource Descriptions in the database. For example, you can correct a typographical error in an RD or manually assign RDs discovered by the robot to categories.

The table below lists the Resource Descriptions attributes and their description.

TABLE 5-3 Resource Descriptions Attributes

Attribute	Default Value	Description
New		Opens up the New Resource Description page where you can enter the URL to create a new search RD.
Edit		Opens up the Edit URL page where you can modify only the attributes of a search RD, which can be edited.
Edit All		Opens up the Edit Resource Descriptions page where you can modify a group of search RD.
Delete		Deletes the selected search RD.
Filter	All	The options available are Categorized (to list Categorized RDs), Uncategorized (to list Uncategorized RDs), and Custom Filter.



TABLE 5-3 Resource Descriptions Attributes (Continued)

Attribute	Default Value	Description
Custom Filter		<p>This attribute provides the following options:</p> <p>Query (Selected by default)</p> <p>URL</p> <p>Category</p> <p>Text box — To enter the search string.</p> <p>On selecting the Category option, the Choose button appears. Click the Choose button to go to the Select a Category page where you can select the category.</p>

A successful search displays the Number of RDs found and a list box with the RDs found. If you navigate to the Edit page for a resource description, you can modify only the attributes of a resource description, which can be edited. By default, you cannot edit some of the RD attributes listed in the table below. To edit all these attributes except the Classification attribute, change the settings in the Database/Schema/Edit schema attribute page.

The table below lists the Database RD Editable attributes and their description. The default value for these attributes depends on the selected RD.

TABLE 5-4 Database RD Editable Attributes

Attribute	Description
Author	Author(s) of the document.
Author e-mail	Email address to contact the Author(s) of the document.
Classification	Category name if classified; No Classification if not classified.
ReadACL	Related to document level security.
Content-Charset	Content-Charset information from HTTP Server.
Content-Encoding	Content-Encoding information from HTTP Server.
Content-Language	Content-Language information from HTTP Server.
Content-Length	Content-Length information from HTTP Server.
Content-Type	Content-Type information from HTTP Server.
Description	Description from RD.

TABLE 5-4 Database RD Editable Attributes (Continued)

Attribute	Description
Expires	Date on which resource description is no longer valid.
Full-Text	Entire contents of the document.
Keywords	Keywords taken from meta- tags.
Last-Modified	Date when the document was last modified.
Partial-text	Partial selection of text from the document
Phone	Phone number for Author contact
Title	Title of RD
URL	Uniform Resource Locator for the document
virtual-db	Used to implement virtual database.

## Schema

When you click the Schema tab under Databases, you will get the Manage Search Schema page. This page lists the available Search Schema attributes. The schema determines what information is in a resource description and what form that information is in. You can add new attributes or fields to an RD and set which ones can be edited and which ones can be indexed. When importing new RDs, you can convert schemas embedded in new RDs into your own schema.

The table below lists the Search Schema attributes and their description.

TABLE 5-5 Search Schema Attributes

Attribute	Description
Author	Author(s) of the document.
Author-EMail	Email address to contact the Author(s) of the document.
Content-Charset	Content-Charset information from HTTP Server.
Content-Encoding	Content-Encoding information from HTTP Server.
Content-Language	Content-Language information from HTTP Server.
Content-Length	Content-Length information from HTTP Server.
Content-Type	Content-Type information from HTTP Server.
Description	Brief one-line description for document.

TABLE 5-5 Search Schema Attributes (Continued)

Attribute	Description
Expires	Date on which resource description is no longer valid.
Full-Text	Entire contents of the document.
Keywords	Keywords that best describe the document.
Last-modified	Date when the document was last modified.
Partial-Text	Partial selection of text from the document.
Phone	Phone number for Author contact.
ReadACL	Used by Search servers to enforce security.
Title	Title of the document.
URL	Uniform Resource Locator for the document
virtual-db	Used to implement virtual database.

When you select the checkbox preceding to a search schema attribute and click on it, the Edit *search schema name* page appears. This page displays all the attributes to edit a search schema attribute. The table below lists the attributes and their description to edit a search schema attribute.

TABLE 5-6 Edit Search Schema Attribute Attributes

Attribute	Default Value	Description
Name	Author	When you import new RDs, you can convert schemas embedded in new RDs into your own schema. You would use this conversion when there are discrepancies between the names used for fields in the import database schema and the schema used for RDs in your database.  An example would be if you imported RDs that used Writer as a field for the author and you used Author in your RDs as the field for the author. The conversion would be Writer to Author, so you would enter Writer in this text box.
Description	Author(s) of the document	
Aliases	Blank	

TABLE 5-6 Edit Search Schema Attribute Attributes (Continued)

Attribute	Default Value	Description
Editable	false	If true (checked), the selected attribute (field) appears as Editable attribute in the Edit page for a resource description.  Description, Keywords, Title and ReadACL are editable.
Indexable	true	If true (checked), the selected attribute (field) can be used as a basis for indexing.  Author, Title and URL appear in the menu in the Advanced Search screen  for the end user. This allows end users to search for values in those particular fields.  Author, Expires, Keywords, Last Modified, Title, URL and ReadACL can be used as the basis for indexing.
Score Multiplier	Blank	A weighting field for scoring a particular element. Any positive value is valid.
Data Type	String	Defines the data type. You need to choose the data type from the list box.

## Analysis

The Analysis page shows a sorted list of all sites and the number of resources from that site currently in the search database. Select Update Analysis to update the analysis on file.

The table below lists the Database Analysis attributes and their description.

TABLE 5-7 Database Analysis Attributes

Attribute	Default Value	Description
Number of RDs	Current number of RDs retrieved from the URL.	Lists current number of RDs from that URL.

TABLE 5-7 Database Analysis Attributes (Continued)

Attribute	Default Value	Description
URL	URL that the robot has successfully searched.	A URL that has added.
Protocol	Protocol it uses to retrieve the RDs from that URL.	Lists the protocol used while collecting the RDs from a web site.



# Search Attributes: Categories

---

This chapter explains the attributes available for Search Categories. This chapter contains the following sections:

- [“Overview” on page 79](#)
- [“Manage Categories” on page 79](#)
- [“Manage Classification Rules” on page 80](#)

## Overview

End users interact with the search database in two distinct ways: They can type direct queries to search the database, or they can browse through the database contents using a set of categories you design. You assign resources in a search database to categories to clarify complexity. If a large number of items are in the database, it is helpful to group related items together. Your primary concern in setting up your categories should be usability so that end users can more quickly locate specific kinds of items.

The search server uses a hierarchy of categories called a taxonomy. The term taxonomy in general describes any system of categories. In the context of a networked resource database such as the search server database, it describes any method you choose of categorizing network resources to facilitate retrieval.

## Manage Categories

When you click the Browse/Search tab under Categories, you will get the Manage Categories page. This page displays the categories in the taxonomy allowing you to browse the categories. You can also use this page to search for a category.

The attributes present in the Manage Categories page are shown in two tables. The first table lists the Manage Categories Attributes and the second table lists the Edit Category Attributes.

TABLE 6-1 Manage Categories Attributes

Attribute	Default Value	Description
New		Opens up the New Search Category page which you can use to create a new category.
Edit		Opens up the Edit Category page for that category where you can modify attributes of a Category.
Delete		Deletes the selected Category.
Reindex		Reindexes the database. If you have just created your taxonomy, you need to index the database to make category search available to your end users. If you have changed your categories, you need to reindex the database to make it up-to-date. Save the categories tree before you reindex the database. Load the new taxonomy.
Search	Blank	Used to search for a Category.

When you select the checkbox preceding to a Category and click the Edit button, the Edit *selected Category name* page appears. You can use the attributes in this page to edit a Category. The following table lists the attributes available in the Edit page and their description.

TABLE 6-2 Edit Category Attributes

Attribute	Description
Category	Displays the Category name.
Description	Displays the description of the selected category.
Matching Rule	Displays the matching rule to use with the selected category.

## Manage Classification Rules

After you set up the categories for your database, Click New to set or change the rules the robot for selected categories to assign resources to categories. The table below lists the Classification Rules attributes and their description.



TABLE 6-3 Classification Rules Attributes

Attribute	Default Value	Description
Source	Author	The valid attributes include: <ul style="list-style-type: none"> <li>■ Author</li> <li>■ Author-EMail</li> <li>■ Content-Charset</li> <li>■ Content-Encoding</li> <li>■ Content-Language</li> <li>■ Content-Length</li> <li>■ Content-Type</li> <li>■ Description</li> <li>■ Expires</li> <li>■ Full-Text</li> <li>■ Keywords</li> <li>■ Last-modified</li> <li>■ Partial-Text</li> <li>■ Phone</li> <li>■ ReadACL</li> <li>■ Title</li> <li>■ URL</li> <li>■ Host</li> <li>■ Protocol</li> <li>■ IP</li> <li>■ Path</li> <li>■ Type</li> </ul>
Method	is	is, contains, begins with, ends with, regular expression
Criteria	Blank	Specifies the criteria for the rule.
Classification	Blank	Category to in which to classify the RD if the rule conditions are met. Type the category or use the Select a Category page to browse to it.



# Search Attributes: Scheduling

---

This chapter describes how you can schedule the Robot and Import Agents in Portal server 7.1 console application. This chapter contains the following sections:

- “Scheduling Robot” on page 83
- “Scheduling Import Agents” on page 84
- “Scheduling Autoclassify” on page 84

## Scheduling Robot

In the Sun Java System Portal Server 7.1 Administration console, click the Scheduling tab. By default, the Robot tab is enabled and shows the Scheduling Robot page. This page is where you set up the automatic search schedule for the robot. Use the attributes in the page to start and stop the robot at the times you specify. The table below lists the robot schedule attributes and their description.

TABLE 7-1 Robot Schedule Attributes

Attribute	Default Value	Description
Start Robot Time in hours and minutes	none selected	This is the time that the robot starts to search. You need to specify the robot start time to schedule. The format is hh : mm : am or pm. For example, 12 : 00 : am.
Days	none selected	Sun, Mon, Tue, Wed, Thu, Fri, or Sat Select at least one day.

TABLE 7-1 Robot Schedule Attributes (Continued)

Attribute	Default Value	Description
Stop Robot Time in hours and minutes	none selected	If you plan to run the robot continuously, it is recommended that you stop and restart it at least once per day. This gives the robot a chance to release resources and re-initialize itself.  You need to specify the time in the format: hh : mm : am or pm. For example, 12 : 00 : am.
Days	none selected	Sun, Mon, Tue, Wed, Thu, Fri, or Sat

## Scheduling Import Agents

The Import Agents tab is located under the Scheduling tab. When you click the Import Agents tab, the Scheduling Import Agents page appears. This page is where you set up the schedule for running the import agents. The table below lists the database import schedule attributes and their description.

TABLE 7-2 Database Import Schedule Attributes

Attribute	Default Value	Description
Start Import Time in hours and minutes	none selected	Time that the import agent starts to import. You need to specify the start import time to schedule. The format is hh : mm : am or pm. For example, 12 : 00 : am.
Days	none selected	Sun -Sat  Select at least one day.

## Scheduling Autoclassify

When you click the Autoclassify tab under Scheduling, the Scheduling Autoclassify page appears. This page lists the available Start Autoclassify attributes, which you can use to schedule autoclassify to start at the times you specify. The table below lists the attributes Start Autoclassify attributes and their description.

TABLE 7-3 Scheduling Autoclassify Attributes

Attributes	Default Value	Description
Day of the Week	None selected	You need to choose the particular day in a week to schedule autoclassify. The listed days are Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday.
Time	None selected	You need to specify the time to schedule autoclassify. The format is hh : mm : am or pm. For example, 12 : 00 : am.



# Search Attributes: Reports

---

This chapter explains the attributes available for Search Reports. This chapter contains the following sections:

- “Introduction” on page 87
- “Excluded URLs” on page 87
- “Advanced Robot Reports” on page 88
- “Log Files” on page 88
- “Popular Searches” on page 89
- “Search Server Log Settings” on page 89

## Introduction

The Reports section allows you to monitor your search server. You can see a summary of its activity: what sites were searched, what URLs were excluded and why, detailed information about URLs visited by the robot, and what your end users are interested in.

## Excluded URLs

This page shows a list of robot runs. To display a list of reasons URLs were excluded, set a date, select Show, then select one of the Reasons for Exclusion. Displayed is a list of the excluded URLs for that reason. Duplicate and warning exclusions have been removed.

The table below lists Reports Excluded URLs attributes and their description.

TABLE 8-1 Reports Excluded URLs Attributes

Attribute	Default Value	Description
Count	Numbers	List of numbers with reasons for exclusion.
Reason for Exclusion	List of reasons sites have not been allowed. Each reason is linked to a list of all the URLs that were excluded for that reason.	Filter rules, file not found, site not allowed, protocol not allowed, errors, duplication are some of the reasons URLs were excluded.

## Advanced Robot Reports

When you click the Advanced Robot Reports tab under Reports, you will get the Robot Report Files page. This page gives you access to a number of different reports from the robot. Select from a drop down list to get information for chosen report to show up. The Refresh button gets the current information.

The table below lists the Reports Robot Report Files attributes and their description.

TABLE 8-2 Robot Report Files Attributes

Attribute	Default Value	Description
Reports	Blank	Overview, Version, Configuration, Performance, Database, Libnet, Modules, DNS Cache Dump, All Servers Found, URLs-Ready For Extraction, URLs-Ready For Indexing, URLs- Waiting For Filtering (URL pool), URLs- Waiting for Indexing.

## Log Files

This page allows you to view the entries or specific lines from a log file. Drop down list of log files. Enter the number of lines you want to be displayed when you select Show button.

The table below lists the Reports View Log Files attributes and their description.



TABLE 8-3 Reports —&gt; Log Files Attributes

Attribute	Default Value	Description
Log File	Blank	Excluded URLs (filter), RD Manager (rdmgr), RDM Server (rdmsvr), Robot Activities (robot), Search Engine (searchengine), User Queries (rdm).
Lines to show	25	A number you can enter to display the most current entries in the log file.

## Popular Searches

When you click the Popular Searches tab under Reports, you will get the Manage Popular Searches page. This page lists the information on your popular searches. This page allows you to see what users are searching for. The most frequent searches appear first in the report. The table lists the Popular Searches attributes and their description.

TABLE 8-4 Popular Searches Attributes

Attribute	Default Value	Description
Filter	All Items	All Items includes what categories users browse in. Exclude Browse excludes browsing statistics.

## Search Server Log Settings

You can set the log level for all the server log settings attributes listed in the table below in the `/var/opt/SUNWportal/searchservers/search1/config/SearchLogConfig.properties` file. The table lists the Search Server Log Settings attributes and their description.

TABLE 8-5 Search Server Log Settings Attributes

Attribute	Default Value	Description
Search (rdm)	<code>/var/opt/SUNWportal/searchservers/search1/logs/rdm.log</code>	Logs the queries end users make of the database.
Index Maintenance	<code>/var/opt/SUNWportal/searchservers/search1/logs/searchengine.log</code>	Logs the transactions involving the search engine, except for not registration of resource descriptions.

TABLE 8-5 Search Server Log Settings Attributes (Continued)

Attribute	Default Value	Description
RD Manager	/var/opt/SUNWportal/searchservers/search1/ logs/rdmgr.log	Logs the registration of resource descriptions from the robot or import agents into the database. You can view this log as a RD Manager (rdmgr) report.
RDM Server	/var/opt/SUNWportal/searchservers/search1/ logs/rdmsvr.log	Logs debugging information on RDM transactions. The level of detail is controlled by the Log Level. You can view this log as a RDM Server (rdmsvr) report.

The possible levels of all the above logs are SEVERE, WARNING, INFO, FINE, FINER, and FINEST.

# Subscriptions Attributes

---

A Subscription is a profile of interest covering many sources of information such as categories, discussions, and searchable documents. The Subscriptions Service consists of root level, organization and user attributes. These attributes are available in the Sun Java System Portal Server Console. This chapter explains the following Subscriptions Service attributes:

- “Root Level Attributes” on page 91
- “Organization Attributes” on page 92
- “User Attributes” on page 94
- “Scheduling Profiler Attributes” on page 98

## Root Level Attributes

This section describes the root level attributes for the Subscriptions Service with which you can set the system-wide default maximum number of subscriptions for each type, categories, discussion, and saved searches. You can find these root level attributes in the Subscriptions page, which is in the Subscriptions Settings tab, when you select the TopLevel (Global) DN from the Select DNs drop down list. You can use this Subscriptions page to edit the subscription service attributes at root level.

The table below lists the subscription service attributes at root level and their description.

TABLE 9-1 Subscriptions Service - Root Level Attributes

Attribute	Default Value	Description
Maximum Category Subscriptions	5	Specifies the maximum number of subscriptions on categories that can be defined and stored in the Sun Java System Access Manager.

TABLE 9-1 Subscriptions Service - Root Level Attributes (Continued)

Attribute	Default Value	Description
Maximum Discussion Subscriptions	5	Specifies the maximum number of subscriptions on discussions that can be defined and stored in the Sun Java System Access Manager.
Maximum Saved Search	5	Specifies the maximum number of saved search subscriptions that can be defined and stored in the Sun Java System Access Manager.

## Organization Attributes

This section describes the organization attributes for the Subscriptions Service.

You can find these organization attributes in the Subscriptions page, which is in the Subscriptions Settings tab, when you select the Organization name [Org] DN from the Select DNs drop down list. You can use this Subscriptions page to edit the organization attributes. The table below lists the subscription service organization attributes and their description.

TABLE 9-2 Subscriptions Service — Organization Attributes

Attribute	Default Value	Description
Profiler SMTP	None	The host system that servers as the SMTP server to route Email notifications to the end user.
Profiler Email	None	This is the Email address, which is in the From header in the Email notification sent to the end user. It should be in the format: <code>id@domain</code> .

TABLE 9-2 Subscriptions Service — Organization Attributes *(Continued)*

Attribute	Default Value	Description
Profiler Provider	<p>The syntax for Profiler Provider should be:</p> <pre>http://server.domain:PORT/? provider=Profiler&amp;desktop.suid= uid=devauthlessanonymous, ou=People,o=DeveloperSample, dc=india,dc=sun,dc=com</pre> <p>In the URL, the values of the following attributes for authlessanonymous user defined are:</p> <ul style="list-style-type: none"> <li>■ uid — devauthlessanonymous</li> <li>■ ou — people</li> <li>■ o — DeveloperSample</li> <li>■ dc — india</li> <li>■ dc — sun</li> <li>■ dc — com</li> </ul>	This is the URL of the Profiler Channel present in the global DP, which is accessible by a valid authlessanonymous user defined for a given organization.
Profiler Default Search	<p>This value should be in the format:</p> <pre>http://HOST:PORT/search1/search</pre>	This is the URL for the default search server.
Profiler Maximum Hits	5	This is the maximum number of result hits that any given end user Subscriptions in the organization would see in the Email notification sent to a user
Maximum Category Subscriptions	5	Specifies the maximum number of subscriptions on categories that can be defined and stored in the Sun Java System Access Manager.
Maximum Discussion Subscriptions	5	Specifies the maximum number of subscriptions on discussions that can be defined and stored in the Sun Java System Access Manager.
Maximum Saved Search	5	Specifies the maximum number of saved search subscriptions that can be defined and stored in the Sun Java System Access Manager.

## User Attributes

This section describes the user attributes for Subscriptions Service.

You can find these User attributes in the Subscriptions page, which appears under the Settings tab, when you select the Username [User] DN from the Select DN's drop down list. You can use this Subscriptions page to edit the subscription service user attributes. The table below lists the subscription service user attributes and their description.

TABLE 9-3 Subscriptions Service - User Attributes

Attribute	Default Value	Description
Profiler Enabled	Disabled	Enabled — Evaluates user subscriptions and result in an Email notification.  Disabled — The user subscriptions are skipped during the subscriptions profiling run.

TABLE 9-3 Subscriptions Service - User Attributes (Continued)

Attribute	Default Value	Description
Category Subscriptions <ul style="list-style-type: none"> <li>■ New value:</li> <li>■ Current values:</li> </ul>	Blank Blank	<p>This field defines the subscriptions details. The format is:</p> <p>label   target category   scope   lapsed time   rating   search server   database   status</p> <p>where:</p> <ul style="list-style-type: none"> <li>■ label — Refers to a logical reference given to the edited subscriptions and it must be a string. This is a required field.</li> <li>■ target category — must be the string representation of a target category. For example : ABC:DEF:GHI</li> <li>■ scope — Refers to a search query and it must be of a string format that is valid search string, including search operators.</li> <li>■ lapsed time — Must be one of the following numbers:               <ul style="list-style-type: none"> <li>■ 0 = forever</li> <li>■ 1 = since yesterday</li> <li>■ 7 = since last week</li> <li>■ 30 = since last year</li> <li>■ 180 = since last six months</li> <li>■ 365 = since last year</li> </ul> </li> <li>■ rating — This is the minimum rating that a matching document should be to be selected as a match for the subscription. The values are:               <ul style="list-style-type: none"> <li>■ -1 = irrelevant</li> <li>■ 0 = routine</li> <li>■ 1 = interesting</li> <li>■ 2 = important</li> <li>■ 3 = must read</li> </ul> </li> <li>■ search server — This is the URL of the target search server that will be queried by this subscription to find content matching the criteria of the subscription.</li> <li>■ database — This is the target search database that the subscription is searching potential</li> </ul>

TABLE 9-3 Subscriptions Service - User Attributes (Continued)

Attribute	Default Value	Description
Discussion Subscriptions <ul style="list-style-type: none"> <li>■ New value:</li> <li>■ Current values:</li> </ul>	Blank Blank	<p>This field defines the subscriptions details. The format is:</p> <p>label   target discussion   scope   lapsed time   rating   search server   database   status</p> <p>where:</p> <ul style="list-style-type: none"> <li>■ label — Refers to a logical reference given to the edited suscriptions and it must be a string. This is a required field.</li> <li>■ target discussion — This is the space-separated string representation of the discussion ID.</li> <li>■ scope — Refers to a search query and it must be of a string fomat that is valid search string, including search operators.</li> <li>■ lapsed time — Must be one of the following numbers:               <ul style="list-style-type: none"> <li>■ 0 = forever</li> <li>■ 1 = since yesterday</li> <li>■ 7 = since last week</li> <li>■ 30 = since last year</li> <li>■ 180 = since last six months</li> <li>■ 365 = since last year</li> </ul> </li> <li>■ rating — This is the minimum rating that a matching document should be to be selected as a match for the subscription. The numbers are:               <ul style="list-style-type: none"> <li>■ -1 = irrelevant</li> <li>■ 0 = routine</li> <li>■ 1 = interesting</li> <li>■ 2 = important</li> <li>■ 3 = must read</li> </ul> </li> <li>■ search server — This is the URL of the target search server that will be queried by this subscription to find content matching the criteria of the subscription.</li> <li>■ database — This is the target search database that the subscription is searching potential</li> </ul>



TABLE 9-3 Subscriptions Service - User Attributes (Continued)

Attribute	Default Value	Description
Saved Search <ul style="list-style-type: none"> <li>■ New value:</li> <li>■ Current values:</li> </ul>	Blank  Blank	<p>This field defines the subscriptions details. The format is:</p> <p>label   scope   lapsed time   rating   search server   database   status</p> <p>where:</p> <ul style="list-style-type: none"> <li>■ label — Refers to a logical reference given to the edited subscriptions and it must be a string. This is a required field.</li> <li>■ scope — Refers to a search query and it must be of a string format that is valid search string, including search operators.</li> <li>■ lapsed time — Must be one of the following numbers:               <ul style="list-style-type: none"> <li>■ 0 = forever</li> <li>■ 1 = since yesterday</li> <li>■ 7 = since last week</li> <li>■ 30 = since last year</li> <li>■ 180 = since last six months</li> <li>■ 365 = since last year</li> </ul> </li> <li>■ rating — This is the minimum rating that a matching document should be to be selected as a match for the subscription. The numbers are:               <ul style="list-style-type: none"> <li>■ -1 = irrelevant</li> <li>■ 0 = routine</li> <li>■ 1 = interesting</li> <li>■ 2 = important</li> <li>■ 3 = must read</li> </ul> </li> <li>■ search server — This is the URL of the target search server that will be queried by this subscription to find content matching the criteria of the subscription.</li> <li>■ database — This is the target search database that the subscription is searching potential matches in. This is a single value database.</li> <li>■ status — This is boolean value that marks whether the subscriptions is active or inactive. Active implies</li> </ul>

## Scheduling Profiler Attributes

This section describes the scheduling profiler attributes for Subscriptions Service.

You can find the scheduling profiler attributes in the Subscriptions page, which is in the Subscriptions Scheduling Profiler tab. You can use this Subscriptions page to start and stop times for the profiler. The following tables provide attributes for profiler run.

Click the New tab to enable the Start Profiler attributes. You can use these Start Profiler attributes shown in the table to setup the automatic start time for the profiler.

**TABLE 9-4** Attributes for Start Profiler

Attribute	Default Value	Description
Day of the Week	None Selected	Sun, Mon, Tue, Wed, Thu, Fri, or Sat
Time	None Selected	This is the time when the profiler starts to search. You need to select the hours : minutes : am or pm. For example, 12 : 00 : pm

Click the New tab to enable the Stop Profiler attributes. You can use these Stop Profiler attributes shown in the table to setup the automatic stop time for the profiler.

**TABLE 9-5** Attributes for Stop Profiler

Attribute	Default Value	Description
Day of the Week	None Selected	Sun, Mon, Tue, Wed, Thu, Fri, or Sat
Time	None Selected	This is the time when the profiler stops. You need to select the hours : minutes : am or pm. For example, 12 : 00 : pm

PART II

Configuration Files

- Chapter 10
- Chapter 11
- Chapter 12



# Desktop Configuration Properties File

---

This chapter contains the following sections:

- “Overview” on page 101
- “Parameters” on page 101

## Overview

The `desktopconfig.properties` file defines server-specific parameters that the Desktop reads during initialization. Any changes to this file require a server restart in order to go into effect. By default, this file is in the `/var/opt/SUNWportal/portals/<portal_id>/config` directory.

## Parameters

You cannot customize the parameters, which are marked as `Internal`. So, you can only configure the debug level and the base directory for additional classes. The parameters available in the `desktopconfig.properties` file and their description are:

```
defaultDesktopType=default  
[Internal]
```

Default desktop type used by the `ErrorProvider` when `DesktopAppContext` is available but `DesktopContext` is not available.

```
getterPoolMinSize=0  
[Internal]
```

```
getterPoolMaxSize=0  
[Internal]
```

```
getterPoolPartitionSize=0  
[Internal]
```

callerPoolMinSize=0  
[Internal]

callerPoolMaxSize=0  
[Internal]

callerPoolPartitionSize=0  
[Internal]

cookiePrefix=desktop  
[Internal]

Prefix used for all desktop cookies.

lb.cookie.name=

Value of this cookie is <portal\_id>.<instance\_id>. If Portal Server and Access Manager are running on separate nodes and both of them are available at load-balancer URLs, such cookie will not set by the Portal Server. Although Portal Server works with some performance hit in terms of resource consumption and response times. The recommended value, if used, is <cookiePrefixPropertyValue>.lb.cookie.

templateScanInterval=30

Defines number of seconds between scans (checking for changes) of the template files in the /var/opt/SUNWportal/portals/<portalid>/config directory. This interval can improve the performance and scalability because the server uses the cached information between scans. The default value is 30 seconds.

dpScanInterval=0

DP scan interval controls how often DP updates done by someone other than the user. For example, Administrator is selected before serving portal request. If a DP document (user or shared) is present in the cache, and DP scan interval has not yet elapsed, then the cached copy will be used and the check for modified DP documents does not happen.

The values are:

- -1 — Scan happens only when user logs in or membership changes.
- 0 — Scan happens for every request.
- n — After n seconds from previous scan, any updates to user DP or shared DP will be reflected in system behavior.

community.contributor.types=jdo

This property tells, which type of DP documents should be present as community request parameter to Desktop when membership is determined for DP merge. Contributor types can be separated by | as in communitymc.properties.

classLoaderRevalidateInterval=3600

Provider class loader re-validate time interval in seconds, for authless anonymous session only.

---

`maxEventGenerations=5`

Maximum number of event generations.

`portletRenderModeParallel=true`

This is the Application Server execution mode. To turn the parallel execution of the portlets in Application Server off, set this parameter to off.

`jspScratchDir=/var/opt/SUNWportal/portals/<portal_id>/desktop/compiled`

JSP Scratch directory is used for placing compiled JSPs.

`jspCompilerWARClassPath=<Used only on application server>`

[Internal]

Used only on application server. For more information on `jspCompileWARClassPath`, see the `desktopconfig.properties` file in the `/var/opt/SUNWportal/portals/<portal_id>/config` directory.

`templateBaseDir=/var/opt/SUNWportal/portals/<portal_id>/desktop/`

Root directory under which all template files are located.

`communityTemplateBaseDir=/var/opt/SUNWportal/portals/<portal_id>communitytemplates`

This is the community template base directory.

`providerClassBaseDir=/var/opt/SUNWportal/portals/<portal_id>/desktop/classes`

Root directory under which the customer is allowed to place provider classes, whether those are overriding the bundled providers, or their own new providers (usually the case). They must be placed in this directory, either in a jar at the top level, or in a com (or whatever) package directory.

`serviceAppContextClassName=com.sun.portal.desktop.context.DSAMEServiceAppContext`

[Internal]





# Search Configuration Properties File

---

This chapter contains the following sections:

- “Overview” on page 105
- “Parameters” on page 105

## Overview

In the default installation, the `search.conf` file is in the `/var/opt/SUNWportal/searchservers/<searchserver_id>/config` directory. The `search.conf` file lists all the specific search values you have set. The `/var/opt/SUNWportal/samples/config` directory contains a sample `search.conf` file.

The default install assigns `$CSROOT` to `/var/opt/SUNWportal/searchservers/search1`, `$CSBIN` to `/var/opt/SUNWportal/bin`, and `$CSLIB` to `/var/opt/SUNWportal/lib`.

## Parameters

The parameters available in the `search.conf` file and their description are:

`search.conf` File Parameters

`csid=x-catalog://$HOST:$PORT/$NICK`

Defined at installation. Server identifier string, mainly for backward compatibility with Search Server.

`bindir=$CSBIN`

Defined at installation. Location of binaries.

`database-directory=$CSROOT/db`

Defined at installation. Location of database (used by server).

`database-root=$CSR00T/db`

Defined at installation. Location of database (used by indexer).

`database-max-concurrent=8`

Limits the number of server threads that can access the database at any one time. You can change this value for performance reasons, but it should be set to about 1.25 times the number of index threads for best performance.

`database-name=default`

The logical database name. You can change this value to another database including an external one.

`database-logdir=db`

Directory where database transaction logs are kept.

`security-mode=OFF`

Enables or disables document level security. Can be reset in the administration console under Server Settings.

`security-manager=com.sun.portal.search.rdmserver.DSameSecurityManager`

Security manager class name. Do not edit.

`security-dsame-group=OFF`

Whether to use group in addition to user role for security control.

`debug-logfile=$CSR00T/logs/rdmserver.log`

Logs internal server activity. Defined at installation. Can be reset in the administration console under Server Advanced Settings.

`debug-loglevel=1`

Sets the default log level. Can be reset in the administration console under Server Advanced Settings.

`filters-check-dns=on`

Checks for number of servers aliased to the same address. Can be reset in the administration console under Robot Simulator.

`filters-check-redirect=on`

Checks for any server redirects. Can be reset in the administration console under Robot Simulator.

`import-config=$CSR00T/config/import.conf`

Defined at installation. Contents generated by the Search server when you define an import agent in the administration console under Database Import.

`libdir=$CSLIB`

Defined at installation.

`logfile=$CSR00T/logs/rdm.log`

Log of RDM server requests. Defined at installation. Can be reset in the administration console under Server Advanced Settings.

---

`disable-rdm-log=false`

Disables RDM request logging. Can be reset in the administration console under Server Advanced Settings.

`classification-stats-during-browse=true`

If true, server records how many documents are found in each browse category.

`browse-root-classification=false`

Whether to browse for documents at the root of the category tree.

`search-logfile=${CSRROOT}/logs/searchengine.log`

Search engine logfile. Defined at installation. Can be reset in the administration console under Server Advanced Settings.

`search-max-index-batch=2000`

Maximum number of documents in each index batch.

`search-query-threads=6`

Number of search query threads. Should be set to 3-6 threads per CPU that you wish to utilize.

`search-index-threads=1`

Number of search index threads. Usually left at 1.

`search-index-type=AWord`

The format of the search engine index. Do not edit.

`search-index-partition-size=32`

The blocking factor used during index merges. Do not edit.

`search-dictionary-type=partial`

Format of the search dictionary. Do not edit.

`search-lookup-limit=-1`

Controls the timeout (milliseconds) of slow wildcard searches.

-1 means unlimited.

`search-highlights=true`

Enable search result highlighting.

`search-max-passages=3`

Maximum number of dynamic summary passages to generate.

`search-passage-context=6`

Size of context (in words) around each highlight passage.

`#search-field-multipliers="title 1.0"`

Search weights assigned to different document fields. Can be a comma separated list.

`rdmgr-logfile=$CSR00T/logs/rdmgr.log`

Log file for the indexer process. Defined at installation. Can be reset in the administration console under Server Advanced Settings.

`schema-description=$CSR00T/config/schema.rdm`

The default Search Engine Schema. Defined at installation.

`server-description=$CSR00T/config/server.rdm`

The RDM server description returned by server description requests. Defined at installation.

`server-root=$CSR00T`

Server instance root directory. Defined at installation. Can be reset in the administration console under Server Settings.

`taxonomy-database-name=taxonomy`

The logical name of the taxonomy index database.

`taxonomy-description-refresh-rate=3600 -> 60`

Polling interval for automatic taxonomy reloads.

`taxonomy-description=$CSR00T/config/taxonomy.rdm`

The RDM Taxonomy definition. Edit using the Category Editor under Categories. Defined at installation.

`tmpdir=$CSR00T/tmp`

Defined at installation. Can be reset in the administration console under Robot Crawling.

`robot-refresh=30000`

Number of milliseconds between refreshes of the Robot Control page of the administration console.

`admin-category_editor_node_size_per_page=25,50,100,250,500,-1`

List of available choices, defining the maximum number of categories displayed per page.

-1 = display all tree.

`admin-category_editor_max_combo_element=10`

Maximum number of elements in the category editor drop down select list of target categories.

The following parameters are not used:

`filters-check-virtual`            Filters Check Virtual.

`multiple-classifications`        Multiple Classifications.

`reports-exclude-gv-queries`       Reports exclude gv queries.

`reports-exclude-browse`          reports exclude browse.

`rdmgr-pidfile`                    rdmgr pidfile.

`rlog-max-logs`                    rlog max logs.

## XML and Schema Files

---

The Sun Java System Portal Server 7.1 registers its services into the Sun Java System Sun Java System Access Manager Service Management Services (SMS) framework. This occurs during the pre-installation of the Portal Server 7.1 and post-installation for Sun Java System Access Manager software.

### Service Management Services

---

**Note** – In general, any service-related data that is not server-specific is stored in the Sun Java System Access Manager directory. Server-specific data can be stored in properties files that are local to the specific server.

---

SMS provides a mechanism for services to define and manage their configuration data by using an Extensible Markup Language (XML) file that adheres to the SMS Document Type Definition (DTD). The definition of the configuration parameters through the XML file is called the schema for the service. Each Portal Server 7.1 service (Desktop, Rewriter, and Search) has its own XML and properties files for presenting and modifying service specific data.

Within the Sun Java System Access Manager framework, Portal Server 7.1 defines services related to the following functional areas:

#### Desktop

The SunPortalDesktopService includes data associated with the Desktop component, including the display profile and other configuration parameters associated with the Desktop.

#### Search Engine

The SunPortalSearchService defines the data associated with the Search component, such as the search person and search instances. One or more instances of Search service instances can be defined.

## Rewriter

The SunPortalRewriterService includes data associated with the Rewriter component, including the named rule sets that control the rewriting operation. The Rewriter API makes reference to the named rule sets that are stored in the directory.

In addition, the Portal Server 7.1 also uses other DTDs to define LDAP attribute values for the display profile and the Rewriter ruleset.

The Display Profile Document Type Definition (DTD) defines how the Display Profile is structured. The underlying data format for a display profile document is XML. It is intended to define the display configuration for the Desktop. It does that by defining provider, portlet, and channel objects, and their properties. The Rewriter ruleset DTD defines the structure of the ruleset. The Rewriter includes a default ruleset.

The file paths for the various XML, DTD, and schema files used to define the services of the Portal Server 7.1 (in the first column) and the service which uses the corresponding file (in the second column) are:

<code>/opt/SUNWam/dtd/sms.dtd</code>	Service Management Services Document Type Definition (DTD)
<code>/opt/SUNWportal/export/service/psDesktop.xml</code>	Portal Server 7.1 Desktop service definition
<code>/opt/SUNWportal/export/service/psRewriter.xml</code>	Portal Server 7.1 Rewriter service definition
<code>/opt/SUNWportal/export/service/psSearch.xml</code>	Portal Server 7.1 Search service definition
<code>/var/opt/SUNWportal/dtd/psdp.dtd</code>	Display Profile document type definition (DTD)
<code>/opt/SUNWportal/web-src/WEB-INF/lib/rewriter.jar</code>	Rewriter Ruleset document type definition (DTD) under resources/RuleSet.dtd
<code>/opt/SUNWportal/web-src/WEB-INF/lib/rewriter.jar</code>	Default ruleset under resources/DefaultRuleSet.xml
<code>/opt/SUNWportal/export/ldif</code>	ldif file available in desktop directory.
<code>/opt/SUNWportal/export/ldif/psDesktop.ldif</code>	Portal Server 7.1 Desktop Schema
<code>/opt/SUNWportal/export/ldif/psSearch.ldif</code>	Portal Server 7.1 Search Schema

PART III

Display Profile

- Chapter 13
- Chapter 14
- Chapter 15
- Chapter 16
- Chapter 17
- Chapter 18
- Chapter 19
- Chapter 20
- Chapter 21
- Chapter 22
- Chapter 23
- Chapter 24
- Chapter 25





# Introduction to Display Profile

---

This chapter contains the following sections:

- [“What is Display Profile?” on page 113](#)
- [“Administering the Display Profile” on page 114](#)

## What is Display Profile?

The display profile is a series of XML documents describing container management and properties for providers and channels. The display profile creates the display configuration for the Desktop by defining the following items in the XML document:

### Provider definition

Specifies the name and the Java class for the provider. A provider is a template used to generate content, which is displayed in the channel. See [“Provider Object” on page 120](#) for more information.

### Channel definition

Specifies the run-time configuration of an instance of the provider class. 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. See [“Channel Object” on page 119](#) for more information.

The container channel properties include the display definition about 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 (minimized or detached).

Provider and channel property definitions

Specify the values for provider and channel properties. Properties defined in a provider usually specify default values for the channels that are derived from the provider. The display configurations for the channels include properties such as the title, description, channel width, and so on. The properties defined in the channel usually specify the specific value for that channel that is different from the default value. See “[Property object](#)” on page 121 for more information.

---

**Note** – If a property is not defined in the channel, then the default value for the property as defined in the provider is used. If a property is defined in the channel, then the value for the property defined in the provider is ignored.

---

The display profile does not actually define the overall layout or organization of what users see on their Desktops. The display profile exists only to provide property values for channels. However, the display profile does indirectly control some aspects of channel presentation, such as column layout for a table container or how the table container draws channels in a table.

## Administering the Display Profile

You can use the Portal Server Console and `psadmin` to administer the Portal Desktop Service data and the Display Profile. In the Portal Server console, you can upload, download, and remove the display profile from a specific dn. You can edit the desktop service attributes. You can modify channels and containers properties. You can also create, edit, and remove channels in the containers. The `psadmin` provides commands, which allow you to list, add, remove, modify, and merge display profiles. For more information, see the *Sun Java System Portal Server 7.1 Administration Guide*.

# Display Profile Document

---

This chapter contains the following sections:

- “Document Structure” on page 115
- “How are the Display Profile XML Documents Stored?” on page 116
- “Types of Display Profile Documents” on page 116

## Document Structure

This chapter describes the overall structure of the display profile documents. The underlying data format for a display profile document is XML.

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 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.

The following sections describe the display profile objects in detail.

## How are the Display Profile XML Documents Stored?

Display profile documents are stored in their entirety as a single attribute in the Sun Java System Sun Java System Access Manager software services layer. Potentially, the 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.

The user's display profile is a series of XML documents describing container management and properties for channels. (One display profile document is equivalent to one XML document.) The 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.

To change display profile property values, the providers use the provider APIs (PAPI) to get and set the values. When the channel values are set to the display profile, the PAPI internal implementation uses the Sun Java System Access Manager SDK to set the display profile document in the Sun Java System Access Manager software Desktop service attribute.

---

**Note** – Though possible, you should not edit the display profile using the Sun Java System Access Manager SDK.

---

## Types of Display Profile Documents

This section explains the different types of display profile documents and how to use the Sun Java System Portal Server console to administer them.

### Global Display Profile Document

Defines display profile elements that are inherited by all users on the system, regardless of the organization or role to which they belong. (Although currently not enforced, you might also want to use the display profile XML document to define the common providers that will be used by everyone.)

### Dynamic Display Profile Document

Describes container management and properties for channels. This display profile is not 'used' to generate a user's Desktop at runtime, but becomes the default for each newly created organization and role. By default, the dynamic display profile document is blank. To use the dynamic display profile, you need to first populate it.

#### Organization, Suborganization, or Role Display Profile

Shows the display profile for the selected organization, suborganization, or role. When you create a new organization, suborganization, or role, you create a template for this entity. When you create the template for the Desktop service, the initial display profile is set to the dynamic display profile document as mentioned above. Thus, if the dynamic display profile is blank, nothing is filled in.

Most likely, you use this display profile document to customize container management and channel properties to fit the needs of different organizations and roles.



# Display Profile Objects

---

This chapter contains the following sections:

- “Introduction” on page 119
- “Object Lookup” on page 122

## Introduction

There are three basic types of objects in the display profile: channels, providers, and properties. Every object in the display profile is associated with a DN. 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 display profile XML documents.

## 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.

EXAMPLE 15-1 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:///var/opt/SUNWportal/portals
      /<portal_id>/desktop/
```

**EXAMPLE 15-1** Example Channel Object XML Syntax (Continued)

```

        default/SampleXML/getQuotes.xml"/>
    <String name="xslFileName" value="/var/opt/SUNWportal/
        portals/<portal_id>/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.

**EXAMPLE 15-2** 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-fontFace1" value="Sans-serif"/>
        <Collection name="categories">
            <String value="Personal Channels"/>
            <String value="Sample Channels"/>
            <String value="News Channels"/>
        </Collection>
        ...
    </Properties>

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

</Container>

```

**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="providerClass">`.



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 stores provider definitions that are available to the channel and containers to implement their content generation behavior. 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 class that implements the provider's behavior is defined in the provider attribute. Channels use the name attribute values to refer to the provider.

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.

#### EXAMPLE 15-3 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="color" value="blue"/>
    <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 Java System Portal Server"/>
    <String name="url" value="file:///var/opt/SUNWportal/portals/<portal_id>/
      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="urlScraperRulesetID" value="default_ruleset"/>
    <Boolean name="cookiesToForwardAll" value="true"/>
    <Collection name="cookiesToForwardList">
  </Collection>
  </Properties>
</Provider>
```

## 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.

## Object Lookup

At runtime, the system never asks for properties directly from a provider. The request always goes to a channel. If a Java provider object requests a property, it searches the display profile in the following order until it finds the property, or until it reaches the top of the containment hierarchy:

### ▼ To Perform Object Lookup:

- 1 Channel's properties
- 2 Channel's provider's properties
- 3 Channel's parent's properties
- 4 Channel's parent's provider's properties
- 5 Channel's parent's properties (and so on)
- 6 The global properties bag defined in the display profile root definition

Therefore, when a channel asks for the names of its properties, it gets the set of the union of all the above.

Properties that exist in a provider object are intended to have the semantics of default values for the channel. For example, for a provider XML that defines property title, all channels that are derived from provider XML inherit the title property. If the channel wants to override this property, it can set the value within its own properties.

# Document Priorities

---

This chapter contains the following sections:

- “Overview” on page 123
- “Examples” on page 124
- “Summary” on page 126

## Overview

At runtime, when a user logs in, the system determines the set of documents that makes up the user’s display profile document set. The Desktop internal implementation of the display profile (the part that interprets the display profile) determines this set by looking at all of the LDAP nodes that the user belongs to. This can be the organization DN (`dc=sesta.com`), suborganizations, role DNs (`cn=Role1,dc=sesta.com`), and uid (`uid=user,ou=People,cn=Role1,dc=sesta.com`), as well as the global display profile. The display profile documents from each of these LDAP nodes and global display profile are then read (if it exists there), and all of the documents are put into a set. The system sorts the set according to the document priorities. A lower number represents a lower priority. For example, a 1 is a lower priority than a 2. The documents are then sorted from lower number to higher number. See “[Process of Merging](#)” on page 129 for more information on this process.

The user level document (`uid=admin,ou=People,...`) is a special case referred to as the base document. Think of the base document as a priority equal to infinity. Thus, it is always the highest number (and hence highest priority). All of the mergers are associated with the base document in sorted order, and the priority setting on a user document is always the highest. The priority attribute used in the `<DisplayProfile>` tag takes the special keyword `user` to indicate that the current display profile is the user level display profile.

When a merge occurs, it starts at the lowest priority document (lowest number) and proceeds in increasing priority number, until it arrives at the user (base) document.

Thus, the implication of display profile document priorities is that what really matters is the priority number. For example, an organization level document can have a higher priority than a role level document, but it does not have to. It depends on how you need to prioritize these documents for your site.

Specify the display profile document priority in the XML file with the `<DisplayProfile priority= syntax>` tag. You can change the priority by directly editing the display profile XML by using the Sun Java System Portal Server console or by using the `psadmin` command to load the display profile.

---

**Note** – Do not assign the same priority to two display profile documents. Doing so causes the Desktop to not appear properly. However, the product does not check for duplicate document priorities.

---

## Examples

The examples provided below shows the display profile documents for organization and a bill.

### Example 1

This example uses two display profiles, one for the organization example and one for the uid bill. When *Bill* logs in (`uid=bill`) to the Desktop, the bookmark channel titled “*Bill’s* Bookmarks” is displayed with the following three bookmarks (in that order):

- ACME
- Amazon
- EBay

**EXAMPLE 16-1** Display Profile Document for the Organization (`dc=acme.com`)

```
<DisplayProfile version="1.0" priority="10">
...
<Channel name="Bookmark" provider="BookmarkProvider" merge="fuse">
  <Properties>
    <String name="title" value="My Bookmarks" merge="replace"
      lock="false" propagate="true"/>
    <String name="refreshTime" value="600" merge="replace"
      lock="false" propagate="true"/>
    <Collection name="targets" merge="fuse" lock="false"
      propagate="true">
      <String value="ACME home page|http://www.acme.com"
        merge="replace"
        lock="false" propagate="true"/>
    
```

**EXAMPLE 16-1** Display Profile Document for the Organization (dc=acme.com) (Continued)

```

    </Collection>
  </Properties>
</Channel>
...
</DisplayProfile>

```

Display Profile Document for the uid=Bill,ou=people,o=acme.com

```

<DisplayProfile version="1.0" priority="10"> ... <Channel name="Bookmark"
provider="BookmarkProvider" merge="fuse"> <Properties> <String name="title"
value="Bill's Bookmarks" merge="replace" lock="false" propagate="true"/>
<Collection name="targets" merge="fuse" lock="false" propagate="true">
<String value="Amazon|http://www.amazon.com" merge="replace" lock="false"
propagate="true"/> <String value="EBay|http://www.ebay.com" merge="replace"
lock="false" propagate="true"/> </Collection> </Properties> </Channel> ...
</DisplayProfile>

```

## Example 2

This example uses three display profiles, the global display profile, the display profile for the organization acme, and the display profile for the role hradmin. When the user who is assigned to the hradmin role logs in to the Desktop, the JSPTTableContainer appears with the following channels:

- SampleSimpleWebService

**EXAMPLE 16-2** Global Display Profile Document

```

<DisplayProfile version="1.0" priority="0">
...
<Container name="JSPTTableContainer" provider="JSPTTableContainerProvider" merge="fuse">
  <Properties>
    ...
  </Properties>
  <Available>
    ...
  </Available>
  <Selected merge="fuse" lock="false">
    <Reference value="UserInfo"/>
  </Selected>
  <Channels/>
</Container>
...
</DisplayProfile>

```

Display Profile Document for the Organization (dc=acme.com)

**EXAMPLE 16-2** Global Display Profile Document *(Continued)*

```
<DisplayProfile version="1.0" priority="0"> ...
  <Container name="JSPTTableContainer"
provider="JSPTTableContainerProvider" merge="fuse">
<Properties> ... </Properties>
<Available> ... </Available> <Selected merge="fuse"
lock="false"> <Reference value="UserInfo"/>
<Reference value="Notes"/> </Selected> <Channels/>
</Container> ... </DisplayProfile>
```

## Display Profile Document for the hradmin Role

```
<DisplayProfile version="1.0" priority="0">
...
<Container name="JSPTTableContainer" provider="JSPTTableContainerProvider" merge="fuse">
  <Properties>
    ...
  </Properties>
  <Available>
    ...
  </Available>
  <Selected merge="fuse" lock="true">
    <Reference value="SampleSimpleWebService"/>
  </Selected>
  <Channels/>
</Container>
...
</DisplayProfile>
```

## Summary

A display profile document has a low or high priority depending on whether you consider the merge order or the ability to lock as the defining factor.

Without considering locking, the lower numbered display profile document has a lower priority. The lower numbered display profile document gets merged first so the value of a higher priority document overrides the value of a lower priority document. In this sense, the lower numbered document has a lower priority.

However, the lower numbered display profile document can also lock an object so it cannot be affected by a higher numbered document. In this sense, the lower numbered document has a higher priority.

# Merge Semantics

---

This chapter contains the following sections

- “Introduction” on page 127
- “Process of Merging” on page 129
- “Merge Locking” on page 134

## Introduction

The display profile is composed of a hierarchy of XML documents. The software can store a display profile document for the user, each role the user belongs to, and the user’s organization or suborganization. At runtime, the system merges these multiple display profile documents to deliver a particular portal desktop to the user. This process of merging display profile documents affects the final display profile by potentially changing channel, provider, and property definitions.

The display profile data format contains syntax that defines how these documents are combined. This definition is commonly known as merge semantics.

Merge semantics control how attributes are combined as display profile documents from different LDAP nodes (base DN, DN, and role DNs) which are merged to form a single representation (that is, Desktop). Merge semantics assume an ordering to display profile documents. The Sun Java System Portal Server 7.1 software imposes an additional ordering on Sun Java System Access Manager software roles to simulate a hierarchical structure.

The set of display profile documents for a user consists of: the documents that exist at the user’s LDAP organization and suborganization nodes; the documents that exist at each of the user’s role nodes; and the document that exists at the user’s entry node. Documents do not need to be defined at each of these nodes, but there must be at least one document defined at a node. The set of documents is sorted according to a priority value that the display profile document defines. See Display Profile Document Priorities for more information.

You can visualize the process of document merging as laying one display profile document on top of another. A merge happens where like named channels, providers, and properties fall on top of one another. Merging is based on the name of the display profile object, not the XML structure defined in the display profile document. Like named channels can exist in different containers within the containment hierarchy in the display profile to be merged.

For example, Figure 18–1 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.



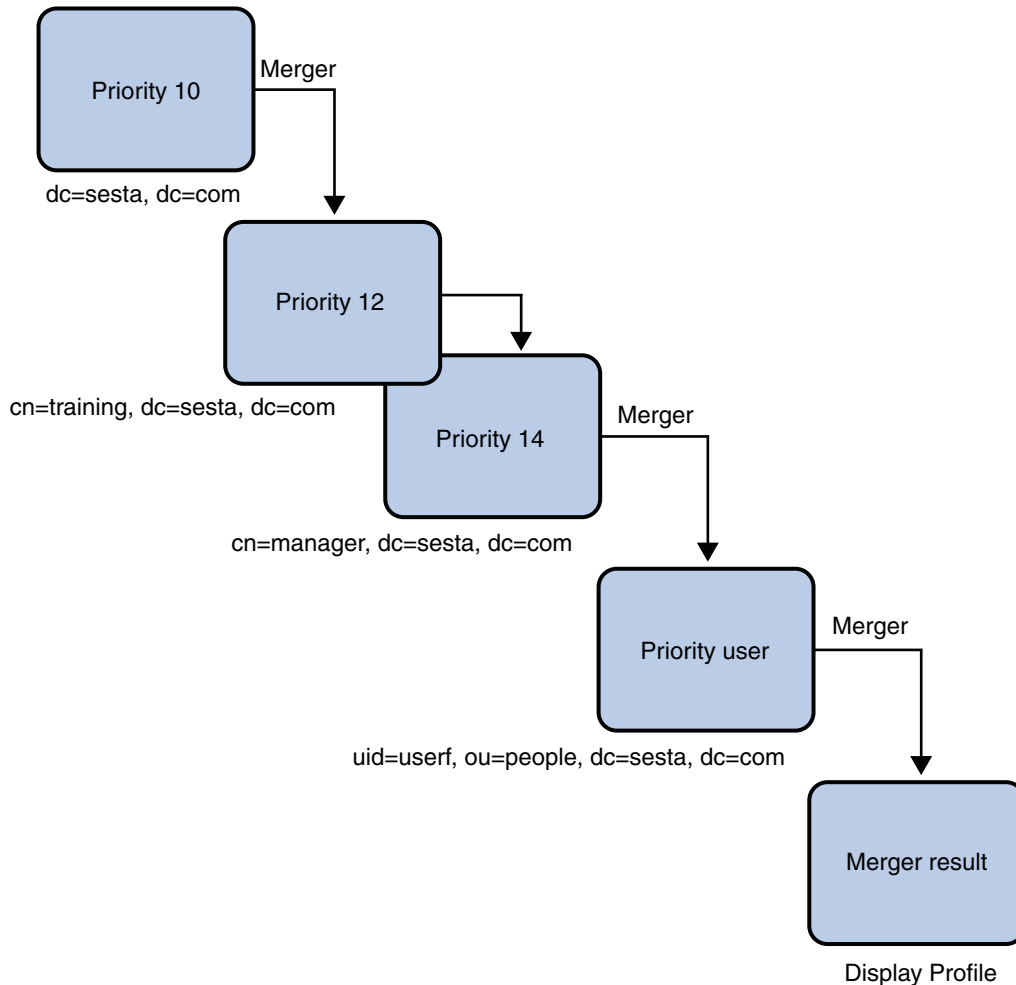


FIGURE 17-1 Example of Merge Process

## Process of Merging

When a user logs in to the Portal Server 7.1 Desktop, and after authentication takes place, the system determines the user's display profile by:

### ▼ To Determine the User Display Profile:

- 1 Locating all the display profile documents for that user by searching through the global display profile, and LDAP organization, suborganization, role, and user nodes that the user belongs to.

- 2 Placing the retrieved display profile documents in a temporary area, which can be visualized as a bag.
- 3 Sorting the display profile documents in the bag based on priority, starting at the lowest priority. (The node at which the document was retrieved does not influence the priority sorting. Also, the user display profile document always has the highest priority.)
- 4 Taking the documents out of the bag, lowest priority first, then placing the next higher level priority document over this document, and applying merge and lock semantics.
- 5 Continuing [“Process of Merging” on page 129](#) until all the documents have been taken out of the bag so that the system returns a value to the user that is a merge of the objects found in the documents.

## Types of Merge

This section explains the types of merge available to combine the display profile documents.

### Overview

Display profile uses the following three types of merges to determine how to combine display profile documents:

replace

All the display profile objects defined in the higher priority document completely override the ones defined at the lower one. If the object does not exist in the lower priority document, it is added to the merge result (the object replaces the value in the merge results).

remove

The named object is removed from the merge up to this point (the object is removed from the merge results). It no longer exists in the display profile (but it can be re-introduced by another document to be merged). It can be redefined by a higher priority document.

fuse

The object from the lower priority document is combined with one from the higher priority document (the object is merged with the value in the merge results).

---

**Note** – The exact meaning of each merge type depends on the display profile object they are applied to.

---

For channels and providers, fuse has special meaning. The channels themselves are not actually fused together. Rather, fuse indicates that the channel's or provider's properties should be combined. The replace semantic replaces the entire channel or provider, including all properties. The remove semantic removes the entire channel or provider from the merge up to that point.

The display profile <DisplayProfile> root node can also have merge semantics. The replace semantic means that all the DP objects defined in the higher priority document completely override the ones defined at the lower one. All merges up to that point are negated and the higher priority document is used as the new base for merging. The remove semantic indicates that all merge results up to the point of this document are to be discarded. The merge begins with the next display profile document found in the sorted set. As with channels and providers, the fuse semantic means that the contained objects (channels and providers) should be combined.

Atomic display profile properties (those that cannot contain other properties) cannot use the fuse semantic. This includes the String, Integer, Boolean, and Reference properties.

The set of properties for a channel consists of the channel's properties plus the channel's provider's properties plus the channel's parent's properties, and so on. You can think of this total set of properties as the channel's single document properties. An implication of document merging is that the total set of properties for a document consists of the set union of the channel's single document properties for all documents in the user's merge set.

## Examples

This section provides examples for merge types such as remove, replace, and fuse.

### Remove Example

This example uses the merge type remove to modify a container's selected channel list.

The following example code shows how the merge sets of all users can consist of an organizational level document that contains the following display profile fragment.

```
<Container name="JSPTableContainer" provider="JSPTableContainerProvider" merge="fuse">
  <Properties> ... </Properties>
  <Available> ... </Available>
  <Selected merge="fuse">
    <Reference value="UnixTipoftheDay"/>
  </Selected>
</Container>
```

The “unix tip of the day” describes ways to use UNIX. It is likely that users that belong to the admin role would not find this channel helpful. To remove this channel from everyone with the admin role, define the TemplateTableContainer channel in the admin role document as follows:

```
<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="UnixTipoftheDay" merge="remove"/>
  </Selected>
</Container>
```

The preceding sample snippet causes the `<Reference value="UnixTipoftheDay">` to be removed from the admin role display profile.

## Replace Example

This example uses the merge type replace to remove channel from all users’ display.

The following example shows how for a particular container, a role admin can ignore all of the channels defined in the organization level. The organization definition resembles the following:

```
<Container name=...>
  ...
  ...
  <Selected>
    <Reference name="X"/>
    <Reference name="Y"/>
    <Reference name="Z"/>
  </Selected>
</Container>
```

Because the role admin does not want any of the users under that role to have the X, Y, or Z channels, the container is defined as follows:

```
<Container name=...>
  ...
  ...
  <Selected merge="replace">
    <Reference name="A"/>
    <Reference name="B"/>
    <Reference name="C"/>
  </Selected>
</Container>
```

The selected list in the role document's container replaces the selected list in the organization document's container.

## Fuse Example

This example uses the merge type fuse to create role-based channel list.

Use the fuse merge semantic to combine non-atomic display profile objects. These objects include Collection and the available or selected channel lists. Here, fuse indicates that all the properties contained in the non-atomic property should also be merged. Using fuse in this way enables the final non-atomic property presented to the user to be build up from various documents.

The following example display profile documents are for a user who belongs to the admin, employee, and movieFreak roles. The selected channels for the user appear at the end.

### EXAMPLE 17-1 Display Profile for the Admin Role

```
<Container name="JSPTTableContainer" provider="JSPTTableContainerProvider" merge="fuse">
  <Properties> ... </Properties>
  <Available> ... </Available>
  <Selected merge="fuse">
    <Reference value="Outages"/>
    <Reference value="SolarisAdmin"/>
    <Reference value="AdminTipoftheDay"/>
  </Selected>
</Container>
```

### Display Profile for the Employee Role

```
<Container name="JSPTTableContainer" provider="JSPTTableContainerProvider" merge="fuse">
  <Properties> ... </Properties>
  <Available> ... </Available>
  <Selected merge="fuse">
    <Reference value="Benefits"/>
    <Reference value="EmployeeNews"/>
  </Selected>
</Container>
```

### Display Profile for the movieFreak Role

```
<Container name="JSPTTableContainer" provider="JSPTTableContainerProvider" merge="fuse">
  <Properties> ... </Properties>
  <Available> ... </Available>
  <Selected merge="fuse">
    <Reference value="NewMoviesReleases"/>
    <Reference value="MovieShowTimes"/>
  </Selected>
```

EXAMPLE 17-1 Display Profile for the Admin Role (Continued)

```
</Container>
```

The resultant list of selected channels for the user is as follows, with the available channel list ordered in the same way that the merging was applied, from lower to higher priority:

```
<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"/>
    <Reference value="NewMoviesReleases"/>
    <Reference value="MovieShowTimes"/>
  </Selected>
</Container>
```

## Merge Locking

Any display profile object that is able to be merged can also be locked. When an object is locked, it cannot be affected by merge semantics in higher priority documents. This enables low-priority documents to prevent a high-priority document from using the merge semantics to change particular aspects of the display profile.

## Examples

### Example 1

This example demonstrates how to use the merge lock feature to force property value for all users. The example shows how to ensure that for a particular organization, all users see the “employee news” channel. The users cannot remove this channel from their display. At the organization level document, the container channel’s selected list is defined as follows:

```
<Selected merge="fuse">
  ...
  <Reference value="EmployeeNews" lock="true"/>
  ...
</Selected>
```

### Example 2

This example demonstrates how to use the merge lock feature to forcibly remove channel from all users' display. The example shows how to force the "online games" channel to be removed. In this scenario, users have added this channel to the selected channels list in their user document, so simply removing it from the organization level document's selected channel's list will not work. Instead, the employee and organization lists will be merged together resulting in the "online games" channel being present. To forcibly remove the channel from all users under the organization, the selected channels list is defined as follows:

```
<Selected merge="fuse">
  ...
  <Reference value="OnlineGames" merge="remove" lock="true"/>
  ...
</Selected>
```

Here, the remove semantic removes the channel from merged result, and lock prevents lower priority documents from merging the value back in.





# Display Profile Properties: Overview

---

This chapter contains the following sections:

- “Introduction” on page 137
- “Display Profile Properties” on page 138
- “Display Profile Property Types” on page 139

## Introduction

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.

The sample portal makes use of the following display profile definitions in the *PortalServer-base/SUNWportal/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-anon.xml</code>	Contains the display profile definitions for channels and containers for the authlessanonymous and anonymous users in the default organization.

## 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 can associate properties with the following display profile objects:

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

There are four 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 46 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:

- They define a property template or schema, defining the properties that will be used by all channels based on the provider.
- 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.

---

## Container

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. 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 [Chapter 17](#) for a complete discussion of the semantics of the display profile merging.

## Display Profile Property Types

This section 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 18-1 Display Profile General Property Types

Property Type	Definition	Example
Boolean	An atomic object representing a Boolean value.	<code>&lt;Boolean name="removable" value="true"/&gt;</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>&lt;Collection name="channelsRow"&gt; &lt;String name="App" value="5"/&gt; &lt;/Collection&gt;</code>

**TABLE 18-1** 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>&lt;ConditionalProperties condition="locale"&gt; &lt;String name="en_US" value="English (United States)"/&gt; &lt;/ConditionalProperties&gt;</pre>
Integer	<p>An atomic object representing an integer value.</p>	<pre>&lt;Integer name="numberOfHeadlines" value="7"/&gt;</pre>
Reference	<p>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.</p>	<pre>&lt;Reference value="UserInfo"/&gt;</pre>
String	<p>An atomic object representing a string value.</p>	<pre>&lt;String name="title" value="Table Container Channel 1"/&gt;</pre>

# Display Profile Properties: 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, this chapter 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.

## Example for Global Properties Sample

EXAMPLE 19-1 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>
</DisplayProfile>
```

EXAMPLE 19-1 Global Properties Sample in the Display Profile `dp-org.xml` File (Continued)

```
</Properties>
```

The list of all global attributes available with the default installation of the sample portal and their description are:

<code>GlobalThemes</code>	<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>
<code>UserTheme</code>	<p>Defines the theme that shows up in the user’s Desktop. The value must be one of the collection values defined in <code>GlobalThemes</code>. In <code>dp-org.xml</code> file, the value can be either <code>theme1</code> or <code>theme2</code>. When users customize their Desktops, the value <code>UserTheme</code> will change.</p>
<code>docroot</code>	<p>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.</p>
<code>helpURL</code>	<p>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.</p>
<code>userDefinedChannels</code>	<p>Specifies the page to allow users to Create New Channel.</p>

# Display Profile Properties: Container Provider Properties

---

This chapter contains the following sections:

- “Introduction” on page 143
- “Available and Selected List” on page 144
- “Common Properties for Table Container” on page 145
- “Common Properties for Tab Container” on page 147
- “Other Container Properties” on page 148

## Introduction

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 Java System Portal Server software.

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` - `JSPTableContainerProvider` is an extension of `JSPPProvider`. This JSP table provider displays the content channels in a table.
- `JSPTabContainerProvider` - `JSPTabContainerProvider` is an extension of `JSPPProvider`. This tab container provider displays a channel that is made up of a number of tabs with titles on them. By default, the `JSPTabContainerProvider` uses `JSPTableContainer` to lay out content for each tab. However, it can use `JSPTableContainer`, `JSPSingleContainer`, or `JSPTabContainer` to layout content for each tab.
- `JSPSingleContainerProvider` - `JSPSingleContainerProvider` is an extension of the JSP container provider. The single container provider displays one channel in it.

See the Javadocs for more information on these containers.

## 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 list of required container properties and their description are:

**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="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="App"/>
<Reference value="Bookmark"/>
</Selected>
```



## Common Properties for Table Container

The following table lists the common properties for table containers and their description.

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

TABLE 20-1 Table 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>	
<code>layout</code>	Defines the width of the table columns. Layout one (1) refers to thin-thick, layout two (2) refers to thick-thin, and layout three (3) refers to thin-thick-thin.
<code>thin_popup_height</code>	Defines the window height in pixels for the thin channel in the detached window.
<code>thin_popup_width</code>	Defines the window width in pixels for the thin channel in the detached window.
<code>thick_popup_height</code>	Defines the window height in pixels for the thick channel in the detached window.
<code>thick_popup_width</code>	Defines the window width in pixels for the thick 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>defaultchannelsIsMaximizable</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.

TABLE 20-1 Table Container Properties (Continued)

Property Tag	Description
<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>isRemovable</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.
<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.

TABLE 20-1 Table Container Properties (Continued)

Property Tag	Description
<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 border-less 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 border-less 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 list of the properties common to all `TabContainerProviders` and their description are:

<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.

`TabProperties` The collection property `TabProperties` creates the new tab. There needs to be a one-to-one mapping between the contents of the `TabProperties` 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 `TabProperties` collection.

## Other Container Properties

The list of properties that are common to all container providers and their description are:

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

# Display Profile Properties: Leaf Building-Block Provider Properties

---

This chapter 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” on page 149
- “URLScrapperProvider” on page 150
- “XMLProvider” on page 151

## JSPProvider

JSPProvider uses JSPs. 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 Java System Portal Server 7.1 Developer's Guide* for more information on how JSPProvider uses these JSPs. See also [Chapter 27](#) for the complete list of JSPProvider Communication channels.

If you need to make other customizations, you do so in the JSP files themselves. The list of properties specific to JSPProvider and their description are:

<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 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.

## 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 the content received contains the presentation markup (if applicable).

The list of the properties specific to URLScrapperProvider and their description are:

`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 URLScrapperProvider 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 URLScrapperProvider cannot be turned on (set to `true`) as this channel is, by default, 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 Portal Server Developer's Guide for more information on extending the 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 Java System Web Server software.

---

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

---

The list of properties specific to XMLProvider and their description are:

String name="url"

Specifies the URL that XMLProvider is to transform.

String name="xslFileName"

Specifies the path to the local file to be used as the XSL style sheet.

The provider code tries to pick up the XSL file either from the XML channel directory (that is, `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/SampleXML`), or if not specified here, from the XML provider directory (`/var/opt/SUNWips/desktop/default/XMLProvider/xml`).

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.

In the URLScrapperProvider, the attributes added to the providerDP are:

```
<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="editType" value="edit_subset" advanced="true"/>
<Boolean name="enableUBT" value="false" advanced="true"/>
```

```
<String name="urlScrapperRulesetID" value="default_ruleset"/>
<String name="width" value="thick"/>

<Collection name="cookiesToForwardList">
</Collection>
  <Integer name="timeout" value="100"/>
  <String name="formData" value="" advanced="true" />
  <Boolean name="isHttpAuth" value="false" advanced="true" />
  <String name="loginUrl" value="" advanced="true" />
  <String name="loginFormData" value="" advanced="true" />
  <String name="uid" value="" advanced="true" />
  <String name="password" value="" advanced="true" />
</Properties>
</Provider>
```

Copy the following code, which is the URLScrapperProvider property from Portal Server 7.0 to Upgrade install.

```
<Provider advanced="false" class="com.sun.portal.providers.
urlscrapper.URLScrapperProvider" container="false"
lock="false" merge="fuse"
name="URLScrapperProvider" version="2">
  <Properties advanced="false" lock="false" merge=
  "fuse" name="_properties"
  propagate="true">
    <String advanced="false" lock="false" merge=
    "replace" name="title"
    propagate="true" value="UrlScrapper Channel"/>
    <String advanced="false" lock="false" merge="
    replace" name="description"
    propagate="true" value="This is a test for
    urlscrapper"/>
    <Boolean advanced="true" lock="false" merge=
    "replace" name="isEditable"
    propagate="true" value="false"/>
    <Boolean advanced="true" lock="false" merge=
    "replace" name="isTopLevel"
    propagate="true" value="false"/>
    <String advanced="true" lock="false" merge=
    "replace" name="editType"
    propagate="true" value="edit_subset"/>
    <Boolean advanced="true" lock="false" merge=
    "replace" name="enableUBT"
    propagate="true" value="false"/>
    <String advanced="false" lock="false"
    merge="replace"
    name="urlScrapperRulesetID" propagate="true"
    value="default_ruleset"/>
```



```
<String advanced="false" lock="false" merge="
replace" name="width"
propagate="true" value="thick"/>
<String advanced="true" lock="false" merge="
replace" name="refreshTime"
propagate="true" value="0"/>
<String advanced="true" lock="false" merge="
replace" name="helpURL"
propagate="true" value="en/desktop/urllscrpr.htm"/>
<String advanced="false" lock="false" merge="
replace" name="url"
propagate="true" value="" />
<String advanced="false" lock="false" merge="
replace" name="fontFace1"
propagate="true" value="Sans-serif"/>
<String advanced="false" lock="false" merge="
replace" name="productName"
propagate="true" value="Sun JavaTM System
Portal Server 7"/>
<Boolean advanced="false" lock="false"
merge="replace"
name="cookiesToForwardAll" propagate="true"
value="true"/>
<String advanced="false" lock="false" merge="
replace"
name="inputEncoding" propagate="true" value="UTF-8"/>
<Collection advanced="false" lock="false"
merge="fuse"
name="cookiesToForwardList" propagate="true"/>
<Integer advanced="false" lock="false" merge="
replace"
name="timeout" propagate="true" value="100"/>
<String advanced="true" lock="false" merge="
replace"
name="formData" propagate="true" value="" />
<Boolean advanced="true" lock="false" merge="
replace"
name="isHttpAuth" propagate="true" value="false"/>
<String advanced="true" lock="false" merge="
replace" name="loginUrl"
propagate="true" value="" />
<String advanced="true" lock="false" merge="
replace"
name="loginFormData" propagate="true" value="" />
<String advanced="true" lock="false" merge="
replace" name="uid"
propagate="true" value="" />
<String advanced="true" lock="false" merge="
```

```
"replace" name="password"
propagate="true" value=""/>
<ConditionalProperties advanced="false"
condition="client" lock="false"
merge="fuse" propagate="true" value="HTML">
  <ConditionalProperties advanced="false"
condition="locale" lock="false"
merge="fuse" propagate="true" value="en">
    <String advanced="true" lock="false"
merge="replace" name="helpURL"
propagate="true" value="en/desktop/urlscrpr.htm"/>
    <String advanced="false" lock="false"
merge="replace" name="url"
propagate="true" value=""/>
  </ConditionalProperties>
  <String advanced="true" lock="false"
merge="replace" name="helpURL"
propagate="true" value="en/desktop/urlscrpr.htm"/>
  <String advanced="false" lock="false"
merge="replace" name="url"
propagate="true" value=""/>
</ConditionalProperties>
<ConditionalProperties advanced="false"
condition="locale" lock="false"
merge="fuse" propagate="true" value="en">
  <String advanced="false" lock="false"
merge="replace" name="title"
propagate="true" value="UrlScrapper Channel"/>
  <String advanced="false" lock="false" merge=
"replace" name="description"
propagate="true" value="This is a test for
urlscraper"/>
</ConditionalProperties>
</Properties>
</Provider>
```

# Display Profile Properties: Service Provider Properties

---

Service providers provide search service. The Sun Java System Portal Server 7.1 software includes the following service providers:

- “SearchProvider” on page 155
- “DiscussionsProvider” on page 156
- “SubscriptionsProvider” on page 158

## SearchProvider

SearchProvider supplies the search function using the Sun Java System 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. For more information, see Chapter 11, “Customizing the Service Providers,” in *Sun Java System Portal Server 7.1 Developer Sample Guide*.

### Browse

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

Search results are displayed based on the categorySearch and viewHits properties. The list of properties specific to SearchProvider and their description are:

<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.
<code>Boolean name="categorySearch"</code>	Specifies the category search to be displayed by default. If set to false, category matches are not displayed.

## DiscussionsProvider

The DiscussionsProvider 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/SUNWportal/searchservers/search1/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.

The list of properties specific to `DiscussionsProvider` and their description are:

<code>searchServer</code>	Specifies the path to the search server. By default, the value is <code>search1/search</code> .
<code>dbname</code>	Specifies the discussion database where discussions are stored. Any valid database can be specified.
<code>viewHits</code>	Specifies the number of discussions to display on the main discussion page (full view).
<code>defaultDiscussionDisplay</code>	Determines how the comment subtree is displayed. It can be set to <code>flat</code> or <code>threaded</code> to allow the comment subtree to be displayed as <code>flat</code> or <code>threaded</code> .
<code>defaultFilter</code>	Specifies the filter for searching and displaying discussions and this controls display of the subtree. It can be based on ratings such as <code>irrelevant</code> , <code>routine</code> , <code>interesting</code> , <code>important</code> , or <code>must read</code> . By default, its value is <code>irrelevant</code> ; so all comments rated <code>irrelevant</code> and above are displayed. The <code>Must read</code> filter will highlight the highly rated comments.
<code>defaultExpansionThreshold</code>	It can be set to <code>expand all</code> or <code>collapse all</code> . By default, its value is set to <code>collapse all</code> . If set to <code>expand all</code> , 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 author</code> .
<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 <code>irrelevant</code> , <code>routine</code> , <code>interesting</code> , <code>important</code> , or <code>must read</code> . This property is not

used in this release.

## **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.

## Display Profile Properties: Content Provider Properties

---

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

- “AddressBookProvider, LotusNotesAddressBookProvider, and MExchangeAddressBookProvider” on page 159
- “AppProvider” on page 160, “BookmarkProvider” on page 161, “LoginProvider” on page 163 and “NotesProvider” on page 164
- “CalendarProvider, LotusNotesCalendarProvider, and MExchangeCalendarProvider” on page 161
- “IMProvider” on page 162
- “SimpleWebServiceProvider” on page 164 and “SimpleWebServiceConfigurableProvider” on page 165

### AddressBookProvider, LotusNotesAddressBookProvider, and MExchangeAddressBookProvider

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

The list of properties specific to AddressBookProvider, LotusNotesAddressBookProvider, and MExchangeAddressBookProvider and their description are:

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 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="sunPortalABSorderByOrderSelectOptions"

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

## AppProvider

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

The list of properties specific to AppProvider and their description are:

String name="windowPref" Specifies how to launch a link. The possible values are:

- 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 application links in *name* | URL format, where *name* should match the entry in the userApps collection.

Collection name="userApps" Specifies the list of applications that appear in the applications channel.



## BookmarkProvider

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

The list of properties specific to BookmarkProvider and their description are:

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

## CalendarProvider, LotusNotesCalendarProvider, and MExchangeCalendarProvider

The CalendarProvider works with the Sun Java System Calendar Server so that you can view tasks and events and launch Calendar Express without having to sign in. The LotusNotesCalendarProvider works with the Lotus Notes Server so that you can view tasks and events and launch the web application without having to sign in. The MExchangeCalendarProvider works with the Microsoft Exchange Server so that you can view tasks and events and launch Exchanges web application.

The list of properties specific to CalendarProvider, LotusNotesCalendarProvider, and MExchangeCalendarProvider and their description are:

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.

## IMProvider

The IMProvider includes:

- Information needed to help the user decide whether to launch the IM (Instant Messenger) 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.

The list of properties specific to IMProvider and their description are:

mux	Specifies the name of instant messaging multiplexor to use (used by IM client.)
muxport	Specifies the port on which the instant messaging multiplexor listens.
codebase	Specifies where to find the instant messaging client.
netletRule	Specifies where to find the instant messaging client when using the netlet. By default, the value is IM.
clientRunMode	Specifies how the Instant Messaging server client must be run. The client can be run as either a plugin or jnlp. By default, the value is plugin.
authMethod	Specifies the authentication method. Clients can authenticate either via idsvr (for Sun Java System Access Manager) or ldap. By default, the value is idsvr.

<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.

## 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 specify the following URL in a browser, you see the authlessanonymous user's Desktop, which contains 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.

The properties, which are specific to the LoginProvider and their description are:

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&amp; goto=http://www.siroe.con:80/portal/dt</pre>

## NotesProvider

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

The list of properties specific to NotesProvider and their description are:

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.

Example of 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
```

## 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.

The list of properties specific to SimpleWebServiceProvider and their description are:

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

## 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.

The list of properties specific to SimpleWebServiceConfigurableProvider and their description are:

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

# Display Profile Common Properties for Leaf Providers

---

Providers contain required properties and general properties. This chapter contains the following sections, which describe the general properties of providers.

- [“Introduction” on page 167](#)
- [“Common Properties” on page 167](#)
- [“Other Leaf Provider Display Profile Properties” on page 169](#)

## Introduction

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 `psadmin` command (or the administration console) to store it in LDAP.

The 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.

---

## Common Properties

The list of common properties for a `<Provider>` definition and their description is provided below. 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.

`title` Specifies the title that appears in the channel title bar in the Desktop.

description	<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>
refreshTime	<p>The refreshTime property controls how often a channel's content is reloaded.</p>
editType	<p>Specifies the edit type, either edit_complete or edit_subset. If edit_complete, the provider's getEdit() method is responsible for generating the complete Edit page content. If edit_subset, a generic edit provider container is used to put a frame around the Edit page. The provider's getEdit() page is then called, and displays the content of the Edit page.</p>
isEditable	<p>Determines if the provider has an edit view. If 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.</p> <p>If isEditable is false, no edit view is provided and no Edit button is generated in the 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.</p> <p>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.</p>
width	<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 thin, thick, full_top, or full_bottom. In general, these values only make sense for an HTML-based Desktop.</p>
helpURL	<p>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:</p> <pre>http://hostname:port/portal/docs/en/desktop/bkmark.html</pre> <p>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.</p>



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 Java System Portal Server. This is not required by all 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 if the provider class extends ProviderAdapter or the ProfileProviderAdapter class. Note that the channel can set its own properties that override these values.

## Other Leaf Provider Display Profile Properties

### 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.get/setClientProperty() 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.get/setClientProperty() 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 the following example.

### EXAMPLE 24-1 `<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>
</Properties>
```

## Display Profile Channel Properties

---

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.

### Display Profile Channel Properties

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 (see “Available and Selected List” on page 49) and can contain leaf channel definitions.

Both container and leaf channel properties can be configured from the administration console. See the provider-specific display profile properties for more information on the channel properties.



**PART IV**

**Desktop Java Server Pages**

- [Chapter 26](#)
- [Chapter 27](#)
- [Chapter 28](#)
- [Chapter 29](#)
- [Chapter 30](#)
- [Chapter 31](#)
- [Chapter 32](#)



# Overview of Desktop JavaServer Pages

---

This chapter contains the following sections:

- “Introduction” on page 175
- “Installation Location” on page 175
- “The Desktop and JavaServer Pages” on page 176
- “File Lookup Scenario” on page 176

## Introduction

To generate the rendered Desktop user interface (what the industry refers to as the “presentation”), the Sun Java System Portal Server 7.1 software makes use of either JSPs 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.

## Installation Location

The default set of JSPs are installed in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default` directory. The developer sample JSPs are installed in `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample` and `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_anonymous_sample` directories. Files in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_anonymous_sample` directory are specific to the developer sample Anonymous Desktop.

## 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.

## File Lookup Scenario

The Portal Server software uses the lookup scenario outlined in this section to find the JSPs it needs. Use this order to decide the final location of your own JSPs.

### EXAMPLE 26-1 JSP Lookup Scenario

```
desktoptype_  
locale/  
channelname/  
clientPath
```

```
desktoptype_  
locale/  
provider/  
clientPath
```

```
desktoptype_  
locale/  
channelname
```

```
desktoptype_  
locale/  
provider
```

```
desktoptype_  
locale/  
clientPath
```



## EXAMPLE 26-1 JSP Lookup Scenario (Continued)

```

    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/

```

**EXAMPLE 26-1** JSP Lookup Scenario (Continued)

```
channelname
default/
provider
default/
clientPath
default
templateroot
```

If there is no `clientPath` specified, then the directory search order is as follows:

**EXAMPLE 26-2** JSP Lookup Scenario (no `clientPath`)

```
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
```

---

**EXAMPLE 26-2** JSP Lookup Scenario (no `clientPath`) (Continued)*templatereoot*

The lookup scenario relies on the following parameters:

<code>desktoptype</code>	For example <code>default</code> (set in the administration console). Note that <code>desktoptype</code> is now a comma separated string list and so the look up will be based on the <code>desktoptype(s)</code> that are defined in the <code>desktoptype</code> attribute.
<code>locale</code>	Preferred <code>locale</code> is the user's locale. For example, <code>en_US</code> (set by users through the administration console in the "User" setting)
<code>clientPath</code>	This is an optional file-path containing client-specific templates; for example, <code>html</code> (set through the administration console Client Detection service)
<code>channelname</code>	This is the name of the channel; for example, <code>newSingleContainer</code> (set in the display profile)
<code>provider</code>	This is the provider name; for example, <code>JSPSingleContainerProvider</code> (set in the display profile)
<code>templatereoot</code>	This is defined in the <code>desktopconfig.properties</code> file. The root of the search directory (default value of <code>/var/opt/SUNWportal/portals/&lt;portal_id&gt;/config/</code> ) can be changed by modifying the <code>templateBaseDir</code> property in the <code>desktopconfig.properties</code> file.



## JSPs in the default Directory

---

This chapter lists all the JSPs available in the JSP sub directories under the default directory installed in the Sun Java System Portal Server 7.1. The following sections are the names of the sub directories under the default directory. These sections list the JSPs available in each subdirectory and their description.

- “AddressBookJSPProvider” on page 182
- “CalendarJSPProvider” on page 183
- “CommunityDiscussionProvider” on page 183
- “CommunitySearchProvider” on page 184
- “CommunitySubscriptionsProvider” on page 185
- “DefaultChannel” on page 186
- “DiscussionLite” on page 186
- “DiscussionProvider” on page 186
- “IFrameProvider” on page 188
- “IMProvider” on page 188
- “JSPContentContainer” on page 189
- “JSPDynamicSingleContainer” on page 189
- “JSPEditContainer” on page 189
- “JSPFrameCustomTableContainerProvider” on page 190
- “JSPLayoutContainer” on page 191
- “JSPMenuContainerProvider” on page 191
- “JSPNativeContainer” on page 192
- “JSPProvider” on page 192
- “JSPRenderingContainerProvider” on page 192
- “JSPRenderingContentChannel” on page 193
- “JSPRenderingEditContainer” on page 193
- “JSPRenderingErrorChannel” on page 193
- “JSPRenderingLayoutChannel” on page 194
- “JSPRenderingProvider” on page 194
- “JSPSingleContainerProvider” on page 194
- “JSPSingleRenderingContainerProvider” on page 195

- “JSPTabContainerProvider” on page 195
- “JSPTabCustomTableContainerProvider” on page 196
- “JSPTableContainerProvider” on page 197
- “JSPs Within the default Directory” on page 198
- “LotusNotesAddressBookJSPProvider” on page 199
- “LotusNotesCalendarJSPProvider” on page 200
- “LotusNotesMailJSPProvider” on page 200
- “MAAddressBookJSPProvider” on page 200
- “MAMailJSPProvider” on page 201
- “MSEExchangeAddressBookJSPProvider” on page 202
- “MSEExchangeCalendarJSPProvider” on page 203
- “MSEExchangeMailJSPProvider” on page 203
- “MailJSPProvider” on page 204
- “RenderingWrappingProvider” on page 204
- “SampleSimpleWebService” on page 204
- “SampleSimpleWebServiceConfigurable” on page 205
- “SearchProvider” on page 205
- “SimpleWebServiceConfigurableProvider” on page 207
- “SimpleWebServiceProvider” on page 207
- “SubscriptionsProvider” on page 208
- “TabJSPEditContainer” on page 208
- “UWCAddressBookJSPProvider” on page 209
- “UWCCalendarJSPProvider” on page 209
- “UWCMailJSPProvider” on page 210
- “WirelessJSPClientConfigContainer” on page 210
- “WirelessJSPContentChannel” on page 210
- “WirelessJSPDesktopProvider” on page 211
- “WirelessJSPLayoutChannel” on page 212
- “vxml” on page 213

## AddressBookJSPProvider

The AddressBookJSPProvider JSPs are located in html subdirectory under the `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/html/AddressBookJSPProvider` directory.

The JSPs available in the AddressBookJSPProvider subdirectory and their functionality are:

<code>content.jsp</code>	Displays address book entries.
<code>contentJSFunctions.jsp</code>	JavaScript functions used by channel (included by <code>content.jsp</code> ).
<code>contentNavigation.jsp</code>	Displays navigation links in channel (included by <code>content.jsp</code> ).

<code>contentNavigationMenu.jsp</code>	Displays navigation pull-down menu in channel (included in <code>content.jsp</code> ).
<code>contentStateAttrs.jsp</code>	Controls state of channel (included by <code>content.jsp</code> ).
<code>edit.jsp</code>	Allows editing of channel preferences.
<code>error.jsp</code>	Displays error message in channel.
<code>process.jsp</code>	Persists changes to channel preferences.

## CalendarJSPProvider

The CalendarJSPProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/CalendarJSPProvider` directory.

The JSPs available in the CalendarJSPProvider subdirectory and their functionality are:

<code>calheader.jsp</code>	Displays header information for the calendar.
<code>content.jsp</code>	Retrieves calendar session.
<code>edit.jsp</code>	Allows user to edit preferences
<code>error.jsp</code>	Error message used when calendar connection fails.
<code>fetch.jsp</code>	Gets user preferences and retrieves calendar information based on those preferences.
<code>process.jsp</code>	Modifies user preferences from the edit page.
<code>summary.jsp</code>	Summary number events and tasks. Allows creation of new tasks if server supports the operation.
<code>table.jsp</code>	Shows calendar information for a month.
<code>today.jsp</code>	Shows calendar information for the selected day in day and day or week view.

## CommunityDiscussionProvider

The CommunityDiscussionProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/CommunityDiscussionProvider` directory.

The JSPs available in the CommunityDiscussionProvider subdirectory and their functionality are:

<code>declare.jsp</code>	Displays the declare section.
<code>discussionContent.jsp</code>	Displays the discussion content.
<code>discussionDoEdit.jsp</code>	Displays the discussion do edit.
<code>discussionEdit.jsp</code>	Displays the discussion edit.
<code>error.jsp</code>	Error message used when community discussion connection fails.
<code>feedback.jsp</code>	Displays the feedback section.
<code>feedbackDisplay.jsp</code>	Displays the feedback display.
<code>feedbackForm.jsp</code>	Displays the feedback form.
<code>feedbackProcess.jsp</code>	Displays the feedback process.
<code>fullDiscussion.jsp</code>	Displays the full discussion section.
<code>fullDiscussionDisplay.jsp</code>	Displays the full discussion display.
<code>pageFooter.jsp</code>	Displays the page footer.
<code>portal.jsp</code>	Displays the portal.
<code>query.jsp</code>	Displays the query.
<code>rating.jsp</code>	Displays the rating.
<code>ratingProcess.jsp</code>	Displays the rating process.
<code>searchUI.jsp</code>	Displays the search UI.
<code>viewDiscussion.jsp</code>	Displays the view discussion.
<code>viewDiscussionBar.jsp</code>	Displays the view discussion bar.
<code>viewDiscussionDisplay.jsp</code>	Display the view discussion display.
<code>viewDiscussionHeader.jsp</code>	Displays the view discussion header.
<code>viewDiscussionNavigation.jsp</code>	Displays the view discussion navigation.

## CommunitySearchProvider

The CommunitySearchProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/CommunitySearchProvider` directory.

The JSPs available in the CommunitySearchProvider subdirectory and their functionality are:



<code>advQuery.jsp</code>	Displays the advance query.
<code>advancedSearch.jsp</code>	Displays the advanced search.
<code>basicSearch.jsp</code>	Displays the basic search.
<code>browseHeader.jsp</code>	Displays the browse header.
<code>browseOnly.jsp</code>	Displays the browse only section.
<code>browseResults.jsp</code>	Displays browse results.
<code>browseSearch.jsp</code>	Displays browse search.
<code>browseSearchResults.jsp</code>	Displays browse search results.
<code>catMenu.jsp</code>	Displays cat menu.
<code>dbMenu.jsp</code>	Displays database menu.
<code>descMenu.jsp</code>	Displays description menu.
<code>error.jsp</code>	Error message used when community search connection fails.
<code>pageFooter.jsp</code>	Displays page footer.
<code>processRating.jsp</code>	Displays process rating.
<code>psSearch.jsp</code>	Displays ps search.
<code>rating.jsp</code>	Displays rating section.
<code>results.jsp</code>	Displays results.
<code>score.jsp</code>	Displays score section,
<code>searchContent.jsp</code>	Displays search content.
<code>searchDoEdit.jsp</code>	Displays search do edit.
<code>searchEdit.jsp</code>	Displays search edit.
<code>searchMenu.jsp</code>	Displays search menu.
<code>searchOnly.jsp</code>	Displays search only.

## CommunitySubscriptionsProvider

The CommunitySubscriptionsProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/CommunitySubscriptionsProvider` directory.

The JSPs available in the CommunitySubscriptionsProvider subdirectory and their functionality are:

<code>profilerContent.jsp</code>	Displays profiler content.
<code>subcontent.jsp</code>	Displays subscription content.
<code>subsdoedit.jsp</code>	Displays subscription do edit.
<code>subsededit.jsp</code>	Displays subscription edit.

## DefaultChannel

The DefaultChannel JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/DefaultChannel` directory.

The JSPs available in the DefaultChannel subdirectory and their functionality are:

<code>content.jsp</code>	Retrieves default channel session.
<code>menubar.jsp</code>	The menubar for the Default Channel.

## DiscussionLite

The DiscussionLite channel JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/DiscussionLite` directory.

The JSPs available in the DiscussionLite subdirectory and their functionality are:

<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.

## DiscussionProvider

The DiscussionProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/DiscussionProvider` directory.

The JSPs available in the DiscussionProvider subdirectory and their functionality are:

<code>declare.jsp</code>	Declares all portal properties.
<code>discussionContent.jsp</code>	Content page. This JSP routes the request to full if display mode is set to full (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 search result and submits the search result 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=vl</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".

## IFrameProvider

The IFrameProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/IFrameProvider` directory.

The JSPs available in the IFrameProvider subdirectory and their functionality are:

`iframe.jsp` Displays the iframe.

## IMProvider

The IMProvider JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/IMProvider` directory.

The JSPs available in the IMProvider subdirectory and their functionality are:

<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

The JSPContentContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/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 JSPs available in the JSPContentContainer subdirectory and their functionality are:

<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 `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/JSPDynamicSingleContainer` directory. This container is used by the search form in the header on the Desktop front page.

The JSPs available in the JSPDynamicSingleContainer subdirectory and their functionality are:

<code>dynamicSingle.jsp</code>	Used by the DynamicSingleContainer to display the channel specified in the request parameter.
--------------------------------	---

## JSPEditContainer

JSPEditContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/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 JSPs available in the JSPEditContainer subdirectory and their functionality are:

<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

JSPFrameCustomTableContainerProvider JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/JSPFrameCustomTableContainerProvider` directory.

JSPFrameCustomTableContainerProvider JSPs are used when the user creates a new page from Scratch in the sections page.

The JSPs available in the JSPFrameCustomTableContainerProvider subdirectory and their functionality are:

<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

JSPLayoutContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/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 JSPs available in the JSPLayoutContainer subdirectory and their functionality are:

<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.

## JSPMenuContainerProvider

This container is responsible for displaying menu or tab items. The JSPMenuContainerProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/JSPMenuContainerProvider` directory.

The JSPs available in the JSPMenuContainerProvider subdirectory and their functionality are:

<code>channels.jsp</code>	Displays the channels.
<code>header.jsp</code>	Displays the header.
<code>menu.jsp</code>	Displays the vertical menu.
<code>menubar.jsp</code>	The menubar for the Default Channel.
<code>menuedit.jsp</code>	This is the entry point into this container in edit mode.
<code>menulist.jsp</code>	The menubar for the Default Channel.
<code>menumain.jsp</code>	This is the entry point into this container in content mode.
<code>menuoption.jsp</code>	The menubar for the Default Channel.
<code>selectedchannels.jsp</code>	The menubar for the Default Channel.

## JSPNativeContainer

The JSPNativeContainer JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/JSPNativeContainer/aml/chtml` directory.

The JSPs available in the chtml subdirectory under JSPNativeContainer and their functionality are:

<code>channelMenuSection.jsp</code>	Displays the channel menu section.
<code>channelsContent.jsp</code>	The menubar for the Default Channel.
<code>header.jsp</code>	Displays the header.
<code>nonInlineChannelMenuSection.jsp</code>	The menubar for the Default Channel.
<code>wireless.jsp</code>	Displays the wireless section.

## JSPProvider

JSP Provider is used to create a channel using jsp file. JSPProvider JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/JSPProvider` directory. This directory contains default set of JSPs that are used by the JSP channels.

The JSPs available in the JSPProvider subdirectory and their functionality are:

<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.

## JSPRenderingContainerProvider

The JSPRenderingContainerProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/JSPRenderingContainerProvider/aml` directory.

The JSPs available in the aml subdirectory under JSPRenderingContainerProvider and their functionality are:

<code>channelMenuSection.jsp</code>	Displays the channel menu section.
<code>channelsCards.jsp</code>	The menubar for the Default Channel.



<code>desktop.jsp</code>	Displays the desktop.
<code>dtSetup.jsp</code>	The menubar for the Default Channel.
<code>minimizedChannelMenuSection.jsp</code>	Displays the minimized channel menu section.
<code>optionsCard.jsp</code>	Displays the options card.

## JSPRenderingContentChannel

The JSPRenderingContentChannel JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/JSPRenderingContentChannel/aml` directory.

The JSPs available in the aml subdirectory under JSPRenderingContentChannel and their functionality are:

<code>contentSetup.jsp</code>	Displays the content setup.
<code>contentdoedit.jsp</code>	The menubar for the Default Channel.
<code>contentedit.jsp</code>	Displays the content edit.

## JSPRenderingEditContainer

The JSPRenderingEditContainer JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/JSPRenderingEditContainer/aml` directory.

The JSPs available in the aml subdirectory under JSPRenderingEditContainer and their functionality are:

<code>doedit.jsp</code>	Displays the do edit.
<code>edit.jsp</code>	Allows user to edit preferences.

## JSPRenderingErrorChannel

The JSPRenderingErrorChannel JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/JSPRenderingErrorChannel/aml` directory.

The JSPs available in the aml subdirectory under JSPRenderingErrorChannel and their functionality are:

`errorPage.jsp` Displays the error page.

## JSPRenderingLayoutChannel

The JSPRenderingLayoutChannel JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/JSPRenderingLayoutChannel/aml` directory.

The JSPs available in the `aml` subdirectory under JSPRenderingLayoutChannel and their functionality are:

`layoutSetup.jsp` Displays the layout setup.

`layoutdoedit.jsp` Displays the layout do edit.

`layoutedit.jsp` Displays the layout edit.

## JSPRenderingProvider

The JSPRenderingProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/JSPRenderingProvider/aml` directory.

The JSPs available in the `aml` subdirectory under JSPRenderingProvider and their functionality are:

`samplecontent.jsp` Displays the sample content.

## JSPSingleContainerProvider

JSPSingleContainerProvider JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/JSPSingleContainerProvider` directory.

The JSPs available in the JSPSingleContainerProvider subdirectory and their functionality are:

`header.jsp` Displays the product banner that is used by the Single Container page.

`menubar.jsp` Displays the menubar that has the Home, Theme, Help, and Logout links.

`single.jsp` Displays the content for JSPSingleContainerProvider.

## JSPSingleRenderingContainerProvider

The JSPSingleRenderingContainerProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/JSPSingleRenderingContainerProvider` directory.

The JSPs available in the `aml` subdirectory under JSPSingleRenderingContainerProvider and their functionality are:

`header.jsp`     Displays the header.  
`single.jsp`     Displays the single section.

## JSPTabContainerProvider

JSPTabContainerProvider JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/JSPTabContainerProvider` directory. These JSPs are used as the default set of JSPs for a new channel based on JSPTabContainerProvider.

The JSPs available in the JSPTabContainerProvider subdirectory and their functionality are:

`availableTabs.jsp`     Displays the tabs and the URLs associated with each tab for activating them on the front page.

`header.jsp`     Displays the header bar for the Tab Container page. (Dynamically included.)

`makeNewTab.jsp`     Provides the content for the Make New Tab page of the tab container.

`makeTopic.jsp`     Provides the content for each of the tab topics in the Make New Tab page.

`menubar.jsp`     Displays the menubar that has the Home, Tabs, Theme, Help, and Logout links. (Dynamically included.)

`remove.jsp`     Displays the remove, rename, and start links for each of the selected pages of the JSP tab container in the Current Tab Settings page.

`removeRenameTab.jsp`     Displays the remove and rename part of the Edit page for the tab container.

`selectedTab.jsp`     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

JSPTabCustomTableContainerProvider JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/JSPTabCustomTableContainerProvider` directory. JSPTabCustomTableContainerProvider JSPs are used when the user creates a new tab from scratch in the tabs page.

The JSPs available in the JSPTabCustomTableContainerProvider subdirectory and their functionality are:

<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

Files under this directory are used to layout the portal page. This container can display channels at FULL TOP, LEFT, CENTER, RIGHT, and FULL BOTTOM locations.

JSPTableContainerProvider JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/JSPTableContainerProvider` directory. JSPTableContainerProvider JSPs are the default JSPs that are used by the JSPTableContainerProvider channels.

The JSPs available in the JSPTableContainerProvider subdirectory and their functionality are:

<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>	This JSP is included by <code>leafWrapper.jsp</code> . This JSP is responsible to display the maximize, minimize, detach, and delete buttons. 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>	This is the entry point into this container for content mode. 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>	Displays the LEFT, CENTER, and RIGHT channels 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>	Displays the FULL TOP and FULL 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> .)

## JSPs Within the default Directory

The JSPs listed below are located within the `/var/opt/SUNWportal/portals/<portal_id>/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 JSPs available within the `/default` directory and their functionality are:

<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>PortletBanner.jsp</code>	This JSP is used to draw the banner on the edit page of a JSR (Java Specification Request) 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>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>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.
<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.

## LotusNotesAddressBookJSPProvider

The LotusNotesAddressBookJSPProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/LotusNotesAddressBookJSPProvider` directory.

The JSPs available in the `html` subdirectory under LotusNotesAddressBookJSPProvider and their functionality are:

<code>content.jsp</code>	Displays address book entries.
<code>contentJSFunctions.jsp</code>	JavaScript functions used by channel (included by <code>content.jsp</code> ).
<code>contentNavigation.jsp</code>	Displays navigation links in channel (included by <code>content.jsp</code> ).
<code>contentNavigationMenu.jsp</code>	Displays navigation pull-down menu in channel (included in <code>content.jsp</code> ).
<code>contentStateAttrs.jsp</code>	Controls state of channel (included by <code>content.jsp</code> ).
<code>edit.jsp</code>	Allows editing of channel preferences.
<code>error.jsp</code>	Displays error message in channel.

`process.jsp` Persists changes to channel preferences.

## LotusNotesCalendarJSPProvider

The LotusNotesCalendarJSPProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/LotusNotesCalendarJSPProvider` directory.

The JSPs available in the LotusNotesCalendarJSPProvider subdirectory are similar to the JSPs available in the CalendarJSPProvider. For more information on these JSPs, see [“CalendarJSPProvider” on page 183](#).

## LotusNotesMailJSPProvider

The LotusNotesMailJSPProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/LotusNotesMailJSPProvider/html` directory.

The JSPs available in the html subdirectory under LotusNotesCalendarJSPProvider and their functionality are:

<code>content.jsp</code>	Retrieves lotus notes mail session.
<code>edit.jsp</code>	Allows user to edit preferences of the lotus notes mail.
<code>error.jsp</code>	Error message used when lotus notes mail connection fails.
<code>process.jsp</code>	Modifies user preferences from the edit page

## MAAddressBookJSPProvider

The MAAddressBookJSPProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/MAAddressBookJSPProvider/html` directory.

The JSPs available in the html subdirectory under MAAddressBookJSPProvider and their functionality are:

<code>edit.jsp</code>	Allows editing of mobile address book application preferences. Used for Mobile Access application.
<code>process.jsp</code>	Persists changes to application preferences. Used for Mobile Access application.



## MAMailJSPProvider

The MAMailJSPProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/MAMailJSPProvider/html` directory.

The JSPs available in the `html` subdirectory under MAMailJSPProvider and their functionality are:

<code>edit-device.jsp</code>	Used for mobile access. Allows editing of device properties.
<code>edit-devices.jsp</code>	Used for mobile access. Displays a list of devices that can be edited.
<code>edit-main-other.jsp</code>	Used for mobile access. Allows editing properties for the mobile mail application.
<code>edit-main-pop.jsp</code>	Used for mobile access. Allows editing of properties of POP collection feature of the mobile mail application.
<code>edit-main-views.jsp</code>	Used for mobile access. Allows editing of views feature of the mobile mail application.
<code>edit-pop.jsp</code>	Used for mobile access. Allows editing of properties of POP collection feature of the mobile mail application.
<code>edit-rule.jsp</code>	Used for mobile access. Allows editing of rules feature of the mobile mail application.
<code>edit-rules.jsp</code>	Used for mobile access. Allows editing of rules feature of the mobile mail application.
<code>edit-view.jsp</code>	Used for mobile access. Allows editing of views feature of the mobile mail application.
<code>edit-views.jsp</code>	Used for mobile access. Displays list of views configured for the mobile mail application.
<code>edit.jsp</code>	Used for mobile access. Allows editing of mobile mail application preferences.
<code>error.jsp</code>	Used for mobile access. Displays error message when channel connection fails.
<code>process-device.jsp</code>	Used for mobile access. Persists changes to the user preferences.
<code>process-main.jsp</code>	Used for mobile access. Persists changes to the user preferences.

<code>process-pop-add.jsp</code>	Used for mobile access. Persists changes to the user preferences.
<code>process-pop-common.jsp</code>	Used for mobile access. Persists changes to the user preferences.
<code>process-pop-delete.jsp</code>	Used for mobile access. Persists changes to the user preferences.
<code>process-pop-edit.jsp</code>	Used for mobile access. Persists changes to the user preferences.
<code>process-rule-add.jsp</code>	Used for mobile access. Persists changes to the user preferences.
<code>process-rule-common.jsp</code>	Used for mobile access. Persists changes to the user preferences.
<code>process-rule-delete.jsp</code>	Used for mobile access. Persists changes to the user preferences.
<code>process-rule-edit.jsp</code>	Used for mobile access. Persists changes to the user preferences.
<code>process-view-add.jsp</code>	Used for mobile access. Persists changes to the user preferences.
<code>process-view-common.jsp</code>	Used for mobile access. Persists changes to the user preferences.
<code>process-view-delete.jsp</code>	Used for mobile access. Persists changes to the user preferences.
<code>process-view-edit.jsp</code>	Used for mobile access. Persists changes to the user preferences.
<code>process.jsp</code>	Used for mobile access. Persists changes to the user preferences.

## MSExchangeAddressBookJSPProvider

The MSExchangeAddressBookJSPProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/MSExchangeAddressBookJSPProvider/html` directory.

The JSPs available in the html subdirectory under MSExchangeAddressBookJSPProvider and their functionality are:

<code>content.jsp</code>	Displays MSExchange address book entries.
--------------------------	---

<code>contentJSFunctions.jsp</code>	JavaScript functions used by channel (included by <code>content.jsp</code> ).
<code>contentNavigation.jsp</code>	Displays navigation links in channel (included by <code>content.jsp</code> ).
<code>contentNavigationMenu.jsp</code>	Displays navigation pull-down menu in channel (included in <code>content.jsp</code> ).
<code>contentStateAttrs.jsp</code>	Controls state of channel (included by <code>content.jsp</code> ).
<code>edit.jsp</code>	Allows editing of channel preferences.
<code>error.jsp</code>	Displays error message in channel.
<code>process.jsp</code>	Persists changes to channel preferences.

## MSExchangeCalendarJSPProvider

The MSExchangeCalendarJSPProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/MSExchangeCalendarJSPProvider` directory.

The JSPs available in the MSExchangeAddressBookJSPProvider subdirectory are similar to the JSPs available in the CalendarJSPProvider. For more information on these JSPs, see [“CalendarJSPProvider” on page 183](#).

## MSExchangeMailJSPProvider

The MSExchangeMailJSPProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/MSExchangeMailJSPProvider/html` directory.

The JSPs available in the html subdirectory under MSExchangeMailJSPProvider and their functionality are:

<code>content.jsp</code>	Displays a list of email headers.
<code>edit.jsp</code>	Allows editing of channel preferences.
<code>error.jsp</code>	Displays error message when channel connection fails.
<code>process.jsp</code>	Persists changes to the user preferences.

## MailJSPPProvider

The MailJSPPProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/MailJSPPProvider/html` directory.

The JSPs available in the `html` subdirectory under MailJSPPProvider and their functionality are:

- `content.jsp` Displays a list of email headers.
- `edit.jsp` Allows editing of channel preferences.
- `error.jsp` Displays error message when channel connection fails.
- `process.jsp` Persists changes to the user preferences.

## RenderingWrappingProvider

The RenderingWrappingProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/RenderingWrappingProvider/aml` directory.

The JSPs available in the `aml` subdirectory under RenderingWrappingProvider and their functionality are:

- `contentWrapper.jsp` Displays the content wrapper.

## SampleSimpleWebService

SampleSimpleWebService JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/SampleSimpleWebService` directory.

The JSPs available in the SampleSimpleWebService subdirectory and their functionality are:

- `CurrencyExchangeService.wsdl` 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

SampleSimpleWebServiceConfigurable JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/SampleSimpleWebServiceConfigurable` directory.

The JSPs available in the SampleSimpleWebServiceConfigurable subdirectory and their functionality are:

<code>CurrencyExchangeService.wsdl</code>	Displays the WSDL file for the current exchange web service.
<code>webserviceContent.jsp</code>	Displays the Content view of the simple web service configurable channels.
<code>webserviceWsdLedit.jsp</code>	Displays the Edit view of the simple web service configurable channels.

## SearchProvider

The SearchProvider JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/SearchProvider` directory.

The JSPs available in the SearchProvider subdirectory and their functionality are:

<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 <code>RDMTType</code> to <code>rd-request</code> and query language to <code>search</code> , 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>catMenu.jsp</code>	Displays category list.
<code>dbMenu.jsp</code>	Displays database list.
<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>	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.

This JSP includes the `advancedSearch.jsp` or `basicSearch.jsp` based on user selection. The `browseSearch.jsp` and `searchOnly.jsp` files are included only if the user has specified a query; otherwise the category tree (no search) is displayed from `browseOnly.jsp`. The `pageFooter.jsp` file is included to display the pagination bar in the Search channel.

<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.

## SimpleWebServiceConfigurableProvider

`SimpleWebServiceConfigurableProvider` JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/SimpleWebServiceConfigurableProvider` directory. These JSPs are the default JSPs that are used by the `SimpleWebServiceConfigurableProvider` channels.

The JSPs available in the `SimpleWebServiceConfigurableProvider` subdirectory and their functionality are:

<code>webserviceContent.jsp</code>	Displays the Content view of the simple web service configurable channels.
<code>webserviceWsdEdit.jsp</code>	Displays the Edit view of the simple web service configurable channels.

## SimpleWebServiceProvider

`SimpleWebServiceProvider` JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/SimpleWebServiceProvider` directory. These JSPs are default JSPs that are used by the `SimpleWebServiceProvider` channels.

The JSPs available in the `SimpleWebServiceProvider` subdirectory and their functionality are:

<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.

## SubscriptionsProvider

The SubscriptionsProvider JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/SubscriptionsProvider` directory.

The JSPs available in the SubscriptionsProvider subdirectory and their functionality are:

<code>profilerContent.jsp</code>	Displays the profiler content.
<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>subsdoedit.jsp</code>	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.  This JSP segments the subscriptions in to three types: category subscriptions, discussions subscriptions, saved search subscriptions.
<code>subsedit.jsp</code>	This JSP is triggered to handle the subscriptions changes made by the end user in the page presented by <code>subsedit.jsp</code> . The role of this JSP, is to update the Sun Java System Access Manager subscriptions service attributes holding the subscription information.

## TabJSPEditContainer

TabJSPEditContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/TabJSPEditContainer` directory.

The JSPs available in the TabJSPEditContainer subdirectory and their functionality are:

<code>contentLayoutBar.jsp</code>	Displays the Edit page for channels in JSPTabContainer.
-----------------------------------	---



<code>doedit.jsp</code>	Is the process page for TabJSPEditContainer.
<code>tabedit.jsp</code>	Displays the Content and Layout links and the current selected tab on the Edit page.

## UWCAddressBookJSPProvider

The UWCAddressBookJSPProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/UWCAddressBookJSPProvider/html` directory.

The JSPs available in the `html` subdirectory under UWCAddressBookJSPProvider and their functionality are:

<code>content.jsp</code>	Displays UWC address book entries.
<code>contentJSFunctions.jsp</code>	JavaScript functions used by channel (included by <code>content.jsp</code> ).
<code>contentNavigation.jsp</code>	Displays navigation links in channel (included by <code>content.jsp</code> ).
<code>contentNavigationMenu.jsp</code>	Displays navigation pull-down menu in channel (included in <code>content.jsp</code> ).
<code>contentStateAttrs.jsp</code>	Controls state of channel (included by <code>content.jsp</code> ).
<code>edit.jsp</code>	Allows editing of channel preferences.
<code>error.jsp</code>	Displays error message in channel.
<code>process.jsp</code>	Persists changes to channel preferences.

## UWCCalendarJSPProvider

The UWCCalendarJSPProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/UWCCalendarJSPProvider` directory.

The JSPs available in the UWCCalendarJSPProvider subdirectory are similar to the JSPs available in the CalendarJSPProvider. For more information on these JSPs, see [“CalendarJSPProvider” on page 183](#).

## UWCMailJSPProvider

The UWCMailJSPProvider JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/UWCMailJSPProvider/html` directory.

The JSPs available in the html subdirectory under UWCMailJSPProvider and their functionality are:

- `content.jsp`     Displays a list of email headers.
- `edit.jsp`        Allows editing of channel preferences.
- `error.jsp`        Displays error message when channel connection fails.
- `process.jsp`     Persists changes to the user preferences.

## WirelessJSPClientConfigContainer

The WirelessJSPClientConfigContainer JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/WirelessJSPClientConfigContainer/html` directory.

The JSPs available in the html subdirectory under WirelessJSPClientConfigContainer and their functionality are:

- `clientconfig.jsp`             Displays the client configuration.
- `clientconfigSetup.jsp`        Displays the client configuration setup.
- `clientconfigprocess.jsp`     Displays the client configuration process.

## WirelessJSPContentChannel

The WirelessJSPContentChannel JSPs are located in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/WirelessJSPContentChannel` directory. Under this directory, you can find the `contentSetup.jsp`.

- `contentSetup.jsp`     Displays the content setup.

The JSPs available in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/WirelessJSPContentChannel/aml/chtml` directory and their functionality are:

- `contentdoedit.jsp`     Displays the content do edit.

`contentedit.jsp` Displays the content edit.

The JSPs available in the

`/var/opt/SUNWportal/portals/<portal_id>/desktop/default/WirelessJSPContentChannel/html` directory and their functionality are:

`contentSetup.jsp` Displays the content setup.

`contentdoedit.jsp` Displays the content do edit.

`contentedit.jsp` Displays the content edit.

The JSPs available in the

`/var/opt/SUNWportal/portals/<portal_id>/desktop/default/WirelessJSPContentChannel/wml` directory and their functionality are:

`adjustChannelList.jsp` Displays the adjust channel list.

`contentdoedit.jsp` Displays the content do edit.

`contentedit.jsp` Displays the content edit.

The JSPs available in the

`/var/opt/SUNWportal/portals/<portal_id>/desktop/default/WirelessJSPContentChannel/vxm` directory and their functionality are:

`adjustChannelList.jsp` Displays the adjust channel list.

`contentdoedit.jsp` Displays the content do edit.

`contentedit.jsp` Displays the content edit.

`voiceSetup.jsp` Displays the voice setup.

## WirelessJSPDesktopProvider

The WirelessJSPDesktopProvider JSPs are located in

`/var/opt/SUNWportal/portals/<portal_id>/desktop/default/WirelessJSPDesktopProvider` directory. Under this directory, you can find the `dtSetup.jsp`.

`dtSetup.jsp` Displays the dtsetup.

The JSPs available in the

`/var/opt/SUNWportal/portals/<portal_id>/desktop/default/WirelessJSPDesktopProvider/aml/html` directory and their functionality are:

`channelMenuSection.jsp` Displays the channel menu section.

`channelsContent.jsp` Displays the channels content.

`nonInlineChannelMenuSection.jsp` Displays the non inline channel menu section.

`wireless.jsp` Displays the wireless section.

The JSPs available in the

`/var/opt/SUNWportal/portals/<portal_id>/desktop/default/WirelessJSPDesktopProvider/vxml` directory and their functionality are:

`desktop.jsp` Displays the desktop.

`voiceSetup.jsp` Displays the voice setup.

The JSPs available in the

`/var/opt/SUNWportal/portals/<portal_id>/desktop/default/WirelessJSPContentChannel/wml` directory and their functionality are:

`channelMenuSection.jsp` Displays the channel menu section.

`channelsCards.jsp` Displays the channel cards.

`minimizedChannelMenuSection.jsp` Displays the minimized channel menu section.

`optionsCard.jsp` Displays the options card.

`wireless.jsp` Displays the wireless section.

## WirelessJSPLayoutChannel

The WirelessJSPLayoutChannel JSPs are located in

`/var/opt/SUNWportal/portals/<portal_id>/desktop/default/WirelessJSPLayoutChannel` directory. Under this directory, you can find the following JSPs.

`layoutSetup.jsp` Displays the layout setup.

`layoutdoedit.jsp` Displays the layout do edit.

The JSP available in the

`/var/opt/SUNWportal/portals/<portal_id>/desktop/default/WirelessJSPLayoutChannel/html/chtml` directory and its functionality is:

`layoutedit.jsp` Displays the layout edit.

The JSPs available in the

`/var/opt/SUNWportal/portals/<portal_id>/desktop/default/WirelessJSPLayoutChannel/html` directory and their functionality are:

`layoutSetup.jsp` Displays the layout setup.

`layoutdoedit.jsp` Displays the layout do edit.

`layoutedit.jsp`      Displays the layout edit.

The JSPs available in the  
`/var/opt/SUNWportal/portals/<portal_id>/desktop/default/WirelessJSPLayoutChannel/wml`  
directory and their functionality are:

`layoutdoedit.jsp`      Displays the layout do edit.

`layoutedit.jsp`      Displays the layout edit.

## vxml

The vxml JSPs are located in  
`/var/opt/SUNWportal/portals/<portal_id>/desktop/default/vxml` directory. Under this  
directory, you can find the following JSP.

`docTypeDeclaration.jsp`      Displays the do type declaration.



## JSPs in the community\_sample Directory

---

This chapter contains the following sections:

- “AccountPreferencesContainer” on page 215
- “SearchProvider” on page 216
- “DiscussionProvider” on page 217
- “PagePreferencesContainer” on page 217
- “JSPTableContainerProvider” on page 218
- “JSPEditContainer” on page 219
- “JSPMenuContainerProvider” on page 219
- “JSPs Within the community\_sample Directory” on page 220

### AccountPreferencesContainer

The AccountPreferencesContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/community_sample/AccountPreferencesContainer` directory.

The list of JSPs in AccountPreferencesContainer directory and their description are:

`AccountPreferencesContainer/content.jsp`

The Account Preferences link in the Masthead calls the AccountPreferencesContainer. The `content.jsp` in this container provides initial content by including the `content.jsp` in sub directories such as `timezone`, `password`, and `contactinfo` for changing the respective account preferences.

`AccountPreferencesContainer/process.jsp`

This JSP processes account preferences by forwarding request to `process.jsp` in sub directories for `timezone`, `password`, and `contactinfo`.

AccountPreferencesContainer/timezone/error.jsp  
AccountPreferencesContainer/timezone/content.jsp  
AccountPreferencesContainer/timezone/process.jsp  
AccountPreferencesContainer/timezone/complete.jsp  
AccountPreferencesContainer/timezone/contentFormValidation.jsp

These JSPs handle time zone preferences display (`content.jsp`), processing changes (`process.jsp`), handling process errors (`error.jsp`), the process completion messages (`complete.jsp`), and the client side form validation process (`clientFormValidation.jsp`).

AccountPreferencesContainer/password/contentFormValidation.jsp  
AccountPreferencesContainer/password/complete.jsp  
AccountPreferencesContainer/password/content.jsp  
AccountPreferencesContainer/password/process.jsp  
AccountPreferencesContainer/password/error.jsp

These JSPs handle password preferences display (`content.jsp`), processing changes (`process.jsp`), handling process errors (`error.jsp`), the process completion messages (`complete.jsp`), and the client side form validation process (`clientFormValidation.jsp`).

AccountPreferencesContainer/contactinfo/process.jsp  
AccountPreferencesContainer/contactinfo/complete.jsp  
AccountPreferencesContainer/contactinfo/contentFormValidation.jsp  
AccountPreferencesContainer/contactinfo/error.jsp  
AccountPreferencesContainer/contactinfo/content.jsp

These JSPs handle contactinfo preferences display (`content.jsp`), processing changes (`process.jsp`), handling process errors (`error.jsp`), the process completion messages (`complete.jsp`), and the client side form validation process (`clientFormValidation.jsp`).

## SearchProvider

The Search Provider JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/community_sample/SearchProvider` directory.

The list of JSPs in SearchProvider directory and their description are:



SearchProvider/results.jspSearchProvider/pageFooter.jsp  
 SearchProvider/browseSearch.jspSearchProvider/browseHeader.jsp  
 SearchProvider/searchDoEdit.jspSearchProvider/descMenu.jsp  
 SearchProvider/browseSearchResults.jspSearchProvider/searchEdit.jsp  
 SearchProvider/basicSearch.jspSearchProvider/browseOnly.jsp  
 SearchProvider/searchMenu.jspSearchProvider/psSearch.jsp  
 SearchProvider/processRating.jspSearchProvider/searchOnly.jsp  
 SearchProvider/searchContent.jspSearchProvider/catMenu.jsp  
 SearchProvider/browseResults.jspSearchProvider/dbMenu.jsp  
 SearchProvider/rating.jspSearchProvider/score.jspSearchProvider/error.jsp  
 SearchProvider/advancedSearch.jspSearchProvider/advQuery.jsp

These JSPs handle search provider functionality in the community sample.

## DiscussionProvider

The Discussion Provider JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/community_sample/DiscussionProvider` directory.

The list of JSPs in DiscussionProvider directory and their description are:

DiscussionProvider/viewDiscussionNavigation.jsp  
 DiscussionProvider/discussiondoEdit.jsp  
 DiscussionProvider/viewDiscussionDisplay.jspDiscussionProvider/rating.jsp  
 DiscussionProvider/error.jspDiscussionProvider/viewDiscussionBar.jsp  
 DiscussionProvider/fullDiscussion.jspDiscussionProvider/viewDiscussion.jsp  
 DiscussionProvider/pageFooter.jspDiscussionProvider/viewDiscussionHeader.jsp  
 DiscussionProvider/declare.jspDiscussionProvider/fullDiscussionDisplay.jsp  
 DiscussionProvider/ratingProcess.jspDiscussionProvider/feedbackProcess.jsp  
 DiscussionProvider/searchUI.jspDiscussionProvider/feedback.jsp  
 SearchProvider/feedbackForm.jspSearchProvider/discussionEdit.jsp  
 DiscussionProvider/query.jspDiscussionProvider/portal.jsp  
 DiscussionProvider/feedbackDisplay.jspDiscussionProvider/discussionContent.jsp

These JSPs handle discussion provider functionality in the community sample.

## PagePreferencesContainer

The JSPs in the PagePreferencesContainer are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/community_sample/PagePreferencesContainer` directory.

The list of JSPs in PagePreferencesContainer directory and their description are:

PagePreferencesContainer/process.jsp

This JSP controls the PagePreferencesContainer process. This JSP processes functionality by forwarding request to sub directories such as content and layout process jsps.

PagePreferencesContainer/content.jsp

This JSP controls the PagePreferencesContainer content display by including content from sub directories such as content and layout content jsps.

PagePreferencesContainer/content/process.jsp

PagePreferencesContainer/content/complete.jsp

PagePreferencesContainer/content/contentFormValidation.jsp

PagePreferencesContainer/content/error.jsp

PagePreferencesContainer/content/content.jsp

These jsps handle content display (content.jsp), processing changes (process.jsp), handling process errors (error.jsp), the process completion messages (complete.jsp), and the client side form validation process (clientFormValidation.jsp).

PagePreferencesContainer/layout/content.jsp

PagePreferencesContainer/layout/complete.jsp

PagePreferencesContainer/layout/contentFormValidation.jsp

PagePreferencesContainer/layout/process.jsp

PagePreferencesContainer/layout/tableContent.jsp

PagePreferencesContainer/layout/layout3.jsp

PagePreferencesContainer/layout/layout2.jsp

PagePreferencesContainer/layout/error.jsp

PagePreferencesContainer/layout/menuContent.jsp

PagePreferencesContainer/layout/layout1.jsp

These JSPs handle layout display (content.jsp), processing changes (process.jsp), handling process errors (error.jsp), the process completion messages (complete.jsp), and the client side form validation process (clientFormValidation.jsp). These JSPs also include display of channels in supported containers (menuContent.jsp) and (tableContent.jsp) with supported layouts for tableContent including layout1.jsp, layout2.jsp, and layout3.jsp.

## JSPTableContainerProvider

The JSPTableContainerProvider JSPs are located in the

/var/opt/SUNWportal/portals/<portal\_id>/desktop/community\_sample/JSPTableContainerProvider directory.

The list of JSPs in JSPTableContainerProvider directory and their description are:

JSPTableContainerProviderProvider/popupChannelMenubar.jsp

This is the JSP for menu in channel titlebar area when the channel is detached.

JSPTableContainerProviderProvider/ChannelMenubar.jsp

This is the JSP for menu in channel titlebar area.

JSPTableContainerProviderProvider/Channel.jsp

This is the JSP for a channel.

JSPTableContainerProviderProvider/column.jsp

This is the JSP for a table container column of channels.

JSPTableContainerProviderProvider/popupChannel.jsp

This is the JSP processes the channel content if a channel is detached.

JSPTableContainerProviderProvider/table.jsp

This is the controlling JSP for a table container in the community sample.

JSPTableContainerProviderProvider/tableFullWindow.jsp

This is the JSP for when one of the channels in the container is maximized.

JSPTableContainerProviderProvider/launchPopup.jsp

This JSP includes JavaScript, which allows for a channel to be detached.

## JSPEditContainer

Developer Sample Anonymous Desktop JSPs for the JSPEditContainer are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/community_sample/JSPEditContainer` directory.

The list of JSPs in JSPEditContainer directory and their description are:

<code>JSPEditContainer/edit.jsp</code>	This JSP is the wrapper for channel edit functionality.
<code>JSPEditContainer/doedit.jsp</code>	This JSP handles processing of edited channel preferences by forwarding request to a channel.

## JSPMenuContainerProvider

Developer Sample Anonymous Desktop JSPs for the JSPMenuContainerProvider are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/community_sample/JSPMenuContainerProvider` directory.

The list of JSPs in JSPMenuContainerProvider directory and their description are:

<code>JSPMenuContainerProvider/menumain.jsp</code>	This is the controlling JSP for menu container with left side navigation and right side selected provider or Portlet content.
<code>JSPMenuContainerProvider/menu.jsp</code>	This JSP is the left side navigation for menu container.

## JSPs Within the community\_sample Directory

The JSPs available within the Community Sample Directory are:

`singlePreferenceHeader.jsp`

This JSP is available in the

`/var/opt/SUNWportal/portals/<portal_id>/desktop/community_sample/singlePreferenceHeader.jsp`. This JSP is the masthead included with the `JSPDynamicSingleContainer/dynamicSingle.jsp`.

`singlePreferenceMenubar.jsp`

This JSP is available in the

`/var/opt/SUNWportal/portals/<portal_id>/desktop/community_sample/singlePreferenceMenubar.jsp`. This JSP is the menubar included with the `JSPDynamicSingleContainer/dynamicSingle.jsp`.

`PortletEdit.jsp`

This JSP is available in the

`/var/opt/SUNWportal/portals/<portal_id>/desktop/community_sample/PortletEdit.jsp`. This JSP is the wrapper for Portlet edit functionality.

`datetime.jsp`

This JSP is available in the

`/var/opt/SUNWportal/portals/<portal_id>/desktop/community_sample/datetime.jsp`. This JSP is included in the masthead for current date/time.

`breadcrumb.jsp`

This JSP is available in the

`/var/opt/SUNWportal/portals/<portal_id>/desktop/community_sample/breadcrumb.jsp`. This JSP is included in the Masthead for breadcrumb functionality.

`header.jsp`

This JSP is available in the

`/var/opt/SUNWportal/portals/<portal_id>/desktop/community_sample/header.jsp`. This JSP is the Masthead for all Community Sample Containers.

`JSPDynamicSingleContainer/dynamicSingle.jsp`

This JSP is available in the

`/var/opt/SUNWportal/portals/<portal_id>/desktop/community_sample/JAPDynamicSingleContainer/dynamicSingle.jsp`. This JSP is the clone of `CommunityParentContainer/single.jsp` for search and discussion providers.

`Login/content.jsp`

This JSP is available in the

`/var/opt/SUNWportal/portals/<portal_id>/desktop/community_sample/Login/content.jsp`. This JSP is the channel content for login form.

### CommunityParentContainer/single.jsp

This JSP is available in the

`/var/opt/SUNWportal/portals/<portal_id>/desktop/community_sample/`

`CommunityParentContainer/single.jsp`. This is the controlling JSP for community sample. This JSP includes masthead (`header.jsp`), content for selected community, and static footer (`footer.html`).



## JSPs in the enterprise\_sample Directory

---

This chapter lists the subdirectories and the JSPs under the `/var/opt/SUNWportal/portals/<portal_id>/desktop/enterprise_sample/` directory.

- “ASCTabContainer” on page 223
- “AccountPreferencesContainer” on page 224
- “FullPageContainer” on page 225
- “JSPEditContainer” on page 225
- “JSPTableContainerProvider” on page 225
- “OnlineHelpContainer” on page 226
- “PagePreferencesContainer” on page 226
- “JSPs Within the enterprise\_sample Directory” on page 227

### ASCTabContainer

The ASCTabContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/enterprise_sample/ASCTabContainerProvider` directory.

The ASCTabContainer is the Portal Desktop Parent Container for the Enterprise Sample. The ASCTabContainer JSPs are used to construct the Portal Desktop page.

The list of JSPs in the ASCTabContainer directory and their description are:

<code>anonymous.jsp</code>	This JSP displays the login form for the anonymous user.
<code>body.jsp</code>	This JSP displays the corporate logo and determines whether or not to display the anonymous user logged in the page. This JSP dynamically includes the <code>anonymous.jsp</code> and <code>loggedIn.js</code> .
<code>loggedIn.jsp</code>	This JSP displays the user information for the authenticated user and the logout link.
<code>primaryContainer.jsp</code>	This JSP displays the primary container or top level tabs.

<code>secondaryContainer.jsp</code>	This JSP displays the secondary containers or nested tabs for the current selected primary container.
<code>tab.jsp</code>	This JSP displays the tabs. This JSP dynamically includes the <code>primaryContainer.jsp</code> and the <code>secondaryContainer.jsp</code> .
<code>user.jsp</code>	This JSP determines whether the user is an anonymous user or not.

## AccountPreferencesContainer

The AccountPreferencesContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/enterprise_sample/AccountPreferencesContainer` directory.

The AccountPreferencesContainer JSPs are used to maintain the users account preference settings for contact information, password, and timezone.

The list of JSPs in the AccountPreferencesContainer directory and their description are:

<code>content.jsp</code>	This JSP displays the appropriate page for contact information, password, or timezone settings.
<code>process.jsp</code>	This JSP processes the request for contact information, password, or timezone settings.
<code>title.jsp</code>	This JSP displays the title for Account Settings page.

The AccountPreferencesContainer contact information JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/enterprise_sample/AccountPreferencesContainer/contactinfo` directory.

The AccountPreferencesContainer password JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/enterprise_sample/AccountPreferencesContainer/password` directory.

The AccountPreferencesContainer timezone JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/enterprise_sample/AccountPreferencesContainer/timezone` directory.



## FullPageContainer

The FullPageContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/enterprise_sample/FullPageContainer` directory.

The FullPageContainer is used to display the channel content in a full page view.

The JSP available in the FullPageContainer directory and its description is:

`content.jsp` This JSP displays the channel content in a full page view.

## JSPEditContainer

The JSPEditContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/enterprise_sample/JSPEditContainer` directory.

The JSPs in the EditContainer are used when the Edit icon is selected in a channel title bar inside a JSP based container. The channels that have the `editType` defined as `EDIT_SUBSET` to use these JSPs. The difference between the EditContainer JSP and the default directory JSP is that the EditContainer JSP includes the Enterprise Sample style.

The list of JSPs in the JSPEditContainer directory and their description are:

`edit.jsp` This JSP displays the Edit view of a channel and provides a wrapper around the actual Edit view for a given channel.

`doedit.jsp` This JSP processes the edit request.

## JSPTableContainerProvider

The JSPTableContainerProvider JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/enterprise_sample/JSPTableContainerProvider` directory.

The list of JSPs in the JSPTableContainerProvider directory and their description are:

`channel.jsp` This JSP wraps the channel to display the channel menubar and content view. This JSP dynamically includes the `channelMenubar.jsp`.

`channelMenubar.jsp` This JSP displays the menubar in the channel.

<code>column.jsp</code>	This JSP is responsible for each column in the table layout and to retrieve the channel content.
<code>launchPopup.jsp</code>	This JSP detaches the channel into a separate window.
<code>popupChannel.jsp</code>	This JSP displays the content view of the detached channel.
<code>popupChannelMenubar.jsp</code>	This JSP displays the menubar of the detached channel.
<code>table.jsp</code>	This JSP is responsible for the table layout and displays the content view of the table container. This JSP dynamically includes the <code>column.jsp</code> .
<code>tableFullWindow.jsp</code>	This JSP displays the content view of the table container for single maximized channel.

## OnlineHelpContainer

The OnlineHelpContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/enterprise_sample/OnlineHelpContainer` directory.

The list of JSPs in the OnlineHelpContainer directory and their description are:

<code>content.jsp</code>	This JSP displays the online help.
<code>title.jsp</code>	This JSP displays the title for online help page.

## PagePreferencesContainer

The PagePreferencesContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/enterprise_sample/PagePreferencesContainer` directory.

The PagePreferencesContainer JSPs are used to maintain the users page preference settings for the currently selected tab. These JSPs allow the user to change the page channel content and channel layout.

The list of JSPs in the PagePreferencesContainer directory and their description are:

<code>content.jsp</code>	This JSP displays the appropriate page for contact information, password, or timezone settings.
<code>process.jsp</code>	This JSP processes the request for the content or layout page.
<code>title.jsp</code>	This JSP displays the title for the Page Preferences page.

The PagePreferencesContainer content JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/enterprise_sample/PagePreferencesContainer/content` directory.

The PagePreferencesContainer layout JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/enterprise_sample/PagePreferencesContainer/layout` directory.

## JSPs Within the enterprise\_sample Directory

The JSPs available within the `enterprise_sample` directory are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/enterprise_sample` directory.

The list of JSPs within the `enterprise_sample` directory and their description are:

<code>body.jsp</code>	This JSP displays the corporate logo and determines whether to display the anonymous user in the logged in page. This JSP dynamically includes the <code>anonymous.jsp</code> or the <code>loggedIn.jsp</code> .
<code>channelFullPageContent.jsp</code>	This JSP displays channel content for a full page view.
<code>header.jsp</code>	This JSP displays the header information. This JSP dynamically includes the <code>user.jsp</code> , <code>meta.jsp</code> , <code>title.jsp</code> , <code>script.jsp</code> , and <code>body.jsp</code> .
<code>loggedIn.jsp</code>	This JSP displays the user information for the authenticated user and the logout link.
<code>logout.jsp</code>	This JSP displays the logout link.
<code>meta.jsp</code>	This JSP displays the HTML Meta information.
<code>script.jsp</code>	This JSP displays the JavaScript functions.
<code>title.jsp</code>	This JSP displays the page title.
<code>user.jsp</code>	This JSP determines whether the user is an anonymous user or not.



## JSPs in the developer\_sample Directory

---

This chapter lists the sub directories and the JSPs available under the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/` directory.

- “FrameTabContainer” on page 230
- “JSPContentContainer” on page 230
- “JSPCreateChannelContainer” on page 231
- “JSPCustomThemeContainer” on page 231
- “JSPDynamicSingleContainer” on page 232
- “JSPEditContainer” on page 232
- “JSPFrameCustomTableContainerProvider” on page 232
- “JSPLayoutContainer” on page 233
- “JSPPopupContainer” on page 233
- “JSPPresetThemeContainer” on page 233
- “JSPSingleContainer” on page 234
- and “JSPTabContainer” on page 234
- “JSPTabCustomTableContainerProvider” on page 234
- “JSPTableContainerProvider” on page 235
- “JSPs Within the developer\_sample Directory” on page 236
- “PredefinedFrontPageFramePanelContainerProvider” on page 237
- “PredefinedFrontPageTabPanelContainerProvider” on page 238
- “PredefinedSamplesFramePanelContainerProvider” on page 239
- “PredefinedSamplesTabPanelContainerProvider” on page 240
- “SampleJSP” on page 241
- “SampleSimpleWebService” on page 242

## FrameTabContainer

The FrameTabContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/FrameTabContainer` directory.

The list of JSPs in the FrameTabContainer directory and their description are:

<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

The JSPContentContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/JSPContentContainer` directory. These JSPs are used for the content view when the Content link is selected in a JSP-based table container.

The JSP available in the JSPContentContainer directory and its description is:

`contentedit.jsp` Displays the content Edit page.

## JSPCreateChannelContainer

The JSPCreateChannelContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/JSPCreateChannelContainer` directory.

The list of JSPs in the JSPCreateChannelContainer directory and their description are:

<code>createchannel.jsp</code>	These JSPs are used for the User Defined Channel page on the Desktop.
<code>createchannelcontent.jsp</code>	To create content in the User Defined Channel page.
<code>createchanneldoedit.jsp</code>	To edit content in the User Defined Channel page.
<code>createchanneledit.jsp</code>	To edit content in the User Defined Channel page.
<code>createchannelui.jsp</code>	To create user interface content in the User Defined Channel page.
<code>deletechannel.jsp</code>	To delete a User Defined Channel page.
<code>deletechannelui.jsp</code>	To delete a user interface in the User Defined Channel page.

## JSPCustomThemeContainer

The JSPCustomThemeContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/JSPCustomThemeContainer` directory. These JSPs are used when the Custom Theme link is selected in the Themes page.

The list of JSPs in the JSPCustomThemeContainer directory and their description are:

<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

The JSPDynamicSingleContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/Developer_sample/JSPDynamicSingleContainer` directory.

The JSP available in the JSPDynamicSingleContainer directory and its description is:

`dynamicSingle.jsp`      Used by the DynamicSingleContainer to display the channel specified in the request parameter. This JSP uses the Desktop theme.

## JSPEditContainer

The JSPEditContainer JSP is located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/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 JSP available in the JSPEditContainer directory and its description is:

`edit.jsp`      Displays the Edit view of a channel. Also provides a wrapper around the actual Edit view for a given channel.

## JSPFrameCustomTableContainerProvider

The JSPFrameCustomTableContainerProvider JSP is located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/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 JSP available in the JSPFrameCustomTableContainerProvider directory and its description is:

`framecustomtable.jsp`      Displays the content for the newly created page (table container) from the Sections page.



## JSPLayoutContainer

The JSPLayoutContainer JSP is located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/JSPLayoutContainer` directory. These JSPs are used to display the Layout view when the Layout link is selected in a JSP-based table container.

The JSPs in the JSPLayoutContainer directory and its description is:

`layoutedit.jsp`     Displays the Layout Edit page.

## JSPPopupContainer

The JSPPopupContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/JSPPopupContainer` directory. These JSPs were used to detach a channel from a JSP-based Desktop. However, this container is no longer used by the developer sample. The detached windows in developer sample are now drawn by the `popup.jsp`, `popupmenubar.jsp`, and `providerwrapper.jsp` in JSPTableContainer. These JSPs are present for backward compatibility only.

The list of JSPs in the JSPPopupContainer directory and their description are:

`popup.jsp`                     Displays the contents of the channel inside the detached window.

`popupMenubar.jsp`             Displays the Update, Close, and Logout links inside the popup window.

`providerWrapper.jsp`         Combines the above JSPs.

## JSPPresetThemeContainer

The JSPPresetThemeContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/JSPPresetThemeContainer` directory. These JSPs are used when the Preset Themes link is selected in the Themes page.

The list of JSPs in the JSPPresetThemeContainer directory and their description are:

`themedoedit.jsp`             Processes the result from the Preset Theme page.

`themedit.jsp`                 Displays the Preset Theme Edit page.

`themepreview.jsp`            This file exists for backward compatibility and shows that the theme changes apply to a channel only.

## JSPSingleContainer

The JSPSingleContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/JSPSingleContainer` directory.

The list of JSPs in the JSPSingleContainer directory and their description are:

<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

The JSPTabContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/JSPTabContainer` directory.

The list of JSPs in the JSPTabContainer directory and their description are:

<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.

## JSPTabCustomTableContainerProvider

The JSPTabCustomTableContainer JSP is located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/JSPTabCustomTableContainerProvider` directory.

The JSP available in the JSPTabCustomTableContainerProvider directory and their description are:

`tabcustomtable.jsp` Displays the table container's content view.

## JSPTableContainerProvider

The JSPTableContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/JSPTableContainerProvider` directory.

The list of JSPs in the JSPTableContainerProvider directory and their description are:

<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>tabletable.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> .)

## JSPs Within the developer\_sample Directory

These JSPs are located within the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample` 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 JSPs that are available within the `developer_sample` directory and their description are:

<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 `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/PredefinedFrontPageFramePanelContainerProvider` directory.

The list of JSPs in the PredefinedFrontPageFramePanelContainerProvider directory and their description are:

<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 `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/PredefinedFrontPageTabPanelContainerProvider` directory.

The list of JSPs in the `PredefinedFrontPageTabPanelContainerProvider` directory and their description are:

<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 `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/PredefinedSamplesFramePanelContainerProvider` directory.

The list of JSPs in the `PredefinedSamplesFramePanelContainerProvider` directory and their description are:

<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> .)

## PredefinedSamplesTabPanelContainerProvider

The `PredefinedSamplesTabPanelContainerProvider` JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/PredefinedSamplesTabPanelContainerProvider` directory.

The list of JSPs in the `PredefinedSamplesTabPanelContainerProvider` directory and their description are:

<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

The SampleJSP JSP is located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/SampleJSP` directory. These JSPs are used by the SampleJSP channel.

The JSP available in the SampleJSP directory and its description is:

`samplecontent.jsp` Displays the contents of the Sample JSP channel. This JSP uses of the Desktop theme, and is used as an example of the `channelHighlightColor` attribute.

## SampleSimpleWebService

The SampleSimpleWebService JSP is located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample/SampleSimpleWebService` directory.

The JSP available in the SampleSimpleWebService directory and its description is:

`webserviceContent.jsp` Displays the Content view of the simple web service channels.

# JSPs in the developer\_anonymous\_sample Directory

---

This chapter contains the following sections:

- “FrameTabContainer” on page 243
- “JSPDynamicSingleContainer” on page 244
- “JSPTabContainer” on page 244
- “JSPTabContainerProvider” on page 244
- “PredefinedFrontPageFramePanelContainerProvider” on page 245
- “PredefinedFrontPageTabPanelContainerProvider” on page 246
- “PredefinedSamplesFramePanelContainerProvider” on page 246
- “PredefinedSamplesTabPanelContainerProvider” on page 247

## FrameTabContainer

The JSPs for FrameTabContainer are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/desktop_anonymous_sample/FrameTabContainer` directory.

The list of JSPs in the FrameTabContainer and their description are:

<code>frametab.jsp</code>	This JSP is the developer sample 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 developer sample 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 developer sample anonymous version of the <code>menubar.jsp</code> for frame tab container with the links for customization deactivated.

## JSPDynamicSingleContainer

The JSPDynamicSingleContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_anonymous_sample/JSPDynamicSingleContainer` directory.

The list of JSPs in the JSPDynamicSingleContainer and their description are:

- |                          |   |
|--------------------------|---|
| <code>header.jsp</code>  | This JSP is the developer sample anonymous version of the <code>header.jsp</code> for the JSP dynamic single container with the links for customization deactivated.  |
| <code>menubar.jsp</code> | This JSP is the developer sample anonymous version of the <code>menubar.jsp</code> for the JSP dynamic single container with the links for customization deactivated. |

## JSPTabContainer

The JSPTabContainer JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_anonymous_sample/JSPTabContainer` directory.

The list of JSPs in the JSPTabContainer and their description are:

- |                          |  |
|--------------------------|--|
| <code>header.jsp</code>  | This JSP is the developer sample 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 developer sample anonymous version of the <code>menubar.jsp</code> for the tab container with the links for customization deactivated. |

## JSPTableContainerProvider

The JSPTableContainerProvider JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_anonymous_sample/JSPTableContainerProvider` directory.

The list of JSPs in the JSPTableContainerProvider and their description are:

- |                          |  |
|--------------------------|--|
| <code>header.jsp</code>  | This JSP is the developer sample 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 developer sample 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

The `PredefinedFrontPageFramePanelContainerProvider` JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_anonymous_sample/PredefinedFrontPageFramePanelContainerProvider` directory.

The list of JSPs in the `PredefinedFrontPageFramePanelContainerProvider` and their description are:

<code>header.jsp</code>	This JSP is the developer sample 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 developer sample 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

The PredefinedFrontPageTabPanelContainerProvider JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_anonymous_sample/PredefinedFrontPageTabPanelContainerProvider` directory.

The list of JSPs in the PredefinedFrontPageTabPanelContainerProvider and their description are:

<code>header.jsp</code>	This JSP is the developer sample 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 developer sample 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

The PredefinedSamplesFramePanelContainerProvider JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_anonymous_sample/PredefinedSamplesFramePanelContainerProvider` directory.

The list of JSPs in the PredefinedSamplesFramePanelContainerProvider and their description are:

<code>header.jsp</code>	This JSP is the developer sample 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 developer sample 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

The `PredefinedSamplesTabPanelContainerProvider` JSPs are located in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_anonymous_sample/PredefinedSamplesTabPanelContainerProvider` directory.

The list of JSPs in the `PredefinedSamplesTabPanelContainerProvider` and their description are:

<code>header.jsp</code>	This JSP is the developer sample 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 developer sample 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> .)





## JSPs in the welcome\_sample Directory

---

This chapter lists the subdirectory and the JSPs under the welcome\_sample directory.

- [“RedirectChannel” on page 249](#)

### RedirectChannel

RedirectChannel is the subdirectory under the /var/opt/SUNWportal/portals/<portal\_id>/desktop/welcome\_sample/ directory. There are three other subdirectories such as aml, chtml, and wml under RedirectChannel. All these subdirectories contain the content.jsp.

The JSP available in the welcome\_sample directory and its description is:

content.jsp    This JSP redirects the user to the Portal welcome page.



PART V

Rewriter

- Chapter 33
- Chapter 34
- Chapter 35



## Overview of Rewriter

---

The Sun Java System Portal Server 7.1 Rewriter provides an engine for performing URL translation in markup languages and JavaScript code. The URLScrapperProvider and the XMLProvider in the Desktop and the Sun Java System Portal Server 7.1: Sun Java System Portal Server 7.1 SRA gateway service all use the Rewriter service.

Rewriter scans the content of web pages and identifies the URLs it finds on those web pages. It uses a collection of rules defined in a ruleset to determine the elements of a web page to rewrite. Once Rewriter identifies a URL it can rewrite the URL by:

- “Expanding Relative URLs to Absolute URLs” on page 253
- “URLScrapperProvider Limitations” on page 254
- “Prefixing the Gateway URL to an Existing URL” on page 254
- “Attributes” on page 255

### Expanding Relative URLs to Absolute URLs

The URLScrapperProvider is part of the core Portal Server 7.1 product. In a non-gateway scenario, the URLScrapperProvider can be used to expand relative URLs to absolute URLs. For example, if a user is trying to access the site:

```
<a href=" ../mypage.html">
```

The Rewriter translates this to:

```
<a href="http://www.yahoo.com/mail/mypage.html">
```

where `http://www.yahoo.com/mail/` is the base URL of the page scraped.

## URLScrapperProvider Limitations

The URLScrapperProvider simply tries to display a designated URL in a channel. There's no way to specify parts of a document URL (document) to display. The URLScrapperProvider acts much like an HTTP client, in that it makes a request for the content of the specified URL. Just like in a browser, the target URL to scrape must be network visible, or you must have a proxy configured.

The resultant URL scraper channel, however, is not a mini-browser nor is it a frame. Therefore, if you have a link in the content, it effects the whole page, not just the channel. You should not browse inside the URL scraper channel. If you select a link within the channel the browser can interpret the link and replace the currently displayed page (your portal server Desktop) with the contents of the link location.

The appearance of the scraped channel is controlled by whatever is producing the original content. The URLScrapperProvider does not modify the content at all and only displays whatever is available through the URL. Since the channel is essentially a cell in an HTML table, it can only display HTML content that is legal to appear in table cells. That is, a frameset cannot be scraped using the URLScrapperProvider because a <FRAMESET> tag cannot appear within a <BODY> tag. The URLScrapperProvider will also not execute JavaScript code in <HEAD> tags. Because of this, the following scraping scenarios are inappropriate for the URLScrapperProvider:

- When an Edit function of some sort is required so that the user can customize the channel.
- When the data comes from a non-HTML, non-web server source, that is, a database or mail server.
- When the data needs to be reformatted in some way for the channel.
- When a more efficient solution is required as the URLScrapperProvider will do a request and look up for every Desktop display and user.

When cookies are sent by the origin server, they are forwarded back everytime web content is re-scraped. So the origin should get the cookies it sent as the web content scraped the first time, when portal desktop is updated or reloaded. But those cookies are not expected to be sent back when user clicks on any links in the url scraper channel.

## Prefixing the Gateway URL to an Existing URL

In an implementation with a gateway such as the Sun Java System Portal Server 7.1 SRA, the gateway acts as a proxy for the client and accesses intranet sites and returns responses to the client. The Rewriter translates URLs in downloaded pages so that they point back to the gateway rather than to the original site by prefixing the gateway URL to the existing URL.

For example, if a user tries to access an HTML page on mymachine using the following URL:

```
<a href="http://mymachine.intranet.com/mypage.html">
```

The Rewriter prefixes this URL with a reference to the gateway as follows:

```
<a href="https://gateway.company.com/http://mymachine.intranet.com/mypage.html">
```

When a user selects a link associated with this anchor, the browser contacts the gateway. The gateway fetches the content of `mypage.html` from `mymachine.intranet.com`.

See the *Sun Java System Portal Server 7.1 SRA Administration Guide* for more information on using the Rewriter to prefix a gateway URL to an existing URL.

## Attributes

The Rewriter provides a Java class library for rewriting URL references in various web languages such as HTML, JavaScript, and WML, and in HTTP Location headers (redirections). The Rewriter Service does not consist of any attributes. To implement the service, create Rewriter rules that define how rewriting is to be done and the data to be rewritten.





# Supported URLs

---

This chapter discusses about the supported URLs.

## Supported URLs

Rewriter supports rewriting of all standard URLs as specified by RFC-1738. These URLs are supported whether the protocol is HTTP or HTTPS and regardless of the capitalization of the protocol. For example, hTtP, HTtp, and httP are all valid. Some sample standard URLs are listed below:

```
http://www.my.sesta.com
http://www.example.org:8000/imaginary/test
http://www.example.edu/org/admin/people#andy
http://info.example.org/AboutUs/Index/Phonebook?dobbins
http://www.example.org/RDB/EMP?*%20where%20name%3Dobbins
http://info.example.org/AboutUs/Phonebook
http://user:password@example.com
```

Rewriter supports rewriting of some basic non-standard URLs. The information to convert non-standard URLs to a standard format is taken from the base URL of the page where the URL appears and can include the protocol, host name, and path. The back slash (d) is supported only when it is part of a relative URL and not part of an absolute URL. For example, `http://sesta.com\\index.html` is rewritten, but `http:\\\\sesta.com` is not.

In addition, URLs with a single slash (/) after the protocol or scheme such as `http:/sesta.com` are not rewritten.



# Defining Rewriter Rules and Rulesets

---

This chapter contains the following sections

- “Overview” on page 259
- “Rules for HTML Content” on page 260
- “Rules for JavaScript Content” on page 263
- “Rules for XML Content” on page 266

## Overview

The Rewriter modifies the URL portions of various elements that appear on a web page. The Rewriter comes with a default set of rules to determine the elements of a web page to rewrite. A collection of rules for various categories and subcategories is stored in a `.dtd` file and is called a ruleset. The Rewriter rulesets are defined in XML.

The DTD is located in `/opt/SUNWportal/web-src/WEB-INF/lib/rewriter.jar` (`resources/RuleSet.dtd`). Rulesets are used to identify URLs. By default, all strings in web content starting with characters such as “/”, “./”, “http” and “https” are considered to be URLs and are candidates for rewriting.

To configure the Rewriter for your implementation, you create a ruleset and define rules in the Rewriter section of the Portal Server Configuration in the administration console. See *Administering the Rewriter Service* for details on creating and modifying rulesets. You define multiple rules based on the content type in the web pages. For example, the rule required to rewrite HTML content would be different from the rule required to rewrite JavaScript content. Rewriter rules fall into the following categories:

- Rules for HTML Content
- Rules for JavaScript Content
- Rules for XML Content

---

**Note** – As Wireless Markup Language (WML) is similar to HTML, HTML rules are applied for WML content.

No rules are required for CSS content.

---

The ruleset is an XML document and the XML within it must be properly formed. When defining rules in a ruleset, follow these guidelines:

- All rules need to be enclosed within the `<ruleset>` `</ruleset>` tags.
- Include all rules to rewrite HTML content in the `<HTML>` `</HTML>` section of the ruleset.
- Include all rules to rewrite JavaScript content in the `<JSRules>` `</JSRules>` section of the ruleset.
- Include all rules to rewrite XML content in the `<XML>` `</XML>` section of the ruleset.

## Rules for HTML Content

HTML content in web pages can be classified into attributes, JavaScript tokens, forms, and applets. Accordingly, the rules for HTML content are classified as:

- [“Attribute Rules for HTML Content” on page 260](#)
- [“JavaScript Token Rules for HTML Content” on page 261](#)
- [“Form Rules for HTML Content” on page 262](#)
- [“Applet Rules for HTML Content” on page 262](#)

## Attribute Rules for HTML Content

Attribute rules identify the basic attribute tags in HTML pages to rewrite. Rewriter modifies the various occurrences of the defined tags by expanding or prefixing the existing URL. The default ruleset rewrites the following attribute tags:

- action
- background
- codebase
- code
- href
- src
- value
- imagePath
- lowsrc
- archive

The syntax for attribute rules is:

```
<Attribute name="name" [tag="tag" valuePatterns="patterns"]
```

where name specifies the attribute, tag specifies the tag to which the attribute belongs (set to \* to match all tags), and patterns specifies the possible patterns to match with the attribute. The tag and valuePatterns parameters are optional.

## JavaScript Token Rules for HTML Content

Web pages can contain pure JavaScript code within the JavaScript tags, or they can contain JavaScript tokens or functions. For example, a web page can contain an onClick() function that causes a jump to a different URL. In order for the page to function properly, the value of the onClick() function needs to be translated and rewritten. In most cases, the rules provided in the default ruleset are sufficient to rewrite the URLs in JavaScript tokens. The default ruleset rewrites the following JavaScript tokens:

- onAbort
- onBlur
- onChange
- onClick
- onDbClick
- onError
- onFocus
- onKeyDown
- onKeyPress
- onKeyUp
- onLoad
- onMouseDown
- onMouseMove
- onMouseOut
- onMouseOver
- onMouseUp
- onReset
- onSelect
- onSubmit
- onUnload

The syntax for JavaScript Token rules is:

```
<JSToken>javascript_function_name</JSToken>
```

where javascript\_function\_name is the name of the function such as onLoad or onClick.

## Form Rules for HTML Content

Users can browse HTML pages that contain forms. Form elements, such as input, can take a URL as a value. The default ruleset does not rewrite any form elements. The syntax for form rules is:

```
<Form source="/source.html" name="form1" field="field1"> [valuePatterns="pattern"] />
```

where /source.html is the URL of the HTML page containing the form, form1 is the name of the form, field1 is the field of the form to be rewritten, and pattern indicates the part of the field to be rewritten. All content that follows the pattern specified is rewritten.

The valuePatterns parameter is optional.

## Applet Rules for HTML Content

A single web page can contain many applets, and each applet can contain many parameters. The Rewriter rule for URLs in applets should contain pattern matching information for the following:

- source, such as filename.htm
- code, such as classname.class
- parameter name, such as servername
- parameter value, such as some\_url

Rewriter matches the values specified in the rule with the content of the applet and modifies the URLs as required. This replacement is carried out at the server and not when the user is browsing the particular web page. A wildcard character (\*) can also be used as part of the rule. For example, the parameter name could be \*, in which case, the Rewriter does not compare the parameter name in the applet.

The default ruleset does not rewrite any applet parameters.

The syntax for applet rules is:

```
<Applet source="sourcehtml.jsp" code="class" param="parameter_name"  
[valuePatterns="pattern"]
```

where /sourcehtml.jsp is the URL containing the applet, class is the name of the applet class, parameter\_name is the parameter whose value needs to be rewritten, and pattern indicates the part of the field to be rewritten. All content that follows the pattern specified is rewritten. The valuePatterns parameter is optional.

# Rules for JavaScript Content

URLs can occur in various portions of JavaScript code. The Rewriter cannot directly parse the JavaScript code and determine the URL portion. A special set of rules needs to be written to help the JavaScript processor translate the URL.

JavaScript elements that contain URLs are classified as follows:

- “JavaScript Variables” on page 263
- “JavaScript Function Parameters” on page 265

## JavaScript Variables

JavaScript variables are again classified into five categories:

- JavaScript URL Variables
- JavaScript EXPRESSION Variables
- JavaScript DHTML Variables
- JavaScript DJS (Dynamic JavaScript) Variables
- JavaScript System Variables

### JavaScript URL Variables

URL variables have a URL string on the right hand side. The default ruleset rewrites the following JavaScript URL variables:

- `imgsrc`
- `location.href`
- `_fr.location`
- `mf.location`
- `parent.location`
- `self.location`

The syntax of URL variables in JavaScript content rules is:

```
<Variable type="URL">variable_name</Variable>
```

where `variable_name` is the name of the variable to be rewritten.

### JavaScript EXPRESSION Variables

EXPRESSION variables have an expression on the right hand side. The result of this expression is a URL. The Rewriter appends a JavaScript function for converting the expression to the HTML page as it cannot evaluate such expressions. This function takes the expression as a parameter and evaluates it at the client browser.

The default ruleset rewrites the location JavaScript EXPRESSION variable.

The syntax of EXPRESSION variables in JavaScript content rules is:

```
<Variable type="EXPRESSION">variable_exp</Variable>
```

where variable\_exp is the expression variable.

## JavaScript DHTML Variables

DHTML variables are JavaScript variables that hold HTML content. The default ruleset rewrites the following JavaScript DHTML variables:

- document.write
- document.writeln

The syntax of DHTML variables in JavaScript content is:

```
<Variable type="DHTML">variable</Variable>
```

where variable is the DHTML variable.

## JavaScript DJS (Dynamic JavaScript) Variables

DJS (Dynamic JavaScript) variables are JavaScript variables that hold JavaScript content.

The syntax of DJS variables in JavaScript content is:

```
<Variable type="DJS">variable</Variable>
```

where variable is the DJS variable.

The JavaScript code contained in the variable needs another rule to translate it.

## JavaScript System Variables

System variables are variables that are not declared by the user, but that are available as a part of the JavaScript standard.

The default ruleset rewrites the window.location.pathname JavaScript system variable.

The syntax of system variables in JavaScript content is:

```
<Variable type="SYSTEM">variable</Variable>
```

where variable is the system variable.



## JavaScript Function Parameters

Function parameters are classified into four categories:

- JavaScript URL Parameters
- JavaScript EXPRESSION Parameters
- JavaScript DHTML Parameters
- JavaScript DJS Parameters

### JavaScript URL Parameters

URL parameters are string parameters that directly contain the URL.

The default ruleset rewrites the following JavaScript URL parameters:

- openURL
- openAppURL
- openNewWindow
- parent.openNewWindo
- window.open

The syntax for URL parameters is:

```
<Function type = "URL" name = "function" [paramPatterns="y,y,"] />
```

where function is the name of the function to be evaluated and y indicates the position of the parameter(s) that need to be rewritten. Parameter positions are delimited by commas. For example, in the syntax line the first and second parameters need to be rewritten, but the third parameter should not be rewritten.

### JavaScript EXPRESSION Parameters

EXPRESSION parameters are variables within a function that result in a URL when they are evaluated. The syntax for EXPRESSION parameters is

```
<Function type = "EXPRESSION" name = "function" [paramPatterns="y,y,"] />
```

where function is the name of the function to be evaluated and y indicates the position of the parameter(s) that need to be rewritten. Parameter positions are delimited by commas. For example, in the syntax line the first and second parameters need to be rewritten, but the third parameter should not be rewritten.

### JavaScript DHTML Parameters

DHTML parameters are native JavaScript methods that generate an HTML page dynamically. For example, the document.write() method falls under this category.

The default ruleset rewrites the following JavaScript DHTML parameters:

- document.write
- document.writeln

The syntax for DHTML parameters is:

```
<Function type = "DHTML" name = "function" [paramPatterns="y,y,"] />
```

where function is the name of the function to be evaluated and y indicates the position of the parameter(s) that need to be rewritten. Parameter positions are delimited by commas. For example, in the syntax line the first and second parameters need to be rewritten, but not the third parameter should not be rewritten.

## JavaScript DJS Parameters

Dynamic JavaScript (DJS) parameters such as Cascading Style Sheets (CSS) in HTML are also translated. There are no rules defined for this translation as the URL appears only in the url() function of the CSS. The syntax for DJS parameters is:

```
<Function type = "DJS" name = "function" [paramPatterns="y,y,"] />
```

where function is the name of the function to be evaluated and y indicates the position of the parameter(s) that need to be rewritten. Parameter positions are delimited by commas. For example, in the syntax line the first and second parameters need to be rewritten, but not the third parameter should not be rewritten.

## Rules for XML Content

Web pages can contain XML content which in turn can contain URLs and Rewriter can rewrite URLs in XML content.

XML content that contains URLs is classified as follows:

- [“Tag Text in XML” on page 266](#)
- [“Attributes in XML” on page 267](#)

## Tag Text in XML

Rewriter translates XML content based on the tag name.

The default ruleset rewrites the following tags in XML:

- baseroot
- img

The syntax for tag text is:

```
<TagText tag ="attribute" attributePatterns="name=src"/>
```

where `attribute` is the name of the tag and `src` is the name of the attribute.

## Attributes in XML

The rules for attributes in XML are similar to the rules for attributes in HTML. See [“Attribute Rules for HTML Content” on page 260](#) for additional information. Rewriter translates attribute values based on the attribute and tag names.

The default ruleset rewrites the following attributes in XML:

- `xmlns`
- `href`

The syntax for attributes in HTML is:

```
<Attributes>  
<Attribute name="attribute" [valuePatterns="name=src"/>  
</Attributes>
```

where `attribute` is the name of the tag and `src` is the name of the attribute.



PART VI

Sample Portal

- [Chapter 36](#)
- [Chapter 37](#)
- [Chapter 38](#)
- [Chapter 39](#)
- [Chapter 40](#)



# Understanding the Developer Sample

---

This chapter describes the developer sample that you can choose to install on your system during the Sun Java System Portal Server installation. See the Installation Guide for more information on installing the developer sample. This chapter contains the following sections:

- “Overview” on page 271
- “Sample Desktops” on page 273

## Overview

Conceptually, the Desktop is split into the three well-defined components as shown in Figure 37-1.

### 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 `/var/opt/SUNWportal/portals/<portal_id>/desktop/default` directory), and help files.

### Developer Sample

This includes organization level display profile definitions (such as themes and channel display profile definitions), templates and JSPs for the example Desktops in the developer sample (installed in `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample` 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 developer sample 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 Portal Server 7.1 software.

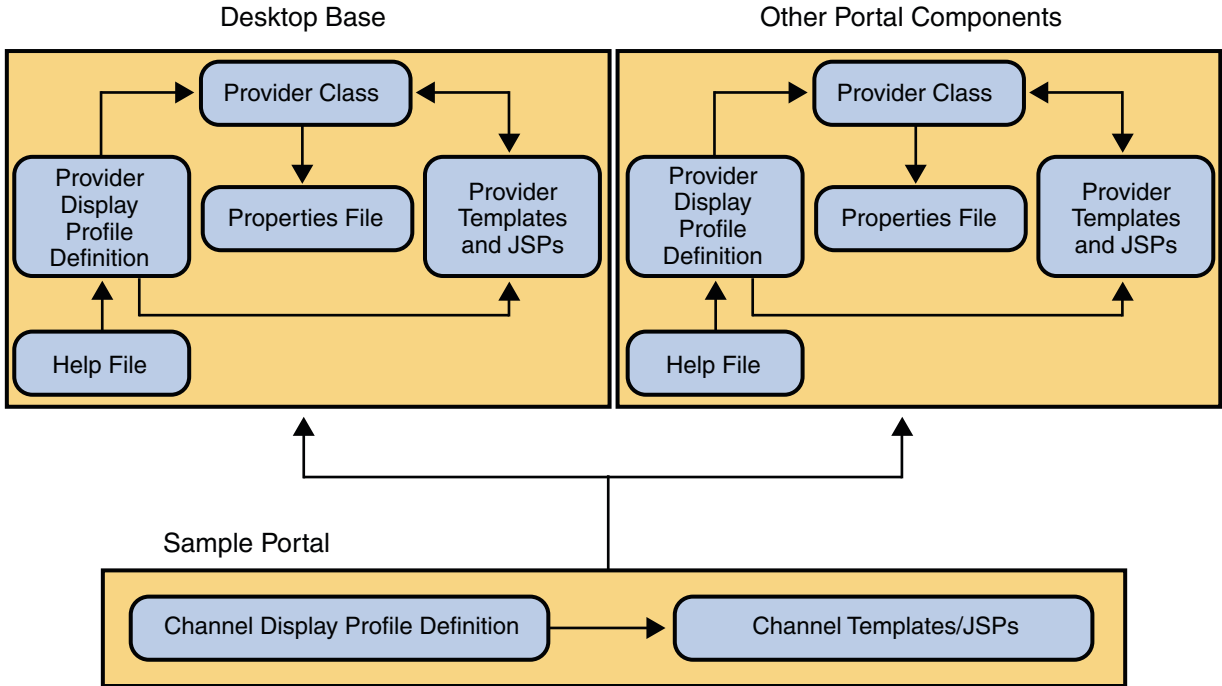


FIGURE 36-1 Desktop Components

When you install the Portal Server 7.1 software, you can choose to install the developer sample. The developer sample is an authentication-less (authless) desktop that consists of containers, channels, portlets, services, and templates which can be used to demonstrate what the Portal Server 7.1 software is capable of. It includes five example Desktops that show the possibilities of the Portal Server 7.1 software. In this way you can quickly get a feel for the kinds of containers that are possible to design.

## Developer Sample Installation Directories

If you choose to install the developer sample, the installer locates the appropriate files in the following directories:



## *PortalServer-base/samples/desktop*

This directory contains the following display profile documents:

- dp-org.xml      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.

The `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample` directory contains the JSP, template, and other support files for the Portal Server 7.1 software developer sample.

The `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_anonymous_sample` contains the JSP, template, and other support files for the Portal Server 7.1 software Desktop anonymous user.

---

**Note** – The developer sample 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 in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/default` directory.

---

## Sample Desktops

The table below shows the three sample Desktops, which make up the developer sample.

TABLE 36-1 Sun Java System Portal Server Sample Desktop Containers

Container (Desktop) Type	Description
JSPTabContainer	Generates a JSP-based tab Desktop. 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	Generates a JSP-based table Desktop. 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	Generates a frame-based Desktop. 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.

The developer sample 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 developer sample 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.

---

**Note** – The Portal Server 7.1 software distinguishes between building-block providers, which you can extend using the Portal Server 7.1 software APIs, and content providers, which you cannot extend. See the Sun Java System Portal Server 7.1 Developer's Guide for more information on extending the providers.

---

# JSP-Based Desktop

---

This chapter contains the following sections:

- “JSPTabContainer” on page 275
- “JSPTableContainer” on page 280

## JSPTabContainer

The JSPTabContainer provides a JSP-based tabbed Desktop.

### Sample Desktop

#### Default Layout

By default, the sample portal Desktop based on the JSPTabContainer includes five tabs, My Front Page, Samples, Search, Collaboration, and Sample Portlet. In the figure below, the channels are shown in numbers.

- My Front Page tab (default) — Login or User Information, Bookmark, Sample JSP, and XML Test channels
- 1 — Samples tab: Sun Information, URL Scraper, and Notes channels
- 2 — Search tab: Search channel
- 3 — Collaboration tab: Discussions Lite and Discussions channels
- 4 — Portlet Samples tab: Bookmark, Showtime, and Weather portlet channels

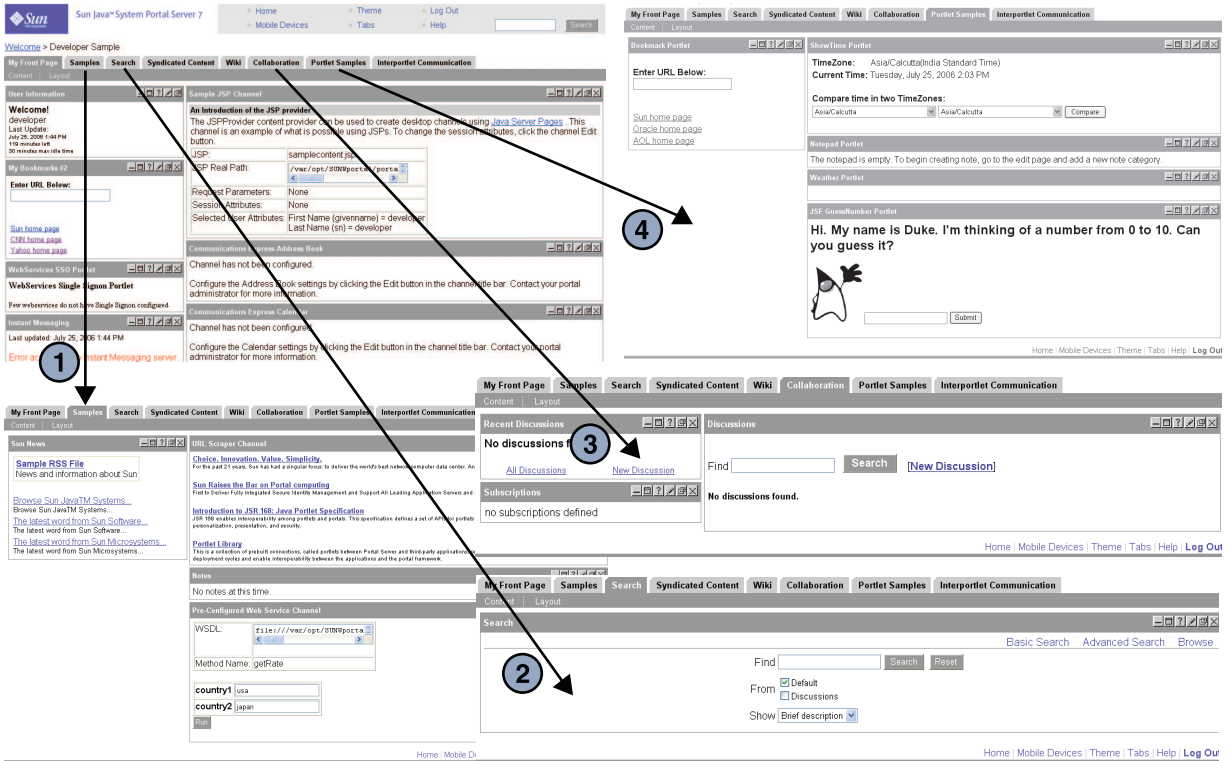


FIGURE 37-1 Sample Desktop Based on JSPTabContainer

## Default Actions

The sample JSPTabContainer channel, by default, includes:

- Banner links to return Home, tabs to allow the user to remove, rename or select the start tab, and also create a new tab (the URL for this page is `action=edit&provider=JSPTabContainer`), theme to allow the user to set the color scheme and font type for the Desktop (the URL for the preset theme page is `action=edit&provider=JSPPresetThemeContainer`), help that displays the Desktop sample online help (the URL is `./docs/locale/desktop/helppage.htm`), Log Out link to log the user out of the Desktop (the URL is `action=logout`), and Search to allow the user to search.

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; and 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.

- Links specific to the contained containers. 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.

## Default Display Profile Settings

The provider responsible for generating the JSPTabContainer channel is JSPTabContainerProvider. 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.

The properties that make up JSPTabContainer work as follows by default.

<code>contentPage</code>	Set to <code>tab.jsp</code> . This draws the Content Page for the tab container.
<code>editPage</code>	Set to <code>tabedit.jsp</code> . This displays the Edit page for the tab container where new tabs can be added, and existing tabs removed or renamed.
<code>startTab</code>	Sets the tab that opens first on the Desktop as <code>MyFrontPageTabPanelContainer</code> .
<code>maxTabs</code>	Allows six tabs to be created. As there are currently five tabs, one more can be added.
<code>makeTabProvider</code>	Used when creating a new tab from scratch.
<code>channelNumber</code>	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 <code>MyFrontPageTabPanelContainer</code> in <code>JSPTabContainer</code> , the new tab channel name would be <code>JSPTabContainer/MyFrontPageTabPanelContainer1</code> . (The new tab name is actually the <code>channelName</code> property in the display profile plus the value of the <code>channelNumber</code> property. The <code>channelNumber</code> is incremented by one each time a new tab is created.)
<code>contentChannel</code>	Specifies <code>JSPContentContainer</code> as the content channel that provides the Content page displaying channels to add to a user-created tab.

<code>presetThemeChannel</code>	Specifies <code>JSPPresetThemeContainer</code> as the channel that is displayed in the Theme - Preset Theme page.
<code>customThemeChannel</code>	Specifies <code>JSPCustomThemeContainer</code> as the channel that is displayed in the Theme - Custom Theme page.
<code>TabProperties</code>	This collection has <code>&lt;Collection name=&gt;</code> entries for each of the available tab defined in <code>JSPTabContainer</code> .
<code>Available</code>	This list describes all available channels for this container. The available channels are displayed in the Content Preference page for users to select from.
<code>Selected</code>	This list describes selected channels for this container. Only selected channels are displayed on the Desktop.

## JSPTabContainer Architecture

Figure 38–2 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.

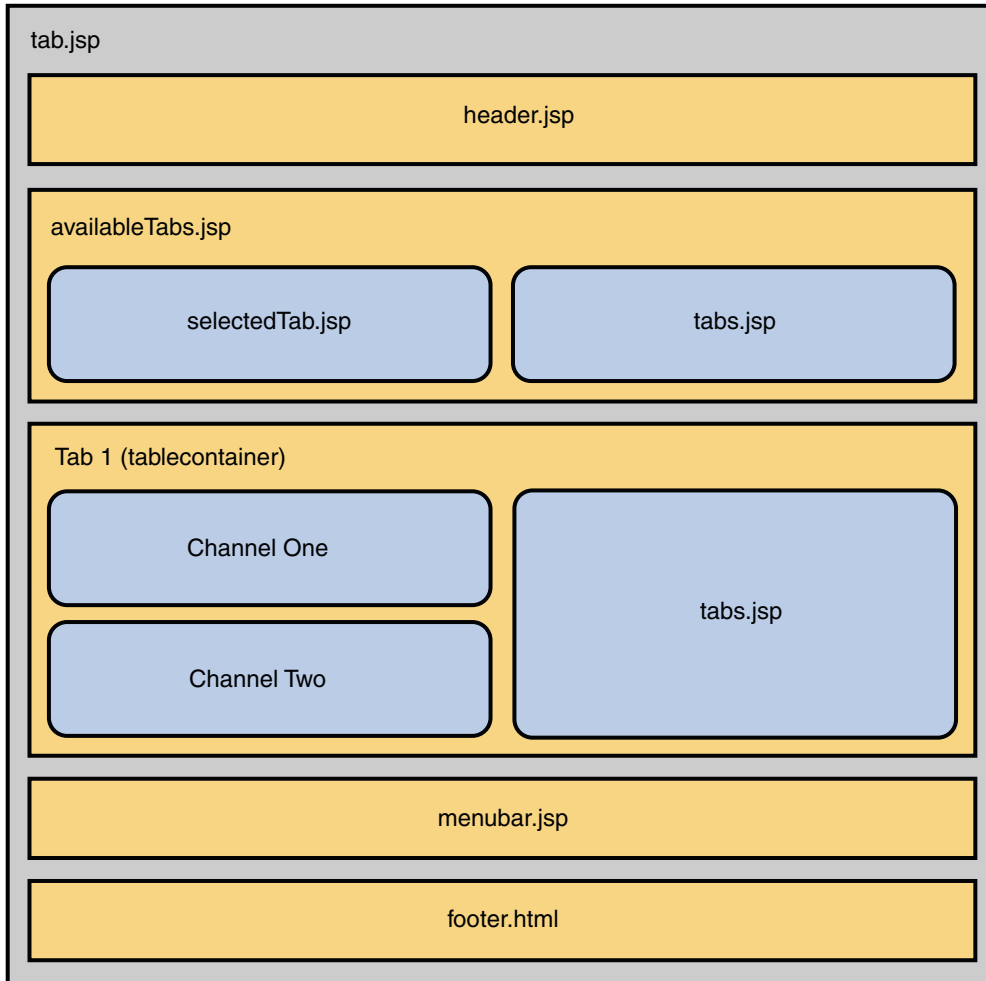


FIGURE 37-2 JSPTabContainer Architecture

## JSP Files Used by JSPTabContainer

The Portal Server 7.1 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.

The Edit page is where you can add, remove, and rename tabs. The `tabedit.jsp` is used to display this page.

# JSPTableContainer

The JSPTableContainer provides a JSP-based table Desktop.

## Sample Desktop

### Default Layout

By default, the sample portal Desktop based on the JSPTableContainer (see Figure 38–3) contains the following channels:

- Thin channels: User Information, Sun Information, My Bookmarks, Mailcheck Provider, and My Applications
- Wide channels: Sample JSP, XML Test, Notes, Personal Notes, and Preconfigured Web Service

**User Information**

**Welcome!**  
 developer  
 Last Update:  
 July 25, 2006 2:30 PM  
 73 minutes left  
 30 minutes max. idle time

**My Bookmarks**

Enter URL Below:

[Sun home page](#)  
[CNN home page](#)  
[Yahoo home page](#)

**My Applications**

[Instant Messenger \(Java Web Start\)](#)  
[Instant Messenger \(Browser\)](#)  
[NetFile](#)

**Sample JSP Channel**

**An Introduction of the JSP provider**  
 The JSPProvider content provider can be used to create desktop channels using [Java Server Pages](#). This channel is an example of what is possible using JSPs. To change the session attributes, click the channel Edit button.

JSP: samplecontent.jsp  
 JSP Real Path: /var/opt/SUNWportal/porta  
 Request Parameters: action=content  
 provider=WirelessDesktopDispatcher  
 Session Attributes: None  
 Selected User Attributes: First Name (givenname) = developer  
 Last Name (sn) = developer

**XML Test Channel**

<b>company22.com</b>			NASDAQ, 15:47
Last	<b>16.240000</b>	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

FIGURE 37–3 Sample Desktop Based on JSPTableContainer



## Default Actions

The sample JSPTableContainer channel, by default, includes:

- Banner links to return to the Desktop Home page, Desktop theme to allow the user to set the color scheme and font type for the Desktop (the URL for this page is `action=edit&provider=JSPPresetThemeContainer`), Log Out to allow the user to log out of the Desktop (the URL for this page is `action=logout`), Help to display the Desktop sample online help (the URL for this page is `../docs/locale/desktop/helppage.htm`), and Search to allow the user to search.
- Leaf channel. JSPTableContainer does not contain any contained 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 Content link

(`action=edit&provider=JSPContentContainer&container=JSPTableContainer`) allows the user to edit the content on the Content page and the Layout link

(`action=edit&provider=JSPLayoutContainer&container=JSPTableContainer`) allows the user to edit the layout of the channels on the Layout page.

## Default Display Profile Settings

The provider responsible for generating the JSPTableContainer channel is JSPTableContainerProvider. 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. The default properties that make up JSPTableContainer work as follows:

<code>contentPage</code>	Set to <code>toptable.jsp</code> . This draws the Content Page for the table container.
<code>categories</code>	This collection 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.
<code>channelsRow</code>	This collection 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.
<code>channelsIsRemovable</code>	This collection defines a collection to contain the <code>isRemovable</code> value for channels in this container. Only one channel, user information, is defined, with a value of false, so that it cannot be removed.

Available	This list describes all available channels for this container. The available channels are displayed in the content preference page for users to select from.
Selected	This list describes selected channels for this container. Only selected channels shows up in the Desktop.

## JSPTableContainer Architecture

Figure 38–4 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.

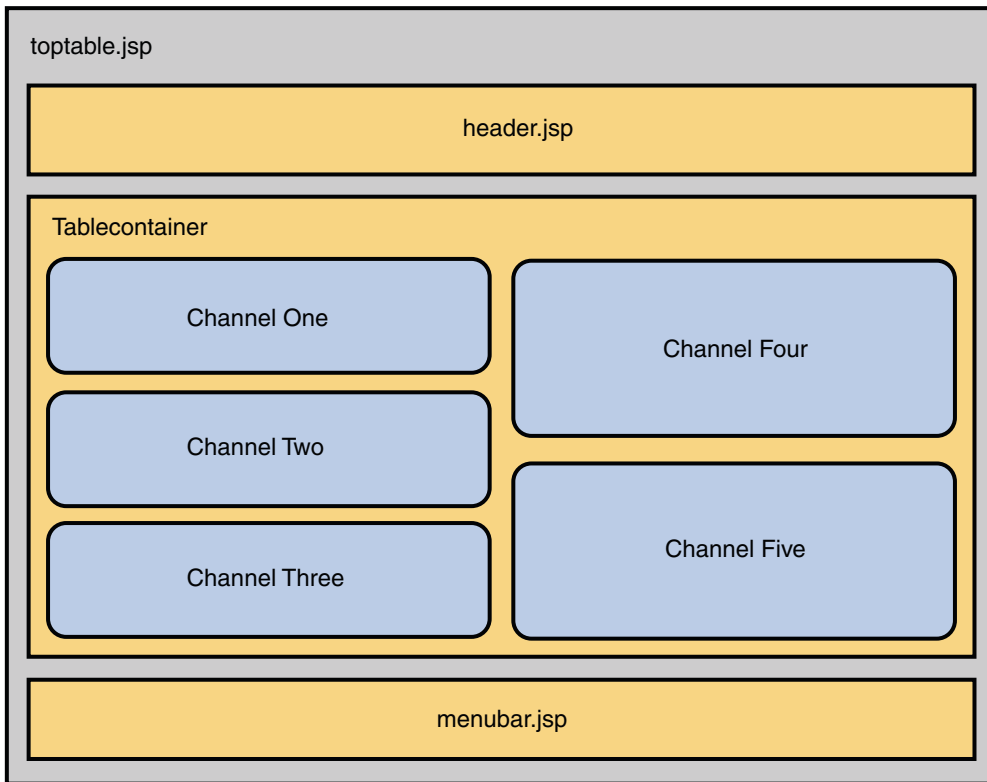


FIGURE 37–4 JSPTableContainer Architecture

## JSP Files Used by JSPTableContainer

The Portal Server 7.1 uses JSP files for a channel's presentation layer. JSPTableContainer references one main JSP, `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.



# Frame-Based Desktop

---

This chapter contains the following sections:

- “Sample Desktop” on page 285
- “FrameTabContainer Architecture” on page 288
- “JSP Files Used by FrameTabContainer” on page 289

The FrameTabContainer provides a frame-based table Desktop.

## Sample Desktop

### Default Layout

By default, the sample portal Desktop based on the FrameTabContainer provides a frame-based table Desktop consisting of two frames (see Figure 39–1), My Front Page and Samples, with the following channels:

- In the My Front Page frame: My Bookmarks, My Applications, Sample JSP, and XML Test
- In the Samples frame: URL Scraper, Notes, and Preconfigured Web Service

FIGURE 38-1 Sample Desktop Based on FrameTabContainer

## Default Actions

The sample FrameTabContainer channel, by default, includes:

- Banner links to return to the Desktop Home page, Sections (`action=edit&provider=FrameTabContainer`) that allow users to rename or select the start page, delete a page, and also create a new page, theme (`action=edit&provider=JSPPresetThemeContainer`) that allows users to set the color scheme and font type for the Desktop, Help (`../docs/locale/desktop/helppage.htm`) to display the Desktop sample online help, and Log Out (`action=logout`) that logs the user out from the Desktop.

The Sections link on the Content page displays the Current Page Settings Edit page where you can make changes. Here, Start page allows the user to set the starting page, Rename allows the user to rename the page, and Delete allows the user to delete a page from the Desktop.

- 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 Content link

(`action=edit&provider=JSPContentContainer&container=MyFrontPageFramePanelContainer`) allows the user to edit the content for this particular page on the Content page and the

Layout link

(`action=edit&provider=JSPLayoutContainer&container=MyFrontPageTabPanelContainer&se`) allows the user to edit the layout of the channels for this particular page on the Layout page.

## Default Display Profile Settings

The provider responsible for generating `FrameTabContainer` channel is `JSPTabContainerProvider`. 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. The properties that make up `FrameTabContainer` work as follows:

<code>contentPage</code>	Set to <code>frametab.jsp</code> . This draws the Content Page for the frame container.
<code>editPage</code>	Set to <code>frametabedit.jsp</code> . This displays the Edit page for the frame container where new pages can be added, or existing pages removed or renamed.
<code>startTab</code>	Sets the page that opens first on the Desktop as <code>MyFrontPageFramePanelContainer</code> .
<code>maxTabs</code>	Allows four pages to be created. As, by default, there are two pages, two more can be added.
<code>makeTabProvider</code>	Specifies <code>JSPFrameCustomTableContainerProvider</code> as the provider to create a new page on the Desktop.
<code>channelNumber</code>	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 <code>MyFrontPageFramePanelContainer</code> in <code>FrameTabContainer</code> , the new page channel name would be <code>FrameTabContainer/MyFrontPageFramePanelContainer1</code> . (The new page name is actually the <code>channelName</code> property in the display profile plus the value of <code>channelNumber</code> property. The <code>channelNumber</code> property value is incremented by one each time a new page is created.)
<code>contentChannel</code>	Specifies <code>JSPContentContainer</code> as the content channel that provides the Content page displaying channels to add to a user-created page.
<code>TabProperties</code>	This collection has entries for each of the four containers that are available in the <code>FrameTabContainer</code> .
<code>Available</code>	This list describes all available channels for this container. The available channels are displayed in the content preference page for users to select from.

Selected

This list describes selected channels for this container. Only selected channels shows up in the Desktop.

## FrameTabContainer Architecture

Figure 39–2 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.

FrameTabContainer is made up of two sub-containers, `MyFrontPageFramePanelContainer` and `SamplesFramePanelContainer`, as represented by Page 1 (`tablecontainer`) in the figure.

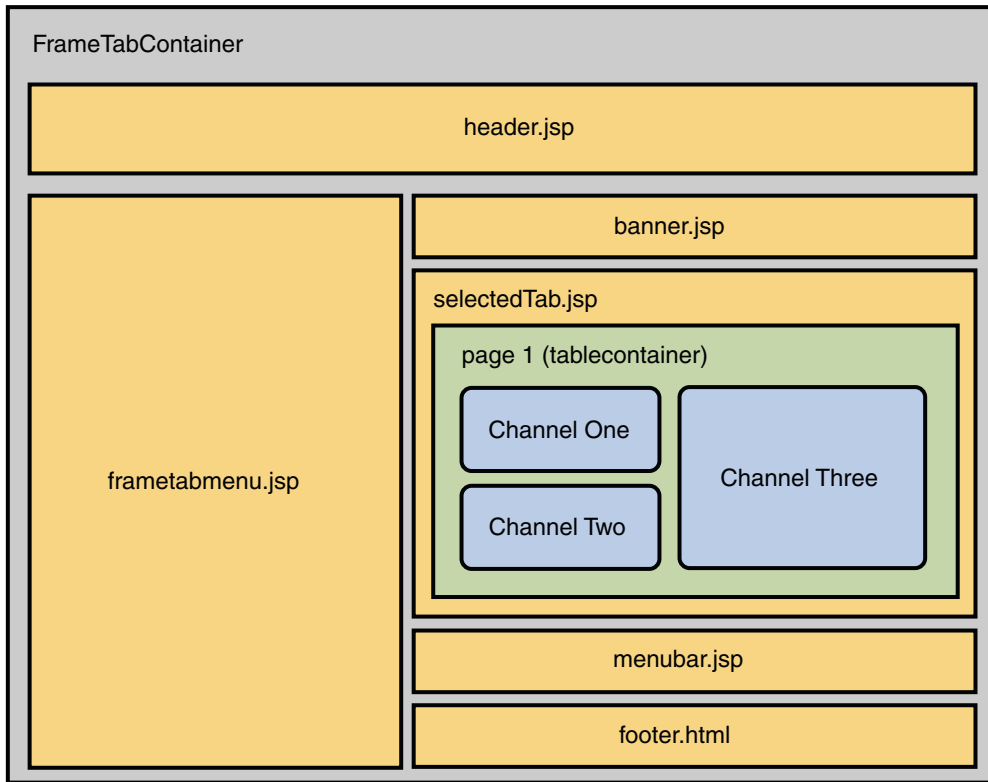


FIGURE 38–2 FrameTabContainer Architecture



## JSP Files Used by FrameTabContainer

The Portal Server 7.1 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.

The Edit page is where you can add, remove, and rename pages. The `frametabedit.jsp` is used to display this page.



## Internally Used 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)

Sun Java System Portal Server 7.1 software uses other container providers internally to perform such tasks as creating new tabs and edit containers.

## Internally Used Containers

This section lists the internally used containers in Portal Server 7.1 and provides their description.

TABLE 39-1 Internally Used Containers

Container Name	Description
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.

TABLE 39-1 Internally Used Containers (Continued)

Container Name	Description
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.
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 NewPage page on FrameTabContainer.
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 NewPage page on FrameTabContainer.
PredefinedToolsTemplatePanelContainerProvider	PredefinedToolsTemplatePanelContainerProvider is the provider for the Tools tab on TemplateTabContainer.
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.
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.
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.
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.

## Global Themes

---

This chapter contains the following sections:

- “What is a Theme?” on page 293
- “GlobalThemes Display Profile Definition” on page 293
- “Theme Properties” on page 294
- “Glossary of Terms” on page 296

### What is a Theme?

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 Java System Portal Server 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.

### GlobalThemes Display Profile Definition

The display profile document, *PortalServer-base/samples/desktop/dp-org.xml* file, contains the XML fragment for the eight default themes. See this file for the definition of these themes. See the Sun Java System Portal Server 7.1 Developer Sample Guide for more information on customizing the Global Themes.

## Theme Properties

The following is a list of theme properties that can be defined, modified, and/or customized in the display profile document. Please refer the [“Glossary of Terms” on page 296](#) for more detailed description for the theme properties.

activeBulletImage	activeBulletGraphics
inactiveBulletImage	inactiveBulletGraphics
brandImage	logo image
brandImage2	Product name image
brandImageBgColor	header logo bg color
brandImage2BgColor	header product name bg color
brandBgColor	header link box bg color
headerBgColor	header bg color and footer bg color
headerFontColor	header font color
	footer font color
headerText	header font size
	footer font size
	header font face
	footer font face
tabNotchImage	tabNotch image
titleText	selected tab font face
	selected tab font size
	unselected tab font face
	unselected tab font size
	channel title font face
	channel title font size
titleFontColor	selected tab font color
	channel title font color
fontSize	channel font size

---

	channel link font size
fontFace	channel font face
	channel link font face
titleBarColor	selected tab bg color
	channel title bar bg color
	content/layout bar color
	channel border color
	page piping color (bottom)
	button bg color
borderColor	border color
tabColor	unselected tab bg color
	secondary channel title bar color
tabFontColor	unselected tab font color
bgColor	channel bg color
fontColor	channel font color
borderWidth	channel border width
tableBgColor	table bg color
	page piping, top
channelHighlightColor	highlight color for channel content (as seen in the Placida theme JSP channel)
linkSeparatorColor	link separator color (in the toolbar, between Content and Layout)
	footer link separator color
channelLinkColor	channel link color
contentLayoutLinkColor	content/layout link color
contentLayoutText	content/layout link font size
	content/layout link font face
brandImageWidth	brand image width

previewImage	preview image (on preset themes page)
removeImage	remove image (for the channel title bar)
detachImage	detach image (for the channel title bar)
helpImage	help image (for the channel title bar)
editImage	edit image (for the channel title bar)
minimizeImage	minimize image (for the channel title bar)
maximizeImage	maximize image (for the channel title bar)
normalizeImage	normalize image (in the maximized channel)
attachImage	attach image (in the popup window)

## Glossary of Terms

The following table provides description of where the theme attributes are actually used in the desktop.

TABLE 40-1 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



TABLE 40-1 Glossary of Terms *(Continued)*

Term	Description
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



PART VII

Search Engine Robot

- Chapter 41
- Chapter 42
- Chapter 43
- Chapter 44
- Chapter 45
- Chapter 46
- Chapter 47
- Chapter 48
- Chapter 49
- Chapter 50
- Chapter 51



# Overview of Search Engine Robot

---

A Search Engine robot is an agent that identifies and reports on resources in its domains. It does so by using two kinds of filters: an enumerator filter and a generator filter.

The enumerator filter locates resources by using network protocols. It tests each resource, and, if it meets the proper criteria, it is enumerated. For example, the enumerator filter can extract hypertext links from an HTML file and use the links to find additional resources.

The generator filter tests each resource to determine if a resource description (RD) should be created. If the resource passes the test, the generator creates an RD which is stored in the Search Engine database.

Figure 42–1 illustrates how the Search Engine robot works. In Figure 42-1, the robot examines URLs and their associated network resources. Each resource is tested by both the enumerator and the generator. If the resource passes the enumeration test, the robot checks it for additional URLs. If the resource passes the generator test, the robot generates a resource description that is stored in the Search Engine database.

## Overview

The Robot Application Functions (RAFs) in the `filter.conf` file can be used to create and modify filter definitions. The file `filter.conf` is located in the `/var/opt/SUNWportal/searchservers/instanceName/config` directory.

The following figure shows how the robot works.

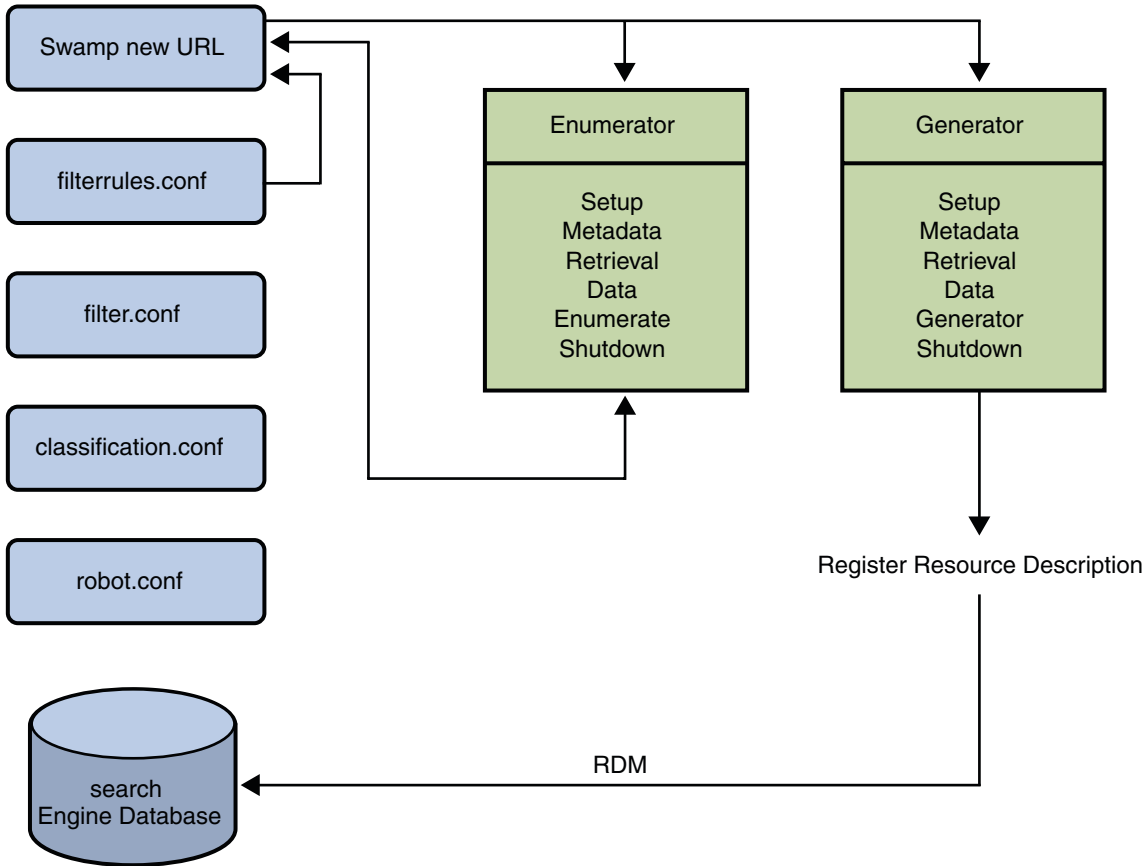


FIGURE 41-1 How the Robot Works

The `filter.conf` file contains definitions for the enumeration and generation filters. Each of these filters invokes a set of rules which are stored in the `filterrules.conf` file. The filter definitions contain instructions that are specific to each filter while the filter rules contain the rules used by both filters.

To understand how filter rules are defined, examine the `filterrules.conf` file. Note that you typically need not manually edit this file since you can create filter rules from the administration console.

For an example of filter definitions, examine the `filter.conf` file. Edit the `filter.conf` file only to modify the filters in a way that is not accommodated in the administration console, such as instructing the robot to enumerate some resources without generating resources for them.

## Process Parameters

---

The `robot.conf` file defines many options for the robot, including pointing the robot to the appropriate filters in `filter.conf` file. For backwards-compatibility with older versions, the `robot.conf` file can also contain the seed URLs.

The administration console is used to edit the file `robot.conf`. Because you can set most parameters by using the administration console, you typically do not need to edit the `robot.conf` file. However, advanced users might manually edit this file in order to set parameters that cannot be set through the administration console.

This chapter lists the user-modifiable parameters in the `robot.conf` file. The first column of the table lists the parameter, the second column provides a description of the parameter, and the third column provides an example.

### User Modifiable Parameters in `robot.conf` File

TABLE 42-1 User Modifiable Parameters in `robot.conf` File

Parameter	Description	Example
<code>auto-proxy</code>	Specifies the proxy setting for the robot. It can be a proxy server or a JavaScript file for automatically configuring the proxy.	<code>auto-proxy="http://proxy_server/proxy.pac"</code>
<code>bindir</code>	Specifies whether the robot will add a bind directory to the PATH environment. This is an extra PATH for users to run an external program in a robot, such as those specified by <code>cmd-hook</code> parameter.	<code>bindir=path</code>

TABLE 42-1 User Modifiable Parameters in robot.conf File (Continued)

Parameter	Description	Example
cmd-hook	Specifies an external completion script to run after the robot completes one run. This must be a full path to the command name. The robot will execute this script from the /var/opt/SUNWportal/ directory. There is no default.  There must be at least one RD registered for the command to run.	cmd-hook="command-string"
command-port	Specifies the socket that the robot listens to in order to accept commands from other programs, such as the Administration Interface or robot control panels.  For security reasons, the robot can accept commands only from the local host unless remote-access is set to yes.	command-port=port_number
connect-timeout	Specifies the maximum time allowed for a network to respond to a connection request. The default is 120 seconds.	command-timeout=seconds
convert-timeout	Specifies the maximum time allowed for document conversion. The default is 600 seconds.	convert-timeout=seconds
depth	Specifies the number of links from the seed URLs (also referred to as starting point) that the robot will examine. This parameter sets the default value for any seed URLs that do not specify a depth. The default is 10.  A value of negative one (depth=-1) indicates that the link depth is infinite.	depth=integer
email	Specifies the email address of the person who runs the robot.  The email address is sent with the user-agent in the HTTP request header, so that Web managers can contact the people who run robots at their sites. The default is user@domain.	email=user@hostname
enable-ip	Generates an IP address for the URL for each RD that is created. The default is true.	enable-ip=[true   yes   false   no]



TABLE 42-1 User Modifiable Parameters in robot.conf File (Continued)

Parameter	Description	Example
enable-rdm-probe	Determines if the server supports RDM, the robot decides whether to query each server it encounters by using this parameter. If the server supports RDM, the robot will not attempt to enumerate the server's resources, since that server is able to act as its own resource description server. The default is false.	enable-rdm-probe= [true   false   yes   no]
enable-robots-txt	Determines if the robot should check the robots.txt file at each site it visits, if available. The default is yes.	enable-robots-txt= [true   false   yes   no]
engine-concurrent	Specifies the number of pre-created threads for the robot to use. The default is 10.  This parameter cannot be set interactively through the administration console.	engine-concurrent=[1..100]
enumeration-filter	Specifies the enumeration filter that is used by the robot to determine if a resource should be enumerated. The value must be the name of a filter defined in the file filter.conf. The default is enumeration-default.  This parameter cannot be set interactively through the administration console.	enumeration-filter= enumfiltername
generation-filter	Specifies the number of minutes that the robot should collect RDs before batching them for the Search Engine.  If you do not specify this parameter, it is set to 256 minutes.	generation-filter=genfiltername
index-after-ngenerated	Specifies the number of minutes that the robot should collect RDs before batching them for the Search Engine.  If you do not specify this parameter, it is set to 256 minutes.	index-after-ngenerated=30

TABLE 42-1 User Modifiable Parameters in robot.conf File (Continued)

Parameter	Description	Example
loglevel	<p>Specifies the levels of logging. The loglevel values are as follows:</p> <ul style="list-style-type: none"> <li>■ Level 0: log nothing but serious errors</li> <li>■ Level 1: also log RD generation (default)</li> <li>■ Level 2: also log retrieval activity</li> <li>■ Level 3: also log filtering activity</li> <li>■ Level 4: also log spawning activity</li> <li>■ Level 5: also log retrieval progress</li> </ul> <p>The default value is 1.</p>	loglevel=[0...100]
max-connections	<p>Specifies the maximum number of concurrent retrievals that a robot can make. The default is 8.</p>	max-connections=[1..100]
max-filesize-kb	<p>Specifies the maximum file size in kilobytes for files retrieved by the robot. The default is 10240.</p>	max-filesize-kb=1024
max-memory-per-url / max-memory	<p>Specifies the maximum memory in bytes used by each URL. If the URL needs more memory, the RD is saved to disk. The default is 64000.</p> <p>This parameter cannot be set interactively through the administration console.</p>	max-memory-per-url=n_bytes
max-working	<p>Specifies the size of the robot working set, which is the maximum number of URLs the robot can work on at one time.</p> <p>This parameter cannot be set interactively through the administration console.</p>	max-working=1024
onCompletion	<p>Determines what the robot does after it has completed a run. The robot can either go into idle mode, loop back and start again, or quit. The default is idle.</p> <p>This parameter works with the cmd-hook parameter. When the robot is done, it will do the action of onCompletion and then run the cmd-hook program.</p>	OnCompletion=[idle   loop   quit]
password	<p>Specifies the password is used for httpd authentication and ftp connection.</p>	password=string

TABLE 42-1 User Modifiable Parameters in robot.conf File (Continued)

Parameter	Description	Example
referer	Specifies the parameter sent in the HTTP request if it is set to identify the robot as the referer when accessing Web pages	referer=string
remote-access	This parameter determines if the robot can accept commands from remote hosts. The default is false.	remote-access=[true   false   yes   no]
robot-state-dir	Specifies the directory where the robot saves its state. In this working directory, the robot can record the number of collected RDs and so on.	robot-state-dir="/var/opt/SUNWportal/ins tance/portal/robot"
server-delay	Specifies the time period between two visits to the same web site, thus preventing the robot from accessing the same site too frequently.	server-delay=delay_in_seconds
site-max-connections	Indicates the maximum number of concurrent connections that a robot can make to any one site. The default is 2.	site-max-connections=[1..100]
smart-host-heuristics	Enables the robot to change sites that are rotating their DNS canonical host names. For example, www123.siroe.com is changed to www.siroe.com. The default is false.	smart-host-heuristics=[true   false]
tmpdir	Specifies a place for the robot to create temporary files. Use this value to set the environment variable TMPDIR.	tmpdir=path
user-agent	Specifies the parameter sent with the email address in the http-request to the server.	user-agent=iPlanetRobot/4.0
username	Specifies the user name of the user who runs the robot and is used for httpd authentication and ftp connection. The default is anonymous.	username=string

The most important parameters are `enumeration-filter` and `generation-filter`, which determine the filters the robot uses for enumeration and generation. The default values for these are `enumeration-default` and `generation-default`, which are the names of the filters provided by default in the `filter.conf` file.

All filters must be defined in the file `filter.conf` file. If you define your own filters in `filter.conf` file, you must add any necessary parameters to `robot.conf` file.

For example, if you define a new enumeration filter named my-enumerator, you would add the parameter to robot.conf file:

```
enumeration-filter=my-enumerator
```

# The Filtering Process

---

This chapter contains the following sections

- “Overview” on page 309
- “Stages in the Filter Process” on page 310
- “Filter Syntax” on page 311
- “Filter Directives” on page 312
- “Writing or Modifying a Filter” on page 312

## Overview

The robot uses filters to determine which resources to process and how to process them. When the robot discovers references to resources as well as the resources themselves, it applies filters to each resource in order to enumerate it and to determine whether or not to generate a resource description to store in the Search Engine database.

The robot examines one or more seed URLs, applies the filters, and then applies the filters to the URLs spawned by enumerating the seed URLs, and so on. The seed URLs are defined in the `filterrules.conf` file.

A filter performs any required initialization operations and applies comparison tests to the current resource. The goal of each test is to either allow or deny the resource. A filter also has a shutdown phase during which it performs any required cleanup operations.

If a resource is allowed, that means that it is allowed to continue passage through the filter. If a resource is denied, then the resource is rejected. No further action is taken by the filter for resources that are denied. If a resource is not denied, the robot will eventually enumerate it, attempting to discover further resources. The generator might also create a resource description for it.

These operations are not necessarily linked. Some resources result in enumeration; others result in RD generation. Many resources result in both enumeration and RD generation. For example,

if the resource is an FTP directory, the resource typically will not have an RD generated for it. However, the robot might enumerate the individual files in the FTP directory. An HTML document that contains links to other documents can receive an RD and can lead to enumeration of the linked documents as well.

## Stages in the Filter Process

Both enumerator and generator filters have five phases in the filtering process. They both have four common phases: Setup, Metadata, Data, and Shutdown. If the resource makes it past the Data phase, it is either in the Enumerate or Generate phase, depending on whether the filter is an enumerator or a generator.

The phases are as follows:

### Setup

Performs initialization operations. Occurs only once in the life of the robot.

### Metadata

Filters the resource based on metadata that is available about the resource. Metadata filtering occurs once per resource before the resource is retrieved over the network. The table below lists the common metadata types and their description.

TABLE 43-1 Common Metadata Types

Metadata	Description	Example
Complete URL	The location of a resource	http://home.siroe.com/
Protocol	The access portion of the URL	http, ftp, file
Host	The address portion of the URL	www.siroe.com
IP address	Numeric version of the host	198.95.249.6
PATH	The path portion of the URL	/index.html
Depth	Number of links from the seed URL	5

### Data

Filters the resource based on its data. Data filtering is done once per resource after it is retrieved over the network. Data that can be used for filtering include:

- content-type
- content-length
- content-encoding

- content-charset
  - last-modified
  - expires
- Enumerate

Enumerates the current resource in order to determine if it points to other resources to be examined.

Generate

Generates a resource description (RD) for the resource and saves it in the Search Engine database.

Shutdown

Performs any needed termination operations. Occurs once in the life of the robot.

## Filter Syntax

The `filter.conf` file contains definitions for enumeration and generation filters. This file can contain multiple filters for both enumeration and generation. Note that the robot can determine which filters to use because they are specified by the `enumeration-filter` and `generation-filter` parameters in the `robot.conf` file.

Filter definitions have a well-defined structure: a header, a body, and an end. The header identifies the beginning of the filter and declares its name; for example:

```
<Filter name="myFilter">
```

The body consists of a series of filter directives that define the filter's behavior during setup, testing, enumeration or generation, and shutdown. Each directive specifies a function, and if applicable, parameters for the function.

The end is marked by `</Filter>`.

The following example shows a filter named `enumeration1`.

### EXAMPLE 43-1 Enumeration File Syntax

```
<Filter name="enumeration1">
Setup fn=filterrules-setup config=./config/filterrules.conf
# Process the rules
MetaData fn=filterrules-process
# Filter by type and process rules again
Data fn=assign-source dst=type src=content-type
```

**EXAMPLE 43-1** Enumeration File Syntax (Continued)

```
Data fn=filterrules-process
# Perform the enumeration on HTML only
Enumerate enable=true fn=enumerate-urls max=1024 type=text/html
# Cleanup
Shutdown fn=filterrules-shutdown
</Filter>
```

## Filter Directives

Filter directives use Robot Application Functions (RAFTs) to perform operations. Their use and flow of execution is similar to that of NSAPI directives and Server Application Functions (SAFs) in the file obj.conf. Like NSAPI and SAF, data are stored and transferred using parameter blocks, also called pblocks.

There are six robot directives, or RAF classes, corresponding to the filtering phases and operations listed below. See [“Stages in the Filter Process” on page 310](#) for more information on these phases.

- Setup
- Metadata
- Data
- Enumerate
- Generate
- Shutdown

Each directive has its own robot application functions. For example, use filtering functions with the Metadata and Data directives, enumeration functions with the Enumerate directive, generation functions with the Generate directive, and so on.

The built-in robot application functions and instructions for writing your own robot application functions are explained in the Sun Java System Portal Server 7.1 Developer's Guide.

## Writing or Modifying a Filter

In most cases, you should not need to write filters from scratch. You can create most of your filters using the administration console. You can then modify the filter.conf and filterrules.conf files to make any desired changes. These files reside in the directory /var/opt/SUNWportal/searchservers/search1/config.

However, if you want to create a more complex set of parameters, you will need to edit the configuration files used by the robot.

Follow these points when writing or modifying a filter:



- The order of execution of directives (especially the available information at each phase)
- The order of rules

For a discussion of the parameters you can modify in the `robot.conf` file, the robot application functions that you can use in the `filter.conf` file, and how to create your own robot application functions, see the Sun Java System Portal Server 7.1 Developer's Guide.



# Robot Application Functions - Sources and Destinations

---

This chapter contains the following sections:

- “Introduction” on page 315
- “Setup Stage” on page 315
- “MetaData Filtering Stage” on page 316
- “Data Stage” on page 316
- “Enumeration, Generation, and Shutdown Stages” on page 317

## Introduction

Most of the Robot Application Functions (RAFTs) require sources of information and generate data that goes to destinations. The sources are defined within the robot itself and are not necessarily related to the fields in the resource description it ultimately generates. Destinations, on the other hand, are generally the names of fields in the resource description, as defined by the resource description server’s schema.

For details on using the administration console to determine the database schema, see Sun Java System Portal Server 7.1 Administration Guide.

The following sections describe the different stages of the filtering process, and the sources available at those stages.

## Setup Stage

At the Setup stage, the filter is set up and cannot yet get information about the resource’s URL or content.

## MetaData Filtering Stage

At the MetaData stage, the robot encounters a URL for a resource, but it has not downloaded the resource's content, thus information is available about the URL as well as data that is derived from other sources such as the `filter.conf` file. At this stage, however, information is not available about the content of the resource.

The table below lists the sources available in the RAFs at the MetaData phase and their description.

TABLE 44-1 Sources Available to the RAFs at the MetaData Phase

Source	Description	Example
csid	Catalog Server ID	x-catalog//budgie.siroe.com:8086/alexandria
depth	Number of links traversed from starting point	10
enumeration filter	Name of Enumeration filter	enumeration1
generation filter	Name of Generation filter	generation1
host	Host portion of URL	home.siroe.com
IP	Numeric version of host	198.95.249.6
protocol	Access portion of the URL	http, https, ftp, file
path	Path portion of the URL	/, /index.html, /documents/listing.html
URL	Complete URL	http://developer.siroe.com/docs/manuals/

## Data Stage

At the Data stage, the robot has downloaded the content of the resource at the URL, and can access data about the content, such as the description, the author, and so on.

If the resource is an HTML file, the Robot parses the `<META>` tags in the HTML headers. Consequently, any data contained in `<META>` tags is available at the Data stage.

During the data phase, the following sources, shown in the following table are available to RAFs, in addition to those available during the MetaData phase.

TABLE 44-2 Sources Available to the RAFs at the Data Phase

Source	Description	Example
content-charset	Character set used by the resource	
content-encoding	Any form of encoding	
content-length	Size of the resource in bytes	
content-type	MIME type of the resource	text/html, image/jpeg
expires	Date the resource itself expires	
last-modified	Date the resource was last modified	
data in <META> tags	Any data that is provided in <META> tags in the header of HTML resources	Author Description Keywords

## Enumeration, Generation, and Shutdown Stages

At the Enumeration and Generation stages, the same data sources are available as the Data stage.

At the Shutdown stage, the filter completes its filtering and is shut down. Although functions written for this stage can use the same data sources as those available at the Data stage, the shutdown functions typically restrict their operations to shutdown and cleanup activities.



## Robot Application Functions - Enable Parameter

---

Each function can have an `enable` parameter. The values can be `true`, `false`, `on`, or `off`. The administration console uses these parameters to turn certain directives on or off.

The following example enables enumeration for `text/html` and disables enumeration for `text/plain`:

### Example

```
# Perform the enumeration on HTML only
Enumerate enable=true fn=enumerate-urls max=1024 type=text/html
Enumerate enable=false fn=enumerate-urls-from-text max=1024 type=text/plain
```

Adding an `enable=false` parameter or an `enable=off` parameter has the same effect as commenting the line. Because the administration console does not write comments, it writes an `enable` parameter instead.





# Robot Application Functions - Setup Functions

---

This section describes the functions that are used during the setup phase by both enumeration and generation filters. The following functions are described:

- “[filterrules-setup](#)” on page 321
- “[setup-regex-cache](#)” on page 322
- “[setup-type-by-extension](#)” on page 322

## filterrules-setup

When you use the `filterrules-setup` function, `logtype` is the type of log file to use. The value can be `verbose`, `normal`, or `terse`.

## Parameters

The list of parameters used with the `filterrules-setup` function and their description are:

`config` Path name to the file containing the filter rules to be used by this filter.

## Example

```
Setup fn=filterrules-setup config=./config/filterrules.conf logtype=normal
```

## setup-regex-cache

The `setup-regex-cache` function initializes the cache size for the `filter-by-regex` and `generate-by-regex` functions. Use this function to specify a number other than the default of 32.

### Parameters

The parameter used with the `setup-regex-cache` function and its description is:

`cache-size`    Maximum number of compiled regular expressions to be kept in the regex cache.

### Example

```
Setup fn=setup-regex-cache cache-size=28
```

## setup-type-by-extension

The `setup-type-by-extension` function configures the filter to recognize file name extensions. It must be called before the `assign-type-by-extension` function can be used. The file specified as a parameter must contain mappings between standard MIME content types and file extension strings.

### Parameters

The parameter used with the `setup-type-by-extension` function and its description is:

`file`    Name of the MIME types configuration file.

### Example

```
Setup fn=setup-type-by-extension file=./config/mime.types
```

# Robot Application Functions - Filtering Functions

---

This chapter contains the following sections:

- “Introduction” on page 323
- “filter-by-exact” on page 323
- “filter-by-max” on page 324
- “filter-by-md5” on page 325
- “filter-by-prefix” on page 325
- “filter-by-regex” on page 326
- “filterrules-process” on page 326

## Introduction

The functions discussed in this chapter operate at the Metadata and Data stages to allow or deny resources based on specific criteria specified by the function and its parameters.

These functions can be used in both Enumeration and Generation filters in the `filter.conf` file.

Each “filter-by” function performs a comparison, then either allows or denies the resource. Allowing the resource means that processing continues to the next filtering step. Denying the resource means that processing should stop, because the resource does not meet the criteria for further enumeration or generation.

## filter-by-exact

The filter-by-exact function allows or denies the resource if the allow/deny string matches the source of information exactly. The keyword `all` matches any string.

## Parameters

The parameters used with the `filter-by-exact` function and their description are:

`src`                Source of information.  
`allow/deny`        Contains a string.

## Example

The following example filters out all resources whose `content-type` is `text/plain`. It allows all other resources to proceed:

```
Data fn=filter-by-exact src=type deny=text/plain
```

## filter-by-max

The `filter-by-max` function allows the resource if the specified information source is less than or equal to the given value. It denies the resource if the information source is greater than the specified value.

This function can be called no more than once per filter.

## Parameters

The parameters used with the `filter-by-max` function and their description are:

`src`                Source of information. It must be one of the following: `hosts`, `objects`, or `depth`.  
`value`             Specifies a value for comparison.

## Example

This example allows resources whose `content-length` is less than 1024 K:

```
MetaData fn=filter-by-max src=content-length value=1024
```

## filter-by-md5

The `filter-by-md5` function only allows the first resource with a given MD5 checksum value. If the current resource's MD5 has been seen in an earlier resource by this robot, the current resource is denied. As a result, duplication of identical resources or single resources with multiple URLs is prevented.

You can only call this function at the Data stage or later. It can be called no more than once per filter. The filter must invoke the `generate-md5` function to generate an MD5 checksum before invoking `filter-by-md5` function.

### Parameters

none

### Example

The following example shows the typical method of handling MD5 checksums by first generating the checksum and then filtering based on it:

```
Data fn=generate-md5
Data fn=filter-by-md5
```

## filter-by-prefix

The `filter-by-prefix` function allows or denies the resource if the given information source begins with the specified prefix string. The resource doesn't have to match completely. The keyword `all` matches any string.

### Parameters

The parameters used with the `filter-by-prefix` function and their description are:

<code>src</code>	Source of information.
<code>allow/deny</code>	Contains a string for prefix comparison.

## Example

The following example allows resources whose content-type is any kind of text, including text/html and text/plain:

```
MetaData fn=filter-by-prefix src=type allow=text
```

## filter-by-regex

The `filter-by-regex` function supports regular expression pattern matching. It allows resources that match the given regular expression. The supported regular expression syntax is defined by the POSIX.1 specification. The regular expression `\\\\\\*` matches anything.

## Parameters

The parameters used with the `filter-by-regex` function and their description are:

<code>src</code>	Source of information.
<code>allow/deny</code>	Contains a string for prefix comparison.

## Example

The following example denies all resources from sites in the government domain:

```
MetaData fn=filter-by-regex src=host deny=\\\\\\*.gov
```

## filterrules-process

The `filterrules-process` function handles in the rules in the `filterrules.conf` file.

## Parameters

`none`

## Example

```
MetaData fn=filterrules-process
```

# Robot Application Functions - Filtering Support Functions

---

This chapter contains the following sections:

- “Introduction” on page 327
- “assign-source” on page 327 and “assign-type-by-extension” on page 328
- “clear-source” on page 328
- “convert-to-html” on page 329
- “copy-attribute” on page 329
- “generate-by-exact” on page 330, “generate-by-prefix” on page 331, and “generate-by-regex” on page 331
- “generate-md5” on page 332
- “generate-rd-expires” on page 332 and “generate-rd-last-modified” on page 333
- “rename-attribute” on page 333

## Introduction

The functions discussed in this chapter are used during filtering to manipulate or generate information on the resource. The robot can then process the resource by calling filtering functions. These functions can be used in Enumeration and Generation filters in the `filter.conf` file.

## assign-source

The `assign-source` function assigns a new value to a given information source. This permits editing during the filtering process. The function can assign an explicit new value, or it can copy a value from another information source.

## Parameters

The parameters used with the `assign-source` function and their description are:

`dst` Name of the source whose value is to be changed.

`value` Specifies an explicit value.

`src` Information source to copy to `dst`

You must specify either a `value` parameter or a `src` parameter, but not both.

## Example

```
Data fn=assign-source dst=type src=content-type
```

## assign-type-by-extension

The `assign-type-by-extension` function uses the resource's file name to determine its type and assigns this type to the resource for further processing.

The `setup-type-by-extension` function must be called during setup before `assign-type-by-extension` function can be used.

## Parameters

The parameter used with the `assign-type-by-extension` function and its description is:

`src` Source of file name to compare. If you do not specify a source, the default is the resource's path.

## Example

```
MetaData fn=assign-type-by-extclear-source
```

## clear-source

The `clear-source` function deletes the specified data source. You typically do not need to perform this function. You can create or replace a source by using the `assign-source` function.

## Parameters

The parameter used with the `clear-source` function and its description is:



`src` Name of source to delete.

## Example

The following example deletes the path source:

```
MetaData fn=clear-source src=path
```

## convert-to-html

The `convert-to-html` function converts the current resource into an HTML file for further processing, if its type matches a specified MIME type. The conversion filter automatically detects the type of the file it is converting.

## Parameters

The parameter used with the `convert-to-html` function and its description is:

`type` MIME type from which to convert.

## Example

The following sequence of function calls causes the filter to convert all Adobe Acrobat PDF files, Microsoft RTF files, and FrameMaker MIF files to HTML, as well as any files whose type was not specified by the server that delivered it.

```
Data fn=convert-to-html type=application/pdf  
Data fn=convert-to-html type=application/rtf  
Data fn=convert-to-html type=application/x-mif  
Data fn=convert-to-html type=unknown
```

## copy-attribute

The `copy-attribute` function copies the value from one field in the resource description into another.

## Parameters

The parameters used with the `copy-attribute` function and their description are:

<code>src</code>	Field in the resource description from which to copy.
<code>dst</code>	Item in the resource description into which to copy the source.
<code>truncate</code>	Maximum length of the source to copy.
<code>clean</code>	Boolean parameter indicating whether to fix truncated text (such as not leaving partial words). This parameter is false by default.

## Example

```
Generate fn=copy-attribute \<\  
src=partial-text dst=description truncate=200 clean=true
```

## generate-by-exact

The `generate-by-exact` function generates a source with a specified value, but only if an existing source exactly matches another value.

## Parameters

The parameters used with the `generate-by-exact` function and their description are:

<code>dst</code>	Name of source to generate.
<code>value</code>	Value to assign <code>dst</code> .
<code>src</code>	Source against which to match.

## Example

The following example sets the classification to Siroe if the host is `www.siroe.com`.

```
Generate fn="generate-by-exact" match="www.siroe.com:80" src="host"  
value="Siroe" dst="classification"
```

## generate-by-prefix

This `generate-by-prefix` function generates a source with a specified value, but only if the prefix of an existing source matches another value.

### Parameters

The parameters used with the `generate-by-prefix` function and their description are:

<code>dst</code>	Name of the source to generate.
<code>value</code>	Value to assign to <code>dst</code> .
<code>src</code>	Source against which to match.
<code>match</code>	Value to compare to <code>src</code> .

### Example

The following example sets the classification to Search if the protocol prefix is HTTP:

```
Generate fn="generate-by-prefix" match="http" src="protocol"
value="World Wide Web" dst="classification"
```

## generate-by-regex

The `generate-by-regex` function generates a source with a specified value, but only if an existing source matches a regular expression.

### Parameters

The parameters used with the `generate-by-regex` function and their description are:

<code>dst</code>	Name of the source to generate.
<code>value</code>	Value to assign to <code>dst</code> .
<code>src</code>	Source against which to match.
<code>match</code>	Regular expression string to compare to <code>src</code> .

## Example

The following example sets the classification to Siroe if the host name matches the regular expression `*.siroe.com`. For example, resources at both `developer.siroe.com` and `home.siroe.com` will be classified as Siroe:

```
Generate fn="generate-by-regex" match="\\.*.siroe.com"  
src="host" value="Siroe" dst="classification"
```

## generate-md5

The `generate-md5` function generates an MD5 checksum and adds it to the resource. You can then use the `filter-by-md5` function to deny resources with duplicate MD5 checksums.

## Parameters

none

## Example

```
Data fn=generate-md5
```

## generate-rd-expires

The `generate-rd-expires` function generates an expiration date and adds it to the specified source. The function uses metadata such as the HTTP header and HTML `<META>` tags to obtain any expiration data from the resource. If none exists, it generates an expiration date three months from the current date.

## Parameters

The parameter used with the `generate-rd-expires` function and its description is:

`dst`     Name of the source. If you omit it, it defaults to `rd-expires`.

## Example

```
Generate fn=generate-rd-expires
```

## generate-rd-last-modified

The `generate-rd-last-modified` function adds the current time to the specified source.

## Parameters

The parameter used with the `generate-rd-last-modified` function and its description is:

`dst` Name of the source. If you omit it, it defaults to `rd-last-modified`.

## Example

```
Generate fn=generate-last-modified
```

## rename-attribute

The `rename-attribute` function changes the name of a field in the resource description. It is most useful in cases where, for example, `extract-html-meta` copies information from a `<META>` tag into a field, and you want to change the name of the field.

## Parameters

The parameter used with the `generate-rd-last-modified` function and its description is:

`src` String containing a mapping from one name to another.

## Example

The following example renames an attribute from `author` to `author-name`:

```
Generate fn=rename-attribute src="author->author-name"
```



# Robot Application Functions - Enumeration Functions

---

This chapter contains the following sections

- [“Introduction” on page 335](#)
- [“enumerate-urls” on page 335](#)
- [“enumerate-urls-from-text” on page 336](#)

## Introduction

The functions discussed in this chapter operate at the Enumerate stage. These functions control if and how a robot gathers links from a given resource in order to use as starting points for further resource discovery.

### enumerate-urls

The `enumerate-urls` function scans the resource and enumerates all URLs found in hypertext links. The results are used to spawn further resource discovery. You can specify a content-type to restrict the kind of URLs enumerated.

### Parameters

The parameters used with the `enumerate-urls` function and their description are:

- |                   |   |
|-------------------|---|
| <code>max</code>  | The maximum number of URLs to spawn from a given resource. The default, if <code>max</code> is omitted, is 1024.  |
| <code>type</code> | Content-type that restricts enumeration to those URLs that have the specified content-type. <code>type</code> is an optional parameter. If omitted, it will enumerate all URLs. |

## Example

The following example enumerates HTML URLs only, up to a maximum of 1024:

```
Enumerate fn=enumerate-urls type=text/html
```

## enumerate-urls-from-text

The `enumerate-urls-from-text` function scans text resources, looking for strings matching this regular expression: `URL:.*`. It spawns robots to enumerate the URLs from these strings and generate further resource descriptions.

## Parameters

The parameter used with the `enumerate-urls-from-text` function and its description is:

`max`     The maximum number of URLs to spawn from a given resource. The default, if `max` is omitted, is 1024.

## Example

```
Enumerate fn=enumerate-urls-from-text
```



# Robot Application Functions - Generation Functions

---

This chapter contains the following functions:

- “Introduction” on page 337
- “extract-full-text” on page 337
- “extract-html-meta” on page 338, “extract-html-text” on page 338, and “extract-html-toc” on page 339
- “extract-source” on page 340
- “harvest-summarizer” on page 340

## Introduction

The following functions are used in the Generate stage of filtering. Generation functions can generate information that goes into a resource description. In general, they either extract information from the body of the resource itself or copy information from the resource’s metadata.

### extract-full-text

The `extract-full-text` function extracts the complete text of the resource and adds it to the resource description.

---

**Note** – The `extract-full-text` function should be used with caution, because it can significantly increase the size of the resource description, thus causing database bloat and overall negative impact on network bandwidth.

---

### Parameters

The parameters used with the `extract-full-text` function and their description are:

`truncate`     The maximum number of characters to extract from the resource.  
`dst`            Name of the schema item that will receive the full text.

## Example

Generate `fn=extract-full-text`

## extract-html-meta

The `extract-html-meta` function extracts any `<META>` or `<TITLE>` information from an HTML file and adds it to the resource description. A `content-type` may be specified to restrict the kind of URLs that are generated.

## Parameters

The parameters used with the `extract-html-meta` function and their description are:

`truncate`     The maximum number of bytes to extract.  
`type`          Optional parameter. If omitted, it will generate all URLs.

## Example

Generate `fn=extract-html-meta truncate=255 type=text/html`

## extract-html-text

The `extract-html-text` function extracts the first few characters of text from an HTML file, excluding the HTML tags, and adds the text to the resource description. This permits the first part of a document's text to be included in the RD. A `content-type` may be specified to restrict the kind of URLs that are generated.

## Parameters

The parameters used with the `extract-html-text` function and their description are:

`truncate`     The maximum number of bytes to extract.

<code>skip-headings</code>	Set to true to ignore any HTML headers that occur in the document.
<code>type</code>	Optional parameter. If omitted, it will generate all URLs.

## Example

Generate `fn=extract-html-text truncate=255 type=text/html skip-headings=true`

## extract-html-toc

The `extract-html-toc` function extracts the table-of-contents from the HTML headers and add it to the resource description.

## Parameters

The parameters used with the `extract-html-toc` function and their description are:

<code>truncate</code>	The maximum number of bytes to extract.
<code>level</code>	Maximum HTML header level to extract. This parameter controls the depth of the table of contents.

Robot HTML Summarizer does not generate description and partial text for some of the documents, such as `text/HTML`, `application/x-maker`, or `x-frame`. There are three causes for Robot not generating the description and partial text for the following:

- For HTML or text - Unclosed JavaScript tag. This is an error that you need to fix in the HTML page itself.
- Robot does not index the part of the HTML page that falls between `stopindex` and `startindex`.

For any file other than HTML or text, such as `application/x-maker`, or `x-frame`, Robot uses a third party Converter to convert them into HTML. Then, Robot indexes them. In some cases, the Converter might not able to generate the HTML or it may generate an empty HTML body. In this case, Sun will report to the third party for a fix or a patch to solve the issue.

## Example

Generate `fn=extract-html-toc truncate=255 level=3`

## extract-source

The `extract-source` function extracts the specified values from the given sources and adds them to the resource description.

### Parameters

The parameter used with the `extract-source` function and its description is:

`src` List of source names; you can use the `->` operator to define a new name for the RD attribute, for example, `type->content-type` would take the value of the source named `type` and save it in the RD under the attribute named `content-type`.

### Example

```
Generate fn=extract-source src="md5,depth,rd-expires,rd-last-modified"
```

## harvest-summarizer

The `harvest-summarizer` function runs a Harvest summarizer on the resource and adds the result to the resource description.

To run Harvest summarizers, you must have `$HARVEST_HOME/lib/gatherer` in your path before you run the robot.

### Parameters

The parameter used with the `harvest-summarizer` function and its description is:

`summarizer` Name of the summarizer program.

### Example

```
Generate fn=harvest-summarizer summarizer=HTML.sum
```

# Robot Application Functions - Shutdown Functions

---

This chapter contains the following sections:

- “Introduction” on page 341
- “`filterrules-shutdown`” on page 341

## Introduction

The function in this chapter can be used during the shutdown phase by both enumeration and generation functions.

## `filterrules-shutdown`

After the rules are run, the `filterrules-shutdown` function performs clean up and shutdown responsibilities.

## Parameters

none

## Example

```
Shutdown fn=filterrules-shutdown
```



PART VIII

Desktop Tag Library

- Chapter 52
- Chapter 53
- Chapter 54
- Chapter 55
- Chapter 56
- Chapter 57
- Chapter 58
- Chapter 59
- Chapter 60
- Chapter 61
- Chapter 62
- Chapter 63
- Chapter 64
- Chapter 65
- Chapter 66
- Chapter 67
- Chapter 68





# Overview of Desktop Tag Library

---

This chapter contains the following sections:

- “Introduction” on page 345
- “Desktop Tag Library Hierarchy” on page 350
- “Tag Library Descriptors” on page 350

## Introduction

Sun Java System Portal Server 7.1 uses two types of tags: JSP tags and Desktop template tags. JSP tags are used in the JSPs and template tags are used in the HTML pages of the Desktop.

## Types of Tags

The Portal Server 7.1 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.

## Desktop Template Tags

This section describes the tags used in the HTML templates of 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.

## Kinds of Tags Used in the Desktop Templates

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 Web Server instance,  <code>[surl:/desktop/imagesnothing.gif]-&gt;/var/opt/SUNWportal/https-servername/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-&gt;http://server:port/portlet/dt?action=logout</code>

## JavaServer Pages Tags

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.

## Purpose of Tag Library

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 tags in the JSP 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`).

#### Validator tags

These tags validate that the provider in context can legally use the tags in the TLD.

#### Normal tags

These tags serve as wrappers of `PAPI`, `ProviderContext`, `ContainerProviderContext`, and the specific containers in the Sun Java System Portal Server 7.1 software.

## Tag Attributes, Return Values, and Exceptions

### Attributes

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, `jsp:tl`).

---

## Return Values

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.

## Exceptions

The tag library provides five types of exceptions.

### Invalid Tag Sequence

Occurs when obtain tags are nested in an invalid sequence. See *Using the Desktop Tag Library in Your Application* for the sequencing of the obtain tags. The name of the first tag that violates the sequence is provided.

### 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.

### 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.

### 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.

### 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.

The many of the Java classes that support these tags reside in a JAR file `desktoptl.jar` in the *Web-Container-Instance/portal/web-apps/WEB-INF/lib* directory. The classes for the `jax.tld` and `jr.tld` tags reside in the `jsptl.jar` file.

## 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.

### Types of Tags

The `search.tld` file does not have any context setup or validator tags. All of the normal tags in the `search.tld` file have a body content of empty except `searchContext` and `search result`.

### Tag Attributes

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.

### 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

## 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 the 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`.

## Tag Library Descriptors

The Desktop tag library has the following Tag Library Descriptors (TLDs) in the `/var/opt/SUNWportal/portals/<portal_id>/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 ProviderContext interface.
<code>desktopContainerProviderContext.tld</code>	Contains tags that bring forward the functionality available from the ContainerProviderContext interface.
<code>desktopProviderContext.tld</code>	Contains tags to operate on providers that extend the ProviderContext.

---

<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.





## Context Setup Tags

---

The context setup 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.

This chapter 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.

### Context Setup Tags

TABLE 53-1 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

TABLE 53-1 Context Setup Tags (Continued)

Tag Name	Description	in TLD File	Attributes/Descriptions
obtainChannelFromContainer		desktopContainerProviderContext.tld	channel (required) - the name of the channel
obtainSelectedChannel		desktopSingle.tld	none
obtainTab		desktopTab.tld	tab (required) - the name of the tab
obtainTabByName	Gets the tab name.	desktopTab.tld	name (required) - the name of the tab

## Validator Tags

---

The validator 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.

### Validator Tags

The list of Validator tags and their description are:

<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> .



# desktop.tld Tags

---

This chapter lists the `desktop.tld` tags and their attributes. All of the normal tags in the `desktop.tld` file have a `bodycontent` of empty.

## desktop.tld Tags With Attributes

TABLE 55-1 desktop.tld Tags with Attributes

Tag Name	Description	Attributes
<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 false 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)
<code>getDescription</code>	Returns a string with the description for the channel.	<code>id</code> (optional) <code>scope</code> (optional) <code>silentException</code> (optional)

TABLE 55-1 desktop.tld Tags with Attributes (Continued)

Tag Name	Description	Attributes
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 ( <a href="http://server:port/portal/docs/en/desktop/usedesk.htm">http://server:port/portal/docs/en/desktop/usedesk.htm</a> ) or a relative path ( <a href="#">desktop/usedesk.htm</a> ). 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 <i>X</i> , you may choose not to fetch fresh content (and use a cached copy instead) if less than <i>X</i> 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 55-1 desktop.tld Tags with Attributes (Continued)

Tag Name	Description	Attributes
isPresentable	<p>Returns a Boolean that gives the presentable status for a channel. Returns true if the channel is presentable.</p> <p>Searches for the key HTML with the value true on the client data for the session's client type and returns true.</p> <p>If there is no such key, the method returns true if the session's client type is named HTML.</p> <p>In both cases, the content-type for the session's client type must equal text/html in order for the method to return true.</p>	<p>id (optional)</p> <p>scope (optional)</p>
processEdit	<p>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 getEdit() 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.</p>	<p>id (optional)</p> <p>scope (optional)</p>
getContainerClassName	<p>Returns the class name of the container that backs the container.</p>	<p>id (optional)</p> <p>scope (optional)</p>
getSelectedChannels	<p>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.</p>	<p>id (required)</p> <p>scope (optional)</p>
getAvailableChannels	<p>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</p>	<p>id (required)</p> <p>scope (optional)</p>
scontent	<p>Returns the URL of the directory that has the static content (for example, images and style sheet). This utility tag can be used anywhere.</p>	<p>none</p>





## desktopProviderContext.tld Tags

---

This chapter lists the `desktopProviderContext.tld` tags and their attributes. All of the normal tags in the `desktopProviderContext.tld` file have a `bodycontent` of empty.

### desktopProviderContext.tld Normal Tags with Attributes

TABLE 56-1 desktopProviderContext.tld Tags with Attributes

Tag Name	Description	Attributes
<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.	<ul style="list-style-type: none"> <li>key (required) - the name of the property</li> <li>localized (optional)</li> <li>id (optional)</li> <li>scope (optional)</li> <li>pflist (optional)</li> </ul>
<code>getBooleanProperty</code>	Returns a Boolean that gives the value of the property.	<ul style="list-style-type: none"> <li>key (required) - the name of the property</li> <li>id (optional)</li> <li>scope (optional)</li> <li>pflist (optional)</li> </ul>

TABLE 56-1 desktopProviderContext.tld Tags with Attributes (Continued)

Tag Name	Description	Attributes
getCollectionProperty	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.	key (required) - the name of the property id (optional) scope (optional) pflist (optional)
getIntegerProperty	Returns an integer with the integer property. Throws an error if the property does not exist.	key (required) - the name of the property id (optional) scope (optional) pflist (optional)
getProperty	Returns a Java Object with a property. The value returned from this method is a Java Object of type String, Integer, Boolean, or Map. Throws an error if the property does not exist.	key (required) - the name of the property id (optional) scope (optional)
setStringProperty	Sets a string property.	key (required) - the name of the property value (required) - the value of the property to be set pflist (optional)
setBooleanProperty	Sets a Boolean property.	key (required) - the name of the property value (required) - the value of the property to be set pflist (optional)

TABLE 56-1 desktopProviderContext.tld Tags with Attributes (Continued)

Tag Name	Description	Attributes
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 (optional) scope (optional)
getClientPropertiesFilters		id (optional) scope (optional)
getClientAndLocalePropertiesFilters		id (optional) 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)

TABLE 56-1 desktopProviderContext.tld Tags with Attributes (Continued)

Tag Name	Description	Attributes
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)
getDesktopURL	Returns a string with the Desktop URL. The Desktop URL is the absolute URL used to access the Desktop application. For example: http://server:port/portal/dt. 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)

TABLE 56-1 desktopProviderContext.tld Tags with Attributes (Continued)

Tag Name	Description	Attributes
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 get*Property() methods. Whether a particular value is considered a property or an attribute depends on the underlying implementation of ProviderContext.	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 set*Property() methods. Whether a particular value is considered a property or an attribute depends on the underlying implementation of ProviderContext.	key (required) - the name of the attribute value (required) - the value of the attribute
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)

TABLE 56-1 desktopProviderContext.tld Tags with Attributes (Continued)

Tag Name	Description	Attributes
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)
isLogWarningEnabled	Returns a Boolean; true if the log level is set to warning or higher; otherwise false.	id (optional) scope (optional)

TABLE 56-1 desktopProviderContext.tld Tags with Attributes (Continued)

Tag Name	Description	Attributes
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)

TABLE 56-1 desktopProviderContext.tld Tags with Attributes (Continued)

Tag Name	Description	Attributes
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)



## desktopContainerProviderContext.tld Tags

---

This chapter lists the `desktopContainerProviderContext.tld` tags and their attributes. All of the normal tags in the `desktopProviderContext.tld` file have a bodycontent of empty.

### desktopContainerProviderContext.tld Tags with Attributes

TABLE 57-1 desktopContainerProviderContext.tld Tags with Attributes

Tag Name	Description	Attributes
getContent	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> .	channel (required) - the name of the channel id (optional) scope (optional)



# desktopTab.tld Tags

---

This chapter lists the desktopTab.tld tags and their attributes. All of the normal tags in the desktopTab.tld file have a bodycontent of empty.

## desktopTab.tld Normal Tags with Attributes

TABLE 58-1 desktopTab.tld Tags with Attributes

Tag Name	Description	Attributes
getAvailableTabs	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.	id (optional) scope (optional)
getSelectedTabs	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.	id (optional) scope (optional)
getSelectedTab	Returns the selected tab, the current selected Unmodifiable Tab in the user's session.	id (optional) scope (optional)
getMakeTab	Returns the make tab, the tab spec to be used for "Make My Own tab" creation by the user.	id (optional) scope (optional)
getStartTabName	Returns a string with the start tab Name, the name of the tab to be displayed when the user logs in.	id (optional) scope (optional)
getSelectedTabName	Returns a string with the selected tab Name, the current selected tab in the user's session.	id (optional) scope (optional)

TABLE 58-1 desktopTab.tld Tags with Attributes (Continued)

Tag Name	Description	Attributes
getTabURL	Returns the Tab URL. This method gets the tab URL used to switch the selected tab on the user's desktop.	id (optional) scope (optional)
getName	Returns a string with the name of the tab.	id (optional) scope (optional)
getDesc	Returns a string with the description of the tab.	id (optional) scope (optional)
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 true if the tab is removable; otherwise false.	id (optional) scope (optional)
isRenamable	Returns a Boolean that gives the renamable status of the tab. Returns true if the tab is renamable; otherwise false.	id (optional) scope (optional)

# desktopTable.tld Tags

---

This chapter lists the `desktopTable.tld` tags and their attributes. All of the normal tags in the `desktopTable.tld` file have a `bodycontent` of empty.

## desktopTable.tld Normal Tags with Attributes

TABLE 59-1 desktopTable.tld Tags with Attributes

Tag Name	Description	Attributes
<code>getColumns</code>	Returns a list of channel names that belong in that particular column.	<code>column</code> (required) <code>id</code> (required)
<code>getColumnWidth</code>	Returns the width of a column (a percentage with respect to the entire Desktop). Valid columns are left, center, and right.	<code>column</code> (required) <code>id</code> (required)
<code>getHasFrame</code>	Returns a Boolean that gives the frame status of the channel. Returns true if the channel has a frame; otherwise false.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getIsMinimized</code>	Returns a Boolean that gives the minimized status of the channel. Returns true if the channel is minimized; otherwise false.	<code>id</code> (optional) <code>scope</code> (optional)
<code>getIsMovable</code>	Determines whether the channel is movable or not and returns a Boolean that gives the movable status of a channel.	<code>id</code> (optional) <code>scope</code> (optional)

TABLE 59-1 desktopTable.tld Tags with Attributes (Continued)

Tag Name	Description	Attributes
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 minMaximizedCommand, helpCommand, editCommand, detachAttachCommand, and removeCommand.	id (optional) scope (optional)
getIsDetached	Returns a Boolean that gives the detached status of the channel. Returns true if the channel is detached; otherwise false.	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 getIsDetached should be used to verify that a channel has been detached.	id (optional) 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 true if the table container action is a popup and returns false 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)
getChannelsInCategory	Returns the map values of channel names included in the category.	category (required) id (required) scope (optional)

## desktopTheme.tld Tags

---

This chapter lists the `desktopTheme.tld` tags and their attributes. All of the normal tags in the `desktopTheme.tld` file have a `bodycontent` of empty.

### desktopTheme.tld Normal Tags with Attributes

TABLE 60-1 desktopTheme.tld Tags with Attributes

Tag Name	Description	Attributes
<code>getGlobalThemes</code>	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.	<code>id</code> (required) <code>scope</code> (optional)
<code>getSelectedName</code>	Returns the name of the selected theme. The name returned can be the name of one of the globally defined themes or <code>CustomTheme</code> . If <code>CustomTheme</code> is returned, this means the user has defined and is using the custom defined theme.	<code>id</code> (required) <code>scope</code> (optional)
<code>setSelectedName</code>	Sets a theme for the current user. The theme to be set can be one of the globally defined themes or a <code>CustomTheme</code> .	<code>value</code> (required) - the name of the theme set

TABLE 60-1 desktopTheme.tld Tags with Attributes (Continued)

Tag Name	Description	Attributes
getAttribute	Returns the value of a theme attribute. Get a specific attribute value of a specific theme.	<p>name (required) - the name of the attribute. Possible values are: bgColor, borderColor, titleBarColor, fontColor, borderWidth and fontFace</p> <p>theme (optional) - the name of the theme. If this is not specified, the currently selected theme is used.</p> <p>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.</p> <p>id (optional)</p> <p>scope (optional)</p> <p>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.</p>



TABLE 60-1 desktopTheme.tld Tags with Attributes (Continued)

Tag Name	Description	Attributes
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



# searchContext tag in search.tld

---

This chapter lists the normal tags and attributes for the searchContext Java API in the Search tag library.

## search.tld Normal Tags with Attributes for the searchContext

TABLE 61-1 search.tld Tags with Attributes for the searchContext

Tag Name	Description	Attributes
searchContext	Main outer tag that encloses all other tags.	bodycontent - JSP rdmServer (optional) - search server - http://.../search1/search or https://.../search1/search rdmType (optional) - rd-request (default), taxonomy-request, schema-request, server-request or status-request ql (optional) - query language query (optional) - dynamic attribute



## Pre-Search Tags in search.tld

---

This chapter lists the pre-search tags and attributes in the `search.tld` file. These tags are used before executing a search. Before a successful search can be executed these tags must set a value for:

- `setRDMServer`
- `setRDMDType`
- `setQuery` or `setCriteria`

### search.tld Tags for Pre-Search

TABLE 62-1 search.tld Tags for Pre-Search

Tag Name	Description	Attributes
<code>setRDMServer</code>	Sets the <code>RDMServer</code> variable. The server URL should be set explicitly. Format: <code>http://</code> or <code>https://hostname:port/search1/search</code> Value has to be set.	<code>rdmServer</code> (required) - server URL; can be an expression or a hardcoded string value.

TABLE 62-1 search.tld Tags for Pre-Search (Continued)

Tag Name	Description	Attributes
setRDMDType	<p>Sets the RDM Request type. RDMDType - Can be one of:</p> <p>rd-request: The default request type. Resource descriptions (documents).</p> <p>taxonomy-request: Taxonomy.</p> <p>schema-request: The schema.</p> <p>server-request: Server information.</p> <p>status-request: Server status information.</p> <p>Set value explicitly; the default set by the system may not return the expected results.</p>	rdmType (required) - string
setViewAttributes	Sets the search result attributes that are returned for the search. This is an optional tag. It assumes the default values if not set explicitly.	viewAttributes (required) - string; null (all) or comma delimited list of attributes.
setSessionID	The Search Server needs to validate the user's identity for document level security. This tag is a wrapper for setSessionID(String) method in SearchContext. The JSP gets the portal access Token string and passes it along to the search server using this tag.	sessionID (required) - string
setViewOrder	Sets the sorting order for results.	<p>viewOrder (required) - string; null or comma delimited list of attributes each proceeded with + for ascending order or - for descending order. For example:</p> <p>—score — sort results based on the descending relevancy (default)</p> <p>+title, —score — sort results by title first, score second.</p>

TABLE 62-1 search.tld Tags for Pre-Search (Continued)

Tag Name	Description	Attributes
setCategory	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 executeSearch tag.	category (required) - string: current category level; if not set, the search is not restricted by category. For example, category=Internal:Engineering;
setSearchAll	Specifies whether to search in the current category only, or in all the sub-categories. Default value is true.	searchAll (required) - Boolean
setFirstHit	setFirstHit 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 setFirstHit() method in the Search API. Results are returned from this hit. This is an optional tag. Alternately you can use setPage and setViewHits.	fromHit (required) - integer
setViewHits	Sets the maximum number of hits returned.	viewHits (required) - integer - dynamic attribute
setQuery	Query string. Either this value has to be set or else setCriteria has to be set.	query (required) - string - dynamic attribute
setDatabase	Sets the name of the database to search.	database (required) - string - dynamic attribute
setPage	Sets the first hit by taking the view hits (using setViewHits) as a page size. Do not use, setPage and setFirstHit together..	page (required) - integer - dynamic attribute
setQueryLanguage	Sets the query language. ql - Can be one of: search: The default Search query language. Searches documents or the taxonomy. taxonomy-basic: Used for requesting branches or parts of the taxonomy. schema-basic: Queries the Search schema. url: Retrieves RDs by URL (scope=url).	ql (required) - string

TABLE 62-1 search.tld Tags for Pre-Search (Continued)

Tag Name	Description	Attributes
setCriteria	<p>Sets the query string in a list format. Useful in advanced search.</p> <p>The list should have an operand, operator and a value. The list of valid operator's are defined in the SearchContext API. The operand can be a schema field.</p> <p>The user can bypass this tag and just use the setQuery tag directly by converting a complex query into the syntax that the search engine requires.</p> <p>The setCriteria tag is a wrapper for the setScope(list) method in the searchContext API. This method basically parses the list and converts it into a string. For example, author CONTAINS xyz.</p> <p>Either this value has to be set or else setQuery has to be set.</p>	criteria (required) - string - dynamic attribute



## Execute Search tag in search.tld

---

The `search.tld` includes the tag that executes the search.

### desktop.tld Normal Tags with Attributes

TABLE 63-1 desktop.tld Normal Tags with Attributes

Tag Name	Description	Attributes
<code>executeSearch</code>	Tag for the <code>execute</code> method in <code>SearchContext</code> API. Executes search after doing some validation of the search parameters and query string.	none



## Post Search Tags in search.tld

---

This chapter lists the tags and attributes in `search.tld` file. These tags are related to search results and are used after a search is executed. They provide various counts and help display the search results.

### desktop.tld Normal Tags with Attributes

TABLE 64-1 desktop.tld Normal Tags with Attributes

Tag Name	Description	Attributes
<code>getNextResult</code>	Search results iterator, creates a context for each result in the search result stream.	<code>bodycontent</code> - JSP <code>input</code> (required) -search result document <code>input</code> (required) — search result stream
<code>getResult</code>	Retrieves the result stream from the previous search.	<code>id</code> (optional) <code>scope</code> (optional)

TABLE 64-1 desktop.tld Normal Tags with Attributes (Continued)

Tag Name	Description	Attributes
getValue	Returns a string value of the attribute or returns a string value of a multivalue attribute with index.  This tag must be used within the getNextResult tag.	Attribute (required) - string - attribute name  escape (true for HTML encoding)  id (optional)  scope (optional)  truncate (optional) - specifies the number of characters to return in the attribute
getUrl	Returns a string with the search result URL as a string. This tag must be used within the getNextResult tag.	escape (optional)  id (optional)
hasNextPage	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)
hasPreviousPage	Returns true if there is a previous page. Value based on viewHits and current page value.	id (optional)  scope (optional)
noHits	Returns true if no matching hits were found. This is a convenience tag.	id (optional)  scope (optional)
hitCount	Returns the total number of results that matched the query.	id (optional)  scope (optional)
toHit	Returns the last hit being displayed on a page. The value is based on firstHit and viewHits.	id (optional)  scope (optional)
page	Returns the current page. If the value is not set, calculates the page based on viewHits and firstHit.	id (optional)  scope (optional)
totalDocuments	Returns the total number of documents in the database.	id (optional)  scope (optional)
totalPages	Returns the total number of pages of hits that are available. The value is calculated from viewHits and hitCount.	id (optional)  scope (optional)

TABLE 64-1 desktop.tld Normal Tags with Attributes (Continued)

Tag Name	Description	Attributes
getResultCount	Returns the number of results returned by the search. Returns -1 for an error.	id (optional) scope (optional)



## Miscellaneous Tags in search.tld

---

This chapter lists the other available tags and their corresponding attributes in `search.tld` file. All these tags in the `search.tld` file have a `bodycontent` of empty.

### search.tld Normal Tags with Attributes

TABLE 65-1 search.tld Normal Tags with Attributes

Tag Name	Description	Attributes
<code>getSearchString</code>	Returns a string with a the search query being executed. Good for debugging.	id (optional) scope (optional)
<code>getCategory</code>	Returns a string with the current category name or else set to root.	id (optional) scope (optional)
<code>getFirstHit</code>	Returns the starting hit being displayed.	id (optional) scope (optional)
<code>getSessionID</code>	Returns a string with a user's session id. It has no value unless previously set.	id (optional) scope (optional)
<code>getQuery</code>	Returns the query string; returns an empty string if not set.	id (optional) scope (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.	id (optional) scope (optional)
<code>getDatabase</code>	Returns the database being searched.	id (optional) scope (optional)

TABLE 65-1 search.tld Normal Tags with Attributes (Continued)

Tag Name	Description	Attributes
getMaximumPages	Returns the maximum number of pages from any search executed in the current search context. Useful for building the previous or next page iterator.	id (optional) scope (optional)



# Desktop Template Common Tags

---

This chapter contains a listing of all the common tags in the Desktop template files.

## Desktop Template Common Tags

[tag:fontFace]	Font chosen
[tag:fontFace1]	Default font for the Desktop (sans-serif)
[tag:iwtDesktop-fontFace1]	Default font for the Desktop (sans-serif)
[tag:desktop-fontFace1]	Default font for the Desktop (sans-serif)
[tag:fontColor]	Color of font chosen
[tag:bgColor]	Provider background color chosen
[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:staticContent]	Directory you defined for the deployment URI during installation.
[tag:localeString]	Directory designation for chosen locale
[tag:productName]	Product name
[tag:providerTitle]	Provider/Channel title
[tag:title]	The title of the provider/channel
[tag:selectedName]	Name of selected tab
[tag:frontContainerName]	Name of the front container

[tag:parentContainerName]	The top level container name for the TemplateTabContainer
[tag:providerName]	The channel name
[tag:channelName]	Channel name as defined in the channel display profile definition.
[tag:theme_channel]	The theme edit channel name at run time, this will be either the presetThemeContainer or the customThemeContainer.
[tag:ErrorMessage]	error content
[tag:providerContent]	Content of the provider
[tag:detachedContent]	Channel content in the detached window
[tag:content]	Content of the channel
[tag:detachedContent]	Channel content in the detached window
[tag:fullTopContent]	Provider/channel content (HTML) inserted here
[tag:name]	display value
[tag:stackTrace]	Produces a stack trace
[tag:MaximizedContent]	Inserts content in the maximize mode
[tag:minimizeText]	A text that is used as an alternate for the minimize icon
[tag:maximizeText]	A text that is used as an alternate for the maximize icon
[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.
[tag:removeText]	Text for the alt tag for the remove icon
[tag:detachAttachText]	Text for the alt tag for the detach/attach image in the title bar, this is a localized string. The detach or attach mode is determined at run time.
[tag:minMaximizeIcon]	Inserts minimize /maximize icon
[tag:removeTag]	Alternated string for the remove image in the title bar, this is a localized string

---

[ tag:editTag]	Text for the alt tag for the edit icon
[ tag:help_tag]	Text for alt message
[ tag:bulletColor]	Inserts bulletColor.js into this template
[ tag:toolbarRollover]	Inserts toolbarRollover.js into this template.
[ tag:banner]	Inserts banner.template into this template
[ tag:menubar]	Inserts menubar.template into template
[ tag:contentBarInContent]	Inserts contentBarInContent.template into this template
[ tag:contentBarInLayout]	Inserts contentBarInLayout.template into this template
[ tag:arrangeProvider]	Inserts arrangeProvider.js template into this template
[ tag:performSubstitution]	Inserts performSubstitution.js template
[ tag:performColumnSubstitution]	Inserts performColumnSubstitution.js template
[ tag:selectAll]	Inserts selectAll.js template
[ tag:switchColumns]	Inserts switchColumns.js template
[ tag:layoutFullTop]	Inserts layoutFullTop.template into this template
[ tag:layoutFullBottom]	Inserts layoutFullBottom.template into this template
[ tag:openURLInParent]	Inserts openURLInParent.js template
[ tag:popupMenubar]	Inserts popupMenubar.template template
[ tag:launchPopup]	Inserts launchPopup.js template
[ tag:inlineError]	Replaced with inlineError.template if error has occurred
[ tag:removeCommand]	Inserts the removeCommand.template
[ tag:detachAttachCommand]	If the channel is detached, insert the detachCommand.template; if the channel is attached, then insert the attachCommand.template.
[ tag:editCommand]	Inserts the editCommand.template

[tag:helpCommand]	Inserts helpHref.template
[tag:minMaximizeCommand]	Inserts the minMaximizeCommand.template
[tag:resourceName]	Used to dynamically build edit pages. Should not be edited.
[tag:header]	Header.
[tag:attName]	Attribute Name.
[tag:attSelected]	Selected attribute.
[tag:attValue]	Attribute value.
[tag:string]	String.
[tag:selected]	Selected.
[tag:options]	Options.
[tag:process]	Process.
[tag:isAppHandler]	isAppMandler.
[tag:mail-display-error]	Mail display error.
[tag:editLink]	Used to display link for the application helper editing / URL to edit
[tag:link]	URL location
[tag:logoutUrl]	the logout URL
[tag:desktop_url]	URL of Desktop to return to
[tag:removeURL]	URL of the channel to be removed

?action=process&provider=  
thecontainername&thecontainername.  
channelAction=remove&  
thecontainername.targetProvider =providername

[tag:url]	URL of new location
[tag:editURL]	URL of Edit page for this channel
	?action=edit&provider= theeditcontainername&targetprovider= providername& containerName=thecontainername
[tag:detachAttachURL]	URL of the channel to be detached

```

action=process&provider=
thecontainername&thecontainername.channelAction=
attach&thecontainername.targetProvider=
providename

```

[tag:maximizeURL]	A URL to show the channel in maximize mode
[tag:minMaximizeURL]	A URL to show the channel in either the minimize or the normal mode, the mode is decided at run time
	URL of channel to minimize or maximize
	<pre> ?action=process&amp;provider= thecontainername &amp;thecontainername.channelAction= maximize&amp;thecontainername.targetProvide r=providename ?action=process&amp;provider= thecontainername&amp;thecontainername. channelAction= minimize&amp;thecontainername. targetProvid er=providename </pre>
[tag:s_detachImage]	Run time path for the detach image
[tag:s_editImage]	Run time path for the edit image
[tag:s_removeImage]	Run time path for the remove image
[tag:s_helpImage]	Run time path for the help image
[tag:s_minimizeImage]	Run time path for the minimized or the maximized image, the state of minimize or maximize is determined at run time.
[tag:s_normalizeImage]	Run time path for the normalized image
[surl:/desktop/css/style.css]	Style sheet used by the Desktop for the banner and tabs templates
[surl:/docs/en/desktop/usedesk.htm]	Help link for Desktop
[surl:/docs/en/desktop/fdesktop.htm]	Help link for frames on the Desktop
[surl:/desktop/images/nothing.gif]	Used to space gifs or channels
[surl:/desktop/images/b_up.gif]	Icon that shows up arrow
[surl:/desktop/images/b_down.gif]	Icon that shows down arrow

<code>[surl:/desktop/images/b_left.gif]</code>	Icon that shows arrow pointing left
<code>[surl:/desktop/images/b_right.gif]</code>	Icon that shows arrow pointing right
<code>[surl:/desktop/images/b_normal.gif]</code>	A URL that points to the normal image
<code>[surl:/desktop/images/b_minimize.gif]</code>	A URL that points to the maximize image
<code>[surl:/desktop/images/b_maximize.gif]</code>	A URL that points to the maximize image
<code>[surl:/desktop/images/b_attach.gif]</code>	A URL that points to the attach image
<code>[surl:/desktop/images/layout1.gif]</code>	Icon for thin-wide layout
<code>[surl:/desktop/images/layout2.gif]</code>	Icon for wide-thin layout
<code>[surl:/desktop/images/layout3.gif]</code>	Icon for thin-wide-thin layout
<code>[surl:/desktop/images/layout4.gif]</code>	Icon for thin-thin-thin layout
<code>[surl:/images/blueBullet.gif]</code>	Blue button to denote Home, Help, Logout, and so on
<code>[surl:/images/redBullet.gif]</code>	Red button to denote home, help, logout, or whatever has been chosen.
<code>[surl:/images/spacer.gif]</code>	Used to space gifs or channels
<code>[surl:/images/productName.gif]</code>	Logo gif of the product name
<code>[surl:/images/blueBullet.gif]</code>	Blue button to denote Home, Help, Logout, and so on
<code>[tag:help_link]</code>	Help link for Desktop
<code>[tag:help_icon]</code>	Icon for help in channel title bar
<code>[tag:provider_cmds]</code>	Inserts the channel command button links (maximize, detach, and so on)
<code>[tag:serviceTimeout]</code>	Provider Timeout in seconds
<code>[tag:theme_channel]</code>	The theme edit channel name at run time, this will be either the presetThemeContainer or the customThemeContainer.
<code>[tag:detachedContent]</code>	Channel content in the detached window

# Provider-Specific Desktop Template Tags

---

This chapter contains the following sections:

- “AddressBookProvider” on page 399 and “CalendarProvider” on page 401,
- “AppProvider” on page 400 and “LoginProvider” on page 410

## AddressBookProvider

The tags described below are used by AddressBookProvider and providers who extend this provider (such as LotusNotesAddressBookProvider and MExchangeAddressBookProvider).

[tag:ab-display-clientURL-appURL]	Used to display the application launch link
[tag:ab-display-entry-list]	Used as a placeholder to put in all of the address book entries
[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
[tag:ab-display-error]	Displays error message
[tag:ab-display-summary-entries]	Displays a summary of the entries
[tag:ab-display-summary]	Displays summary
[tag:ab-display-entries]	Displays entries
[tag:ab-display-clientURL]	Ab display client URL

## AppProvider

The tags described below are used by AppProvider.

[tag:windowOption]	JavaScript variable default for launching Bookmark windows (taken from preferences)
[tag:bookmarks]	List of bookmark links
[tag:resourceCount]	Number of bookmarks
[tag:resourceName]	Name of the bookmark
[tag:resourceURL]	URL of the bookmark
[tag:resourceList]	Checkable list of bookmarks for Edit page in table format
[tag>windowOptions]	Default for checkboxes of how the bookmark should be opened (new window, existing window, and so on)
[tag:index]	Used to reference the URLs in the edit page
[tag:targetName]	Name of Bookmark used on edit page
[tag:targetValue]	URL for bookmark used on edit page
[tag:all_new_checked]	The value for this is either CHECKED or "" based on the window preference specified in edit page
[tag:ownWindow]	Localized string from resource bundle displaying the text for window options on the edit page.
[tag:one_new_checked]	The value for this is either CHECKED or "" based on the window preference selected in edit page
[tag:singleWindow]	Localized string from resource bundle displaying the text for window options on the edit page.
[tag:same_checked]	The value for this is either CHECKED or "" based on the window preference selected in edit page
[tag:mainWindow]	Localized string from resource bundle displaying the text for window options on the edit page.
[tag:link]	URL for bookmark used for constructing the channel content.
[tag:name]	Name of bookmark used in the channel content.



## CalendarProvider

The tags described below are used by CalendarProvider and providers who extend this provider (such as LotusNotesCalendarProvider and MExchangeCalendarProvider).

- [tag:calendar-display-client-uri]  
Used to display the application launch link
- [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-monthView-event-startHour1]  
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:calendar-display-dayView-event-startHour2]  
start hour based on user preference of 12 or 24 hour clock format
- [tag:calendar-display-dayView-task-dueHour2]  
end hour based on user preference of 12 or 24 hour clock format
- [tag:calendar-display-dayView-task-pendHour2]  
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-endAmPm]  
Calendar display day view event
- [tag:calendar-display-dayView-task-dueAmPm]  
Calendar display day view task
- [tag:calendar-display-dayView-task-pendAmPm]  
Calendar display day view task

- [tag:calendar-display-dayView-event-endHour2]  
end hour based on user preference of 12 or 24 hour clock format
- [tag:calendar-display-dayView-task-dueMinute]  
minutes of the due time
- [tag:calendar-display-dayView-event-endMinute]  
minutes of the end time
- [tag:calendar-display-dayView-description-seperator]  
description separator specified by the resource bundle "seperatorDescription" value
- [tag:calendar-display-dayView-event-description]  
event description
- [tag:calendar-display-dayView-event-location]  
event location
- [tag:calendar-display-dayView-event-summary]  
event summary
- [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
- [tag:calendar-display-dayView-task-summary]  
task summary
- [tag:calendar-display-dayView-task-description]  
task description
- [tag:calendar-display-dayView-overdueTask-description]  
Calendar display day view overdue task
- [tag:calendar-display-dayView-task-d ueHour1]  
hour of task for 12-hour clock
- [tag:calendar-display-dayView-task-d ueMonth]  
month the task was due
- [tag:calendar-display-dayView-task-dueDay]  
day of the month the task was due
- [tag:calendar-display-dayView-task-dueYear]  
year the task was due
- [tag:calendar-display-dayView-overdueTask-summary]  
overdue task summary
- [tag:calendar-display-dayView-task-pendMinute]  
minutes of the task

- 
- [tag:calendar-display-dayView-task-location]  
task location
  - [tag:calendar-display-dayView-task-complete-start]  
start time of completed task
  - [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:calendar-display-dayView-taskList]  
task list content
  - [tag:calendar-display-dayView-eventList]  
event list content
  - [tag:calendar-display-dayView-otherTaskList]  
other tasks list content
  - [tag:calendar-display-dayView-overdueTaskNum]  
number of overdue tasks for today
  - [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:display-dayView-overdueTask-Header]  
if overdue tasks exist, then the overdue task header is identified by the resource bundle "overdueTasks" value
  - [tag:display-dayView-dueEvent-Header]  
If events exist, then the event header is identified by the resource bundle "dueEvents" value
  - [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-event-conflict]  
If the event is in conflict, then it is identified by the resource bundle "conflict" value.
  - [tag:calendar-display-error]  
error message

- [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
- [tag:calendar-display-monthView-event-endHour1]  
end hour for 0 based 12-hour clock
- [tag:calendar-display-monthView-event-startHour2]  
start hour based on user preference of 12 or 24 hour clock format
- [tag:calendar-display-monthView-event-startMinute]  
minutes of the start time
- [tag:calendar-display-monthView-event-startAmPm]  
am or pm identifier
- [tag:calendar-display-monthView-event-endHour2]  
end hour based on user preference of 12 or 24 hour clock format
- [tag:calendar-display-monthView-event-endMinute]  
minutes of the end time
- [tag:calendar-display-monthView-event-endAmPm]  
am or pm identifier
- [tag:calendar-display-monthView-event-summary]  
event summary
- [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
- [tag:calendar-display-monthView-task-pendHour2]  
hour based on user preference of 12 or 24 hour clock format

---

[tag:calendar-display-monthView-task-pendAmPm]  
am or pm identifier

[tag:calendar-display-monthView-task-pendMinute]  
minutes of the task

[tag:calendar-display-monthView-task-summary]  
task summary

[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

[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

[tag:calendar-display-summary-events]  
event summary information

[tag:calendar-display-summary-tasks]  
task summary information

- [tag:calendar-display-summary-events]  
event summary
- [tag:calendar-display-summary-tasks]  
task summary
- [tag:calendar-display-weekView-currentDayOfWeek]  
today's day of week
- [tag:calendar-display-weekView-currentMonth]  
today's month
- [tag:calendar-display-weekView-currentDay]  
today's day in month
- [tag:calendar-display-weekView-currentYear]  
today's year
- [tag:calendar-display-weekView-event-startHour1]  
start hour for 0 based 12-hour clock
- [tag:calendar-display-weekView-event-endHour1]  
end hour for 0 based 12-hour clock
- [tag:calendar-display-event-conflict]  
if the event is in conflict, then it is identified by the resource bundle "conflict" value.
- [tag:calendar-display-weekView-event-startHour2]  
start hour based on user preference of 12 or 24 hour clock format
- [tag:calendar-display-weekView-event-startMinute]  
minutes of the start time
- [tag:calendar-display-weekView-event-startAmPm]  
am or pm identifier
- [tag:calendar-display-weekView-event-endHour2]  
end hour based on user preference of 12 or 24 hour clock format
- [tag:calendar-display-weekView-event-endMinute]  
minutes of the end time
- [tag:calendar-display-weekView-event-endAmPm]  
am or pm identifier
- [tag:calendar-display-weekView-event-summary]  
event summary
- [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
- [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
- [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: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
[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: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

## LoginProvider

The tags described below are used by LoginProvider.

[tag:persistentCookie] Inserts the persistent cookie template

[tag:libertyLogin] Inserts the libertyLogin.Template

[tag:loginHelpUrl] Help link for login

[tag:preLoginURL] Inserts the liberty preLogin URL. The value is specified in the channel property preLoginURL which is typically of the form:

```
http://www.siroe.com:80/amserver/preLogin?metaAlias=  
www.siroe.com&goto=http://www.siroe.com:80/portal/dt
```

# Instant Messaging Tags

---

The IMProvider content page uses a custom tag library defined in a file called `im.tld` which is installed into the `/var/opt/SUNWportal/portals/<portal_id>/desktop/default/tld` directory. The `im.tld` file defines the following tags:

## Tags in im.tld

TABLE 68-1 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 Java System Access Manager software <code>SSOToken</code> or the user's password.)	<code>id</code> (optional) <code>scope</code> (optional)

TABLE 68-1 Tags in im.tld (Continued)

Tag Name	Description	Attributes
obtainContact	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.	contact (required) - The internal identifier of the contact, typically obtained from the getContactGroup tag.
getContactPresence	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.	id (optional) scope (optional)
getContactName	Return the common name for the contact.	id (optional) scope (optional)
getContactUsername	Returns the instant messaging user name for the contact.	id (optional) scope (optional)
getDateTime	Returns the update time.	format (required) id (optional) scope (optional)
isSecureMode	Returns boolean indicating whether the channel is being accessed through the Secure Remote Access Gateway component and the Netlet is loaded.	id (optional) scope (optional)
getCodebase	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.	id (optional) scope (optional)
getIMServer	Returns the -server 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.	id (optional) scope (optional)

**Note** – 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.

PART IX

Desktop Templates

- Chapter 69
- Chapter 70
- Chapter 71
- Chapter 72



# Overview of Desktop Templates

---

This chapter contains the following sections:

- “Introduction” on page 415
- “Installation Location” on page 415
- “The Desktop and Template Files” on page 416
- “File Lookup Scenario” on page 416

## Introduction

To generate the rendered Desktop user interface (what the industry refers to as the “presentation”), the Sun Java System Portal Server 7.1 software makes use of either JSPs or template files.

## Installation Location

The default set of template files are installed in `/var/opt/SUNWportal/portals/<portal_id>/desktop/default` directory. The developer sample template files are installed in `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_sample` and `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_anonymous_sample` directories. Files in the `/var/opt/SUNWportal/portals/<portal_id>/desktop/developer_anonymous_sample` directory are specific to the developer sample Anonymous Desktop.

## 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 7.1. 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 7.1 providers be based on JSP and the Portal Server 7.1 JSPProvider. However, there may be cases where the simplicity of Desktop templates provides an advantage. Desktop templates are fully supported in Portal Server 7.1. Many pieces of the product, such as the communications channels, continue to use them.

## File Lookup Scenario

The Portal Server software uses the lookup scenario outlined in this section to find the JSPs it needs. Use this order to decide the final location of your own JSPs.

### EXAMPLE 69-1 Template File Lookup Scenario

```
desktoptype_  
locale/  
channelname/  
clientPath
```

```
desktoptype_  
locale/  
provider/  
clientPath
```

```
desktoptype_  
locale/  
channelname
```

```
desktoptype_  
locale/  
provider
```

```
desktoptype_  
locale/  
clientPath
```



## EXAMPLE 69-1 Template File Lookup Scenario (Continued)

```

    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/

```

## EXAMPLE 69-1 Template File Lookup Scenario (Continued)

```

        channelname
default/
        provider
default/
        clientPath
default
        templateroot

```

If there is no `clientPath` specified, then the directory search order is as follows:

EXAMPLE 69-2 Template File Lookup Scenario (no `clientPath`)

```

        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

```

---

**EXAMPLE 69-2** Template File Lookup Scenario (no `clientPath`) (Continued)*templateroot*

The lookup scenario relies on the following parameters:

<code>desktoptype</code>	For example <code>default</code> (set in the administration console). Note that <code>desktoptype</code> is now a comma separated string list and so the look up will be based on the <code>desktoptype(s)</code> that are defined in the <code>desktoptype</code> attribute.
<code>locale</code>	Preferred <code>locale</code> is the user's locale. For example, <code>en_US</code> (set by users through the administration console in the "User" setting)
<code>clientPath</code>	This is an optional file-path containing client-specific templates; for example, <code>html</code> (set through the administration console Client Detection service)
<code>channelname</code>	This is the name of the channel; for example, <code>newSingleContainer</code> (set in the display profile)
<code>provider</code>	This is the provider name; for example, <code>JSPSingleContainerProvider</code> (set in the display profile)
<code>templateroot</code>	This is defined in the <code>desktopconfig.properties</code> file. The root of the search directory (default value of <code>/var/opt/SUNWportal/portals/&lt;portal_id&gt;/desktop/</code> ) can be changed by modifying the <code>templateBaseDir</code> property in the <code>desktopconfig.properties</code> file.



# Desktop Templates in the default Directory

---

This chapter contains the following sections

- “AddressBookProvider” on page 421, “AppProvider” on page 422, “BookmarkProvider” on page 422 and “CalendarProvider” on page 423
- “error” on page 425
- “LoginProvider” on page 426
- “default” on page 426 directory template files

## AddressBookProvider

The following table lists the templates in the `html` subdirectory of the `AddressBookProvider`, `LotusNotesAddressBookProvider`, and `MSExchangeAddressBookProvider` and includes a brief description of the template file. In this two columned table, the left column contains the template file name and the right column includes a brief description of the corresponding template file.

<code>display-clientURL.template</code>	Used for displaying the “Launch Address Book” link
<code>display-entries.template</code>	Used for formatting the table headers
<code>display-entry.template</code>	Used for formatting the display of the address book entry
<code>display-error.template</code>	Used for displaying error messages
<code>display-summary.template</code>	Used for formatting number of total and unread messages
<code>display.template</code>	Used for overall channel formatting
<code>edit-checkbox.template</code>	Used for creating edit page checkboxes
<code>edit-end.template</code>	Used for creating end of the edit page
<code>edit-link.template</code>	Used for creating application helper edit link
<code>edit-password.template</code>	Used for creating edit page password boxes

<code>edit-select.template</code>	Used for creating edit page select boxes
<code>edit-selectoption.template</code>	Used for creating edit page select box options
<code>edit-start.template</code>	Used for creating the start of the edit page
<code>edit-string.template</code>	Used for creating edit page text boxes
<code>edit.template</code>	Used for creating edit the template
<code>ma-edit-link.template</code>	ma edit link template
<code>ma-edit.template</code>	ma edit template

## AppProvider

The template available in the AppProvider subdirectory and its description is:

<code>display.template</code>	Contains the JavaScript code that launches the windows that the HTML applications show up in.
-------------------------------	---

## BookmarkProvider

The templates available in the `html` subdirectory of BookmarkProvider and their description are:

<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.
<code>edit.template</code>	Contains the formatting for the Edit page for the Bookmark provider.
<code>editUrlWrapper.template</code>	Used by the edit page of Bookmark Provider to draw the part where the bookmarks are removed.
<code>editWindowOption.template</code>	Contains the markup for the radio buttons used to select the window option.
<code>urlWrapper.template</code>	Contains markup for each URL shown in <code>display.template</code> .

# CalendarProvider

The templates available in the `html` subdirectory of the `CalendarProvider`, `LotusNotesCalendarProvider`, and `MSEExchangeCalendarProvider` and their description are:

`display-clientURL.template`

Used for displaying the Launch Calendar link.

`display-dayView-emptyEventList.template`

Used for displaying the message “No events scheduled for today.”

`display-dayView-emptyTaskList.template`

Used for displaying the message “No tasks are pending for today.”

`display-dayView-event.template`

Used for formatting events

`display-dayView-eventAllDay.template`

Used for formatting an all day event

`display-dayView-otherTasks.template`

Used for formatting other tasks

`display-dayView-overdueTasks.template`

Used for formatting overdue tasks

`display-dayView-task.template`

Used for formatting a “normal” task

`display-dayView.template`

Used for formatting the layout of all tasks and events

`display-error.template`

Used for displaying error messages

`display-monthView-dayOfWeek.template`

Used for creating week layout within the month

`display-monthView-emptyEventList.template`

Used for formatting when there are no events

`display-monthView-emptyTaskList.template`

Used for formatting when there are no tasks

`display-monthView-event.template`

Used for formatting an event

`display-monthView-eventAllDay.template`

Used for formatting an all day event

`display-monthView-task.template`

Used for formatting a task

`display-monthView-weekView.template`  
Used for formatting the week view with a month

`display-monthView.template`  
Used for generating the entire month layout

`display-summary-events.template`  
Used to show number of events

`display-summary-tasks.template`  
Used to show number of tasks

`display-summary.template`  
Used for formatting number of total and unread messages

`display-weekView-currentDayHeader.template`  
Used for formatting header for week view

`display-weekView-emptyEventList.template`  
Used for formatting an empty event list

`display-weekView-emptyTaskList.template`  
Used for formatting an empty task list

`display-weekView-event.template`  
Used for formatting an event

`display-weekView-eventAllDay.template`  
Used for formatting an all day event

`display-weekView-task.template`  
Used for formatting a task

`display-weekView.template`  
Used for the overall week view

`display.template`  
Used for overall channel formatting

`edit-checkbox.template`  
Used for creating edit page checkboxes

`edit-config-options.template`  
Edit config options template

`edit-end.template`  
Used for creating end of the edit page

`edit-link.template`  
Used for creating application helper edit link

`edit-password.template`  
Used for creating edit page password boxes



---

<code>edit-select.template</code>	Used for creating edit page select boxes
<code>edit-selectoption.template</code>	Used for creating edit page select box options
<code>edit-separate.template</code>	Edit separate template
<code>edit-start.template</code>	Used for creating the start of the edit page
<code>edit-string.template</code>	Used for creating edit page text boxes
<code>edit.template</code>	Edit template
<code>url.template</code>	Used for creating hyperlinks

## error

The templates available in the `error` subdirectory and their description are:

<code>banner.template</code>	The banner across the top of the Desktop pages.
<code>banner_nocontext.template</code>	Banner non context template
<code>error.template</code>	Error template
<code>error_nocontext.template</code>	Displayed when no context is available.
<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.
<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.
<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.

## LoginProvider

The templates available in the LoginProvider subdirectory and their description are:

<code>display.template</code>	Contains the JavaScript code that launches the login window.
<code>display_AuthLDAP.template</code>	Contains the JavaScript code that launches the LDAP login window.
<code>display_AuthUnix.template</code>	Contains the JavaScript code that launches the UNIX login window.
<code>libertyLogin.template</code>	Library login template.
<code>persistentCookie.template</code>	Partial HTML template for remembering the user's name and password.

## default

The templates available in the default directory and their description are:

<code>AttachCommand.template</code>	Handles reattaching a detached channel.
<code>banner.template</code>	The banner across the top of the Desktop pages
<code>contentLayout.template</code>	Content bar for template displayed on the Desktop before the user selects Content or Layout.
<code>detachCommand.template</code>	Handles detaching a channel.
<code>detachEditCommand.template</code>	Handles link to the Edit page for the detached channel.
<code>detachRemoveCommand.template</code>	Handles closing or removing a detached channel.
<code>editCommand.template</code>	Handles link to the Edit page for this channel.
<code>helpHref.template</code>	Generates the help URL for each of the channels. Displays the help contents in a new window.
<code>inlineError.template</code>	Inline error template.
<code>MaximizeCommand.template</code>	Allows the channel to be displayed in the maximize mode so that the channel occupies the entire Desktop.
<code>menubar.template</code>	HTML for the menubar across the bottom of the Desktop pages.

---

<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.
<code>minMaximizeCommand.template</code>	Handles minimizing and maximizing a channel.
<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.
<code>providerCommands.template</code>	Commands available in title bar.
<code>redirect.template</code>	Redirect template.
<code>removeCommand.template</code>	Removes the channel.
<code>bulletColor.js</code>	JavaScript code used to select and display bullet color.
<code>isPageCompletelyLoaded.js</code>	Is page completely loaded.
<code>openURLInParent.js</code>	JavaScript to open a URL in the parent window. Used in popup windows.
<code>pageLoaded.js</code>	Page loaded.
<code>toolbarRollovers.js</code>	JavaScript code use to display selection of Content or Layout by color change.



# Desktop Templates in the Developer Sample Directory

---

This chapter contains the following sections:

- “MyFrontPageTemplatePanelContainer” on page 429
- “PredefinedFrontPageTemplatePanelContainerProvider” on page 429 and “PredefinedSamplesTemplatePanelContainerProvider” on page 431
- “SamplesTemplatePanelContainer” on page 432 and “ToolsTemplatePanelContainer” on page 433
- “developersample” on page 433 directory template files

## MyFrontPageTemplatePanelContainer

The templates available in the MyFrontPageTemplatePanelContainer directory and their description are:

<code>banner.template</code>	The banner across the top of the Desktop pages.
<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.

## PredefinedFrontPageTemplatePanelContainerProvider

The templates available in the PredefinedFrontPageTemplatePanelContainerProvider directory and their description are:

<code>banner.template</code>	The banner across the top of the Desktop pages.
<code>bareProviderWrapper.template</code>	Template for each provider wrapper with no titlebar.
<code>contentBarInContent.template</code>	Content bar for template displayed when a user selects Content on any other page’s Content bar.

<code>contentBarInLayout.template</code>	Content bar for template displayed when user selects Layout on any other page's Content bar.
<code>contentLayout.template</code>	Content bar for template displayed on the Desktop before the user selects Content or Layout.
<code>contentTemplate.template</code>	The HTML template for the Content (Channels) page that displays when a user selects Content.
<code>layout1Template.template</code>	left/thin, right/wide.
<code>layout2Template.template</code>	left/wide, right/thin.
<code>layout3Template.template</code>	left/thin, center/wide, right/thin.
<code>layout4Template.template</code>	left/thin, center/thin, right/thin.
<code>layoutFullBottom.template</code>	Partial HTML template for Layout pages when the user has a full width channel available at the bottom of the layout.
<code>layoutFullTop.template</code>	Partial HTML template for Layout pages when the user has a full width channel available at the top of the layout.
<code>maximizedTemplate.template</code>	Template of a provider when it's in its maximized state.
<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.
<code>minimized.template</code>	Template of a provider when it's in its minimized state - just the title/button bar showing, no content area showing.
<code>optionsTemplate.template</code>	Template for the Options page of the Desktop.
<code>popupMenubar.template</code>	Menubar to be used in popup windows.
<code>popupTemplate.template</code>	Used to show provider/channel content in detached windows. Similar use to <code>providerWrapper.template</code> , but not in a table structure.
<code>providerWrapper.template</code>	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.
<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.
<code>arrangeProvider.js</code>	JavaScript code used in the Desktop Layout page.

<code>launchPopup.js</code>	JavaScript code to launch a popup window.
<code>performColumnSubstitution.js</code>	JavaScript code used on the Layout page.
<code>performSubstitution.js</code>	JavaScript code used on the Layout page.
<code>removeProvider.js</code>	JavaScript code used to remove a channel from the Desktop.
<code>selectAll.js</code>	JavaScript code used on the Layout page.
<code>switchColumns.js</code>	JavaScript code used on the Layout page.

## PredefinedSamplesTemplatePanelContainerProvider

The templates available in the `PredefinedFrontPageTemplatePanelContainerProvider` directory and their description are:

<code>banner.template</code>	The banner across the top of the Desktop pages.
<code>bareProviderWrapper.template</code>	Template for each provider wrapper with no titlebar.
<code>contentBarInContent.template</code>	Content bar for template displayed when a user selects Content on any other page's Content bar.
<code>contentBarInLayout.template</code>	Content bar for template displayed when user selects Layout on any other page's Content bar.
<code>contentLayout.template</code>	Content bar for template displayed on the Desktop before the user selects Content or Layout.
<code>contentTemplate.template</code>	The HTML template for the Content (Channels) page that displays when a user selects Content.
<code>layout1Template.template</code>	left/thin, right/wide.
<code>layout2Template.template</code>	left/wide, right/thin.
<code>layout3Template.template</code>	left/thin, center/wide, right/thin.
<code>layout4Template.template</code>	left/thin, center/thin, right/thin.
<code>layoutFullBottom.template</code>	Partial HTML template for Layout pages when the user has a full width channel available at the bottom of the layout.
<code>layoutFullTop.template</code>	Partial HTML template for Layout pages when the user has a full width channel available at the top of the layout
<code>maximizedTemplate.template</code>	Template of a provider when it's in its maximized state.

<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.
<code>minimized.template</code>	Template of a provider when it's in its minimized state - just the title/button bar are displayed and no content area is displayed.
<code>optionsTemplate.template</code>	Template for the Options page of the Desktop.
<code>popupMenubar.template</code>	Menubar to be used in popup windows.
<code>popupTemplate.template</code>	Used to show provider/channel content in detached windows. Similar use to <code>providerWrapper.template</code> , but not in a table structure.
<code>providerWrapper.template</code>	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.
<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.
<code>launchPopup.js</code>	JavaScript code to launch a popup window.
<code>performColumnSubstitution.js</code>	JavaScript code used on the Layout page.
<code>performSubstitution.js</code>	JavaScript code used on the Layout page.
<code>removeProvider.js</code>	JavaScript code used to remove a channel from the Desktop.
<code>selectAll.js</code>	JavaScript code used on the Layout page.
<code>switchColumns.js</code>	JavaScript code used on the Layout page.

## SamplesTemplatePanelContainer

The templates available in the SamplesTemplatePanelContainer directory and their description are:

<code>banner.template</code>	The banner across the top of the Desktop pages.
<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.



## ToolsTemplatePanelContainer

The templates available in the ToolsTemplatePanelContainer directory and their description are:

<code>banner.template</code>	The banner across the top of the Desktop pages.
<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.

## developersample

The templates available in the developersample directory and their description are:

<code>AttachCommand.template</code>	Handles reattaching a detached channel.
<code>MaximizeCommand.template</code>	Allows the channel to be displayed in the maximize mode so that the channel occupies the entire Desktop.
<code>detachEditCommand.template</code>	Handles link to the Edit page for the detached channel.
<code>detachCommand.template</code>	Handles detaching a channel.
<code>detachRemoveCommand.template</code>	Handles closing or removing a detached channel.
<code>editCommand.template</code>	Handles link to the Edit page for the detached channel.
<code>helpHref.template</code>	Generates the help URL for each of the channels. Displays the help contents in a new window.
<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.
<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.
<code>removeCommand.template</code>	Removes the channel.



# Portal Server : Java Enterprise System Monitoring Framework Integration

---

This chapter contains the following sections to explain the integration of Portal Server product and Java Enterprise System Monitoring Framework (JESMF):

- “JESMF” on page 435
- “JESMC” on page 436
- “Common Agent Container” on page 437
- “Portal Server:JESMF Detailed Mapping” on page 437
- “Portal Server:Desktop Detailed Mapping” on page 441
- “JESMF Extensions for Portal Server” on page 447
- “Installing and Configuring Portal Server:JESMF Integration” on page 448
- “Enabling or Disabling Portal Server:JESMF Integration” on page 449

## JESMF

JESMF is a Java Enterprise System (JES) common component that provides a single view of all the elements that compromise the JES stack. JESMF aims to simplify business by providing a single, consistent, qualified (Information relating to each component product is always equivalent. For example, a state of DEGRADED has the same meaning for every component product), relational (all the components in the monitored stack are interrelated) , and actionable (rather than simply regularly polling different monitored components to obtain information about them, it is sent by the component products only when action is required) service-oriented monitoring model for all JES components thereby providing information about the performance and status of the various JES component products in the stack.

JESMF follows a Common Information Model (CIM) - based monitoring model, improves usability, and adds support for existing CP metrics, and SNMP exposure of well known MIBs. It also implements Lockhart based JES Monitoring Console (JESMC) application on top of the infrastructure software that allows users to view monitored data.

# JESMC

JES Monitoring Console helps visualizing JES component products monitored data exposed by JESMF. JESMC provides JES end users with a clear view of how their JES stack is performing and to provide them the means to take action appropriately. The data collected by JESMF is made available to such industry standard management applications as HP Open View, Tivoli or Solaris Management Agent.

The JESMC is a Lockhart based GUI. When the Monitoring Console and Lockhart are installed on your system, you can view the console by entering the following URL into the browser:

`https://host_name:6789` where the `host_name` represents the name or IP address of the machine on which Lockhart and JESMF are running. Port number 6789 is the officially registered port reserved for Lockhart.

The Monitoring Console exposes the following through various data views:

- Hierarchy of the monitored objects implemented in the JES Core Information Base.
- Usage relationships (a component product may depend on other product services).
- Operational status and availability status of the monitored objects.
- Performance statistics of the monitored objects.

The JESMC operates within real-time (given the inherent refresh restrictions of a browser). As a host is added to the scope of the console, installed and running JES component products are automatically discovered. As Component Products are discovered, the monitored objects that represent these products are visualized within the console. When the user refreshes the browser, the current JES installation and associated statistics are reflected in the console.

The Monitoring Console allows the user to set threshold values on any monitored object by creating a Monitoring Rule. An alarm is generated when a threshold value is crossed.

The JESMC is not a management console. No administrative commands, other than creating monitoring rules are possible.

In particular, the Console does not:

- Permit administrative operations on the monitored products.
- Provide a link to the admin console of the product.
- Provide the ability to browse or analyze any of the gathered job results.

The Monitoring Console is composed of two elements:

- A web application integrated into the Lockhart console and based on the Lockhart components.
- The client API runs on the same host as the JESMF Master Agent and provides a simple and efficient way to access JESMF data. Both the elements communicate using the JMX technology. The client API is an internal API that is not exposed to users of the console or users of JESMF. The client API is a Mbean instantiated in the Mbean server in which the MA is running.

Users expect JES to deliver an integrated management and monitoring system, which deliver the services they need in a reliable and predictable manner. The JESMF is a major step toward achieving this.

The benefits expected from JESMF architecture are:

- **Dynamic Adaptability:** JMX monitoring agents can support new monitoring service types for dynamically created services.
- **Easy to Integrate with legacy systems** through protocol adapters and connectors.
- **Low Cost:** JMX manageability can be readily implemented in software products without having to pay the learning curve cost of complex system management technologies and information models.

For more information JESMF and JESMC, refer the *Sun Java Enterprise System 5 Monitoring Guide* in <http://docs.sun.com>.

## Common Agent Container

Common agent container is a stand alone Java program that implements a container for Java management applications. Common agent container provides a management infrastructure designed for JMX and JDMK - based management functionality. Common agent container is designed to accept modules that extend an existing JMX or JDMK agent with new managed objects, connectors, or new functionality.

The Portal Server JESMF integration adds a common agent container module for itself.

## Portal Server:JESMF Detailed Mapping

This section provides a list of attributes, which are used to map the Portal Server and JESMF.

## CMM\_InstalledProduct:PortalServer

TABLE 72-1 CMM\_InstalledProduct:PortalServer

Attributes	<p>Portal Sever : JESMF integration common agent container module is identified as <code>com.sun.cmm.ps</code> module. JESMF uses common agent container module parameters from the deployment descriptor to initialize Portal Server : JESMF integration module. The module MBean ObjectName is of the form, <code>com.sun.cacao:type=ModuleManager,instance=com.sun.cmm.ps</code> with attributes initialized from common agent container deployment descriptor.</p>	<p>An example of Portal Server : JESMF integration common agent container deployment descriptor is listed below.</p> <pre> &lt;xml version='1.0' encoding='UTF-8'?&gt; &lt;!DOCTYPE module SYSTEM "urn:sun:n1:cacao:module:dtd:1_1"&gt; &lt;module name="com.sun.cmm.ps" initial-administrative-state ="UNLOCKED"&gt;   &lt;description&gt;     PS:JESMF Integration Module   &lt;/description&gt;    &lt;module-dependencies&gt;     &lt;instance-dependency dependency-type=       "STRONG-DEPENDENCY"&gt;       &lt;name&gt;         com.sun.mfwk       &lt;/name&gt;     &lt;/instance-dependency&gt;   &lt;/module-dependencies&gt;    &lt;module-class&gt;     com.sun.mfwk.MfCPmodule   &lt;/module-class&gt;    &lt;cacao-version-supported&gt;     1.1   &lt;/cacao-version-supported&gt;    &lt;heap-requirements-kilobytes&gt;     100   &lt;/heap-requirements-kilobytes&gt;    &lt;parameters&gt;     &lt;parameter&gt;       &lt;param-name&gt;ProductName&lt;/param-name&gt;       &lt;param-value&gt;Portal Server&lt;/param-value&gt;     &lt;/parameter&gt;      &lt;parameter&gt;       &lt;param-name&gt;InstalledLocation&lt;/param-name&gt;       &lt;param-value&gt;/opt/SUNWportal&lt;/param-value&gt;     &lt;/parameter&gt;      &lt;parameter&gt;       &lt;param-name&gt;BuildNumber&lt;/param-name&gt;       &lt;param-value&gt;7.1&lt;/param-value&gt;     &lt;/parameter&gt; </pre>
------------	--	--

TABLE 72-1 CMM\_InstalledProduct:PortalServer (Continued)

---

```

<parameter>
  <param-name>PatchID</param-name>
  <param-value>7.1</param-value>
</parameter>

<parameter>
  <param-name>RevisionNumber
  </param-name>
  <param-value>7.1</param-value>
</parameter>

<parameter>
  <param-name>InstallDate</param-name>
  <param-value>1160635803475
  </param-value>
</parameter>

<parameter>
  <param-name>Caption</param-name>
  <param-value>Portal Server
  </param-value>
</parameter>

<parameter>
  <param-name>Description</param-name>
  <param-value>Portal Server
  </param-value>
</parameter>

<parameter>
  <param-name>ProductIdentifyingNumber
  </param-name>
  <param-value>7.1</param-value>
</parameter>

<parameter>
  <param-name>ProductVendor
  </param-name>
  <param-value>SMI</param-value>
</parameter>

<parameter>
  <param-name>ProductVersion
  </param-name>
  <param-value>7.1</param-value>
</parameter>
</parameters>
</module>

```

---

**TABLE 72-1** CMM\_InstalledProduct:PortalServer (Continued)

Settings	None
Statistics	None
States	None
Logs	None
Events	None
Relations	CMM_RunningApplication AGGREG with  CMM_ApplicationSystem:PortalInstance as parent [1..*]

## CMM\_ApplicationSystem:PortalServerInstance

**TABLE 72-2** CMM\_ApplicationSystem:PortalServerInstance

Attributes	Name=<portalDomain>.<portalID>. portalDomain is a String that identifies PortalDomain. Default value is <instanceID>  default.  portalId is a String that represents Portals context URI. Default value is portal1.  instanceId is a String that is typically set as WebContainer instance name, For example, <machineFQDN>-<httpPort>
Settings	None
Statistics	None
States	None
Logs	None
Events	None



TABLE 72-2 CMM\_ApplicationSystem:PortalServerInstance (Continued)

Relations	<ul style="list-style-type: none"> <li>■ CMM_RunningApplication AGGREG with CMM_InstalledProduct:PortalServer as child [1..*]</li> <li>■ CMM_ApplicationSystemLogicalComponent COMP with CMM_LogicalComponent:Desktop as parent [1]</li> <li>■ CMM_HostedService COMP with CMM_Service:Desktop as parent [1]</li> <li>■ CMM_HostedAccessPoint COMP with CMM_ServiceAccessURI:DesktopServiceAccessURI as parent [1]</li> <li>■ CMM_ResourceOfSystem COMP with CMM_ThreadPool:GetterPool as parent [1]</li> <li>■ CMM_ResourceOfSystem COMP with CMM_ThreadPool:CallerPool as parent [1]</li> </ul>
-----------	---

## Portal Server:Desktop Detailed Mapping

### CMM\_LogicalComponent:Desktop

TABLE 72-3 CMM\_LogicalComponent:Desktop

Attributes	Name=<portalDomain>.<portalID>. <instanceID>.Desktop	portalDomain, portalId, instanceId values are same as those that constitute CMM_ApplicationSystem:PortalInstance Name attribute.
Settings	None	
Statistics	None	

TABLE 72-3 CMM\_LogicalComponent:Desktop (Continued)

States	None
Logs	None
Events	None
Relations	<ul style="list-style-type: none"> <li>■ CMM_LogicalComponentHierarchy COMP with CMM_LogicalComponent:Provider as parent [1..*]</li> <li>■ CMM_ServiceLogicalComponent ASSOC with CMM_Service:Desktop as parent [1]</li> <li>■ CMM_ApplicationSystemLogicalComponent COMP with CMM_LogicalComponent:Desktop as child [1]</li> </ul>

## CMM\_Service:Desktop

TABLE 72-4 CMM\_Service:Desktop

Attributes	Name=<portalDomain>.<portalID>.<instanceId>.Desktop	portalDomain, portalId, instanceId values are same as those that constitute CMM_ApplicationSystem:PortalInstance Name attribute.
	Description=Aggregating content from content Providers and presenting it to client	
Settings	None	
Statistics	None	
States	None	
Logs	None	
Events	None	

TABLE 72-4 CMM\_Service:Desktop (Continued)

Relations	<ul style="list-style-type: none"> <li>■ CMM_ServiceComponent COMP with 7 (sub-)services: CMM_Service:- Content, Edit, Process, Logout, PreLogin, LocalAuth, Exception as parent [1 each]</li> <li>■ CMM_ServiceResource ASSOC with CMM_ThreadPool:GetterPool as child [1]</li> <li>■ CMM_ServiceResource ASSOC with CMM_ThreadPool:CallerPool as child [1]</li> <li>■ CMM_HostedService COMP with CMM_ApplicationSystem:PortalInstance as child [1]</li> <li>■ CMM_ServiceLogicalComponent ASSOC with CMM_LogicalComponent:Desktop as child [1]</li> </ul>
-----------	---

## CMM\_ServiceAccessURI:DesktopServiceAccessURI

TABLE 72-5 CMM\_ServiceAccessURI:DesktopServiceAccessURI

Attributes	isSecured=false typically.	If HTTP protocol is used to access Desktop service, isSecured=false. If HTTPS protocol is used to access Desktop service, isSecured=true.
Settings	CMM_ServiceAccessURISetting	
Statistics	CMM_ServiceAccessURISats	Counts number of requests served at PS:Desktop URI with service and resident times.
States	None	
Logs	None	
Events	None	
Relations	<ul style="list-style-type: none"> <li>■ CMM_HostedAccessPoint COMP with CMM_ApplicationSystem:PortalInstance as child [1]</li> </ul>	

## CMM\_LogicalComponent:Provider

TABLE 72-6 CMM\_LogicalComponent:Provider

Attributes	Name=<fullyQualifiedChannelName>	Fully Qualified Channel Name (FQCN) is typically of the form, name=<channelName>, class=<fullyQualifiedProviderClassName>.
Settings	None	
Statistics	None	
States	None	
Logs	None	
Events	None	
Relations	<ul style="list-style-type: none"> <li>■ CMM_ServiceLogicalComponent ASSOC with 3 CMM_Services CMM_Service:- Content, Edit, Process as parent [1 each]</li> <li>■ CMM_LogicalComponentHierarchy COMP with CMM_LogicalComponent:Desktop as child [1..*]</li> </ul>	

## CMM\_Service:- Content, Edit, Process, Logout, PreLogin, LocalAuth, Exception

TABLE 72-7 CMM\_Service:- Content, Edit, Process, Logout, PreLogin, LocalAuth, Exception

Attributes	Name=<portalDomain>.<portalID>.<instanceId>.<Desktop.<Content>   Edit   Process   Logout   PreLogin   LocalAuth   Exception>	portalDomain, portalId, instanceId values are same as those that constitute CMM_ApplicationSystem:PortalInstance Name attribute.
	Description=<Presenting Content View to client   Presenting Edit View to client   Completing Edit processing   ...>	
Settings	CMM_PSRollingAvgServiceSetting	

**TABLE 72-7** CMM\_Service:- Content, Edit, Process, Logout, PreLogin, LocalAuth, Exception  
(Continued)

Statistics	<i>CMM_PSRollingAvgServiceStats</i>
States	None
Logs	None
Events	None
Relations	<ul style="list-style-type: none"> <li>■ CMM_ServiceComponent COMP for the 7 (sub-)services: CMM_Service:- Content, Edit, Process, Logout, PreLogin, LocalAuth, Exception as child with CMM_Service:Desktop [1 each]</li> <li>■ CMM_ServiceServiceDependency ASSOC with CMM_Services: Content, Edit, Process as parent offered by CMM_LogicalComponent:Provider respectively [1 each].</li> </ul>

## CMM\_Service:- Content, Edit, Process

**TABLE 72-8** CMM\_Service:- Content, Edit, Process

Attributes	Name=<fullyQualifiedChannelName>	FQCN is typically of the form name=<channelName>, class=<fullyQualifiedProviderClassName>.
	Description=<Presenting Content View of a channel   Presenting Edit View of a channel   Completing Edit processing of a channel>	
Settings	<i>CMM_PSServiceSetting</i>	
Statistics	<i>CMM_PSServiceStats</i>	
States	None	
Logs	None	
Events	None	

**TABLE 72-8** CMM\_Service:- Content, Edit, Process (Continued)

Relations	<ul style="list-style-type: none"> <li>■ CMM_ServiceLogicalComponent ASSOC of the 3 CMM_Services CMM_Service:- Content, Edit, Process as child to CMM_LogicalComponent:Provider [1 each]</li> <li>■ CMM_ServiceServiceDependency ASSOC with CMM_Services: Content, Edit, Process as child offered by CMM_LogicalComponent:Desktop respectively [1 each].</li> </ul>
-----------	---

## CMM\_ThreadPool:- GetterPool, CallerPool

**TABLE 72-9** CMM\_ThreadPool:- GetterPool, CallerPool

Attributes	Name=<GetterPool   CallerPool>  Description=<Thread pool to invoke Provider.init() method   Thread pool to invoke Provider.getContent() method>	
Settings	CMM_ThreadPoolSetting	MaxThreadPoolSize  MinThreadPoolSize  ThreadPoolId=<PoolClassName>  Max -, Min- ThreadPoolSize are picked up from the /var/opt/SUNWportal/portals/<portalID>/ config/desktopconfig.properties file.  PoolClassName identifies if it is a GetterPool or CallerPool. GetterPool threads are used to initialize Portal Server Desktop providers and CallerPool threads are used to run provider's getContent() method.
Statistics	CMM_ThreadPoolStats	How many threads from the thread pool are present currently.
States	None	
Logs	None	

TABLE 72-9 CMM\_ThreadPool:- GetterPool, CallerPool (Continued)

Events	None
Relations	<ul style="list-style-type: none"> <li>■ CMM_ServiceResource ASSOC with CMM_Service:Desktop as parent [1 each]</li> <li>■ CMM_ResourceOfSystem COMP with CMM_ApplicationSystem:PortalInstance as child [1 each]</li> </ul>

## JESMF Extensions for Portal Server

The JESMF extensions for Portal Server product are:

### CMM\_PSServiceStats

CMM\_PSRollingAvgServiceStats extends CMM\_ServiceStats with the following two additional metrics:

TABLE 72-10 CMM\_PSRollingAvgServiceStats

RollingAvgServiceTime	Rolling average of ServiceTime property of CMM_ServiceStats.
RollingAvgResidentTime	Rolling average of ResidentTime property of CMM_ServiceStats.

### CMM\_PSServiceSetting

CMM\_PSRollingAvgServiceStats works with CMM\_PSRollingAvgServiceSetting. CMM\_PSRollingAvgServiceSetting extends CMM\_ServiceSetting with the following two additional properties that help managing CMM\_PSRollingAvgServiceStats:

TABLE 72-11 CMM\_PSRollingAvgServiceStats

LowerBound	Minimum number of service requests after which rolling average metric comes into effect. Prior to LowerBound number of service requests, rolling average is a simple average. Default value is 100.
UpperBound	Number of service requests whose service or resident time is considered when calculating rolling average. Default value is 1000.

## Installing and Configuring Portal Server:JESMF Integration

In JES 5, the installation and configuration of Portal Server:JESMF integration occurs by default. If you need to upgrade the system to JES 5, then you require to install and configure. This section explains about how you can install and configure Portal Server:JESMF Integration.

PS:JESMF Integration depends on three JES shared components:

- Jdmk5.1-patch1
- Cacao2.0
- JESMF2.0

These shared components are part of JES 5. JES 4 supported earlier versions such as Jdmk5.1, Cacao1.1, and JESMF1.0. Use these components if you use JES 4. While upgrade, you need to first upgrade the shared components to JES 5 and then enable Portal Server:JESMF Integration.

For upgrade or migration, when Cacao2.0 or JESMF2.0 is not installed, then Portal Server:JESMF Integration should be disabled. This means that no configuration should happen for Portal Server:JESMF Integration.

These are high level requirements to configure Portal Server:JESMF Integration:

1. Copy `/opt/SUNWportal/template/jesmf/com.sun.cmm.ps.xml` to `/tmp`, and change the following tokens:
  - `%ps.product.location%`
  - `%PRODUCT_VERSION%`
  - `%ps.install.datetime%`
2. Register JESMF 2.0 module using `com.sun.cmm.ps.xml` from `<baseDir>/template/jesmf/com.sun.cmm.ps.xml`. Following tokens should be replaced with correct values at config time in this file:
3. `InstallLocation` should be set to install directory of portal. For example, on Solaris with default install location as `/opt`.
4. `InstalledDate` should be set to current system time of configuration. For example, `System.currentTimeMillis()` value should be dumped as token value.

---

**Note** – You need to use “`<jesmfInstallDir>/bin/mfwksetup -r <path_to_com.sun.cmm.ps.xml>`” to register Portal Server:JESMF common agent container module. This should happen only for the first instance for any portal getting created on this Operating System (OS) node. There is only one Portal Server:JESMF common agent container 2.0 module on one OS node.

---

5. Check the web containers configuration to see if this jar (and `jdmkrt.jar`) is present in the server classpath. If it is not present, add it to web containers server classpath.



6. Copy `desktopmfwk.properties` to  
`/var/opt/SUNWportal/portals/<portalId>config/<instanceId> form`  
`<baseDir>/template/jesmf/desktopmfwk.properties`.
7. In the `desktopmfwk.properties` file, replace the `%PS_DIR%` token with install location. For example, `/opt/SUNWportal` on Solaris.

The Portal Server:JESMF Integration unconfiguration happens when a portal server instance is deleted from an OS-node. These are high level requirements to unconfigure Portal Server:JESMF Integration:

- Whenever a portal server instance (portal web-application) is deleted on an OS-node, delete `/var/opt/SUNWportal/portals/<portalId>/config/<instanceId>/desktopmfwk.properties`.
- If this is the last portal server instance of the last portal being deleted on this OS-node, unregister PS:JESMF Integration common agent container module using `"<jesmfInstallDir>/bin/mfwksetup -u <path_to_com.sun.cmm.ps.xml>"`.

## Enabling or Disabling Portal Server:JESMF Integration

This section explains how to enable or disable Portal Server:JESMF integration.

### ▼ To Enable Portal Server:JESMF Integration:

- 1 Set the `com.sun.portal.mfwk.disable` property in `desktopmfwk.properties` to `"False"`.
- 2 Restart Portal Server to make the changes take effect.

### ▼ To Disable Portal Server:JESMF Integration:

- 1 Set the `com.sun.portal.mfwk.disable` property in `desktopmfwk.properties` to `"True"`.
- 2 Restart Portal Server to make the changes take effect.



# Index

---

## A

- AccountPreferencesContainer
  - JSPs, 215, 224
- AddressBookJSPProvider, JSPs, 182
- AddressBookProvider
  - properties,display profile, 159
  - tags, 399
  - template files, 421
- AppProvider
  - properties,display profile, 160
  - tag, 400
  - template files, 422
- ASCTabContainer, JSPs, 223
- Attributes
  - desktop
    - dynamic attributes, 42, 45
    - global attributes, 40-42, 42
    - introduction, 39
  - search
    - categories (*See also* Categories,search), 79, 81
    - database (*See also* Database,search), 69, 84
    - reports (*See also* Reports,search), 87, 89
    - robot (*See also* Robot,search), 53, 84
  - subscriptions
    - organization, 92
    - scheduling profiler, 98
    - service, 91
    - users, 94, 97
- Available channels list, 144

## B

- BookmarkProvider
  - properties,display profile, 161
  - template files, 422
- Boolean property, 139

## C

- CalendarJSPProvider, JSPs, 183
- CalendarProvider
  - properties,display profile, 161
  - tags, 401
  - template files, 423
- Categories,search
  - classification rules attributes, 80
  - manage categories page attributes, 79
- Collection property, 139
- Common tags, 393
- CommunityDiscussionProvider, JSPs, 183
- CommunitySearchProvider, JSPs, 184
- CommunitySubscriptionsProvider, JSPs, 185
- Conditional properties, 140
- Context setup tags, 353

## D

- Database,search
  - analysis page attributes, 76
  - import agent attributes, 70, 71
  - import schedule attributes, 84

- Database,search (*Continued*)
    - management attributes, 70
    - resource descriptions attributes, 72,74
    - schema edit attributes, 74
  - DefaultChannel, JSPs, 186
  - Desktop
    - attributes,introduction, 39
    - configuration properties file, 101
    - dynamic attributes, 42,45
    - global attributes, 39,42
    - schema file, 110
    - service definition, 110
    - SunPortalDesktopService, 109
    - tag library (*See* Tag libraries), 345
    - template files (*See* Template files,desktop), 415
    - theme (*See* Theme,desktop), 293
  - desktopconfig.properties file
    - overview, 101
    - parameters, 101
  - Developer Sample
    - desktops, 273
    - installation directories, 272
  - Developer sample, JSPs, 229
  - Disabling Portal Server:JESMF Integration, 448
  - Discussion Provider JSPs, JSPs, 217
  - DiscussionLite, JSPs, 186
  - DiscussionProvider, JSPs, 186
  - DiscussionsProvider, properties,display profile, 156
  - Display Profile
    - administering, 114
    - channel definition, 113
    - document (*See* Document,display profile), 113
    - DTD, 110
    - introduction, 113
    - objects (*See* Objects,display profile), 119
    - properties (*See* Properties,display profile), 116
    - property types (*See* Properties,display profile), 139
    - provider definition, 113,121
    - XML files (*See* XML files,display profile), 137
  - Document,display profile
    - channel definitions, 119
    - dynamic document, 116
    - global document, 116
  - Document,display profile (*Continued*)
    - merging
      - fuse, 130
      - process, 129
      - remove, 130
      - replace, 130
      - semantics, 127
    - organization, 117
    - priority, 123
    - provider definitions, 121
    - role, 117
    - storing, 116
    - structure, 115
    - suborganization, 117
  - Document Type Definition (*See* DTD), 109
  - dp-anon.xml file description, 137
  - dp-org.xml file description, 137
  - dp-providers.xml file description, 137
  - DTD
    - Display Profile, 110
    - files, 110
    - Rewriter, 110
    - Service Management Services, 110
  - Dynamic attributes
    - desktop, 42,45
    - subscriptions, 97
- E**
- Enabling Portal Server:JESMF Integration, 448
  - Error template files, 425
- F**
- filter.conf file
    - editing, 302
    - location, 301
  - FrameTabContainer
    - architecture, 288
    - description, 273
    - JSPs, 230,243,289
    - sample desktop, 285
  - FullPageContainer, JSPs, 225

**G**

Global attributes,desktop, 39, 42

GlobalThemes

display profile definition, 293

glossary of terms, 296

properties, 294, 296

**H**

HTML

content

Rewriter applet rules, 262

Rewriter attribute rules, 260

Rewriter form rules, 262

Rewriter JavaScript token rules, 261

template tags

common tags, 393

introduction, 345

kinds, 346

**I**

IFrameProvider, JSPs, 188

IMProvider

JSPs, 188

properties,display profile, 162

tags in `im.tld`, 411

Integer property, 140

**J**

JavaScript

DHTML

parameters, 265

variables, 264

Dynamic JavaScript (DJS)

parameters, 266

variables, 264

EXPRESSION

parameters, 265

variables, 263

function parameters, 265

JavaScript (*Continued*)

system variables, 264

URL

parameters, 265

variables, 263

JavaServer Pages (*See* JSPs), 175

JESMC, 436

JESMF, 435

JESMF Extensions, 447

JSPContentContainer, JSPs, 189

JSPContentContainer JSPs, 230

JSPCreateChannelContainer JSPs, 231

JSPCustomThemeContainer JSPs, 231

JSPDynamicSingleContainer, JSPs, 189

JSPDynamicSingleContainer JSPs, 232, 244

JSPEditContainer

JSPs, 189, 219, 225

JSPEditContainer JSPs, 232

JSPFrameCustomTableContainerProvider

description, 292

JSPs, 190, 232

JSPLayoutContainer, JSPs, 191

JSPLayoutContainer JSPs, 233

JSPMenuContainerProvider

JSPs, 191, 219

JSPNativeContainer, JSPs, 192

JSPPopupContainer JSPs, 233

JSPPresetThemeContainer JSPs, 233

JSPProvider

JSPs, 192

properties,display profile, 149

JSPRenderingContainerProvider, JSPs, 192

JSPRenderingContentChannel, JSPs, 193

JSPRenderingEditContainer, JSPs, 193

JSPRenderingErrorChannel, JSPs, 193

JSPRenderingLayoutChannel, JSPs, 194

JSPRenderingProvider, JSPs, 194

JSPs

community sample directory, 215, 216, 217, 218

developer sample anonymous desktop, 243, 247

enterprise sample directory, 223, 224, 225, 226, 227

in default directory, 181

in developersample directory, 229

in sampleportal directory, 242

JSPs (*Continued*)

- installation location, 175
- introduction, 175
- lookup scenario, 176, 179
- miscellaneous
  - in developersample directory, 236
- tag library
  - attributes, 347
  - exceptions, 348
  - groups, 346
  - introduction, 346
  - purpose, 346
  - return values, 348
- welcome sample directory, 249
- JSPs in the default directory, 198
- JSPSingleContainer JSPs, 234
- JSPSingleContainerProvider
  - description, 143
  - JSPs, 194
- JSPSingleRenderingContainerProvider, JSPs, 195
- JSPTabContainer
  - architecture, 278
  - description, 273
  - JSPs, 234, 244, 279
  - sample desktop, 275
- JSPTabContainerProvider
  - description, 143
  - JSPs, 195
- JSPTabCustomTableContainer JSPs, 234
- JSPTabCustomTableContainerProvider
  - description, 291
  - JSPs, 196
- JSPTableContainer
  - architecture, 282
  - description, 273
  - JSPs, 283
  - sample desktop, 280
- JSPTableContainer JSPs, 235
- JSPTableContainerProvider
  - description, 143
  - JSPs, 197, 218, 225, 244

**L**

- LoginProvider
  - properties,display profile, 163
  - tags, 410
  - template files, 426
- LotusNotesAddressBookJSPProvider, JSPs, 199
- LotusNotesAddressBookProvider
  - properties,display profile, 159
  - tags, 399
  - template files, 421
- LotusNotesCalendarJSPProvider, JSPs, 200
- LotusNotesCalendarProvider
  - properties,display profile, 161
  - tags, 401
  - template files, 423
- LotusNotesMailJSPProvider, JSPs, 200

**M**

- MAAddressBookJSPProvider, JSPs, 200
- MailJSPProvider, JSPs, 204
- MAMailJSPProvider, JSPs, 201
- Miscellaneous, JSPs, 227
- MSExchangeAddressBookJSPProvider, JSPs, 202
- MSExchangeAddressBookProvider
  - properties,display profile, 159
  - tags, 399
  - template files, 421
- MSExchangeCalendarJSPProvider, JSPs, 203
- MSExchangeCalendarProvider
  - properties,display profile, 161
  - tags, 401
  - template files, 423
- MSExchangeMailJSPProvider, JSPs, 203
- MyFrontPageFramePanelContainer parent, 291
- MyFrontPageTabPanelContainer parent, 291
- MyFrontPageTemplatePanelContainer
  - parent, 291
  - template files, 429

**N**

## Normal tags

- desktop.tld, 357
- desktopProviderContext.tld, 361, 369
- desktopTab.tld, 371
- desktopTable.tld, 373
- desktopTheme.tld, 375
- search.tld, 391
- searchContext Java API, 379

NotesProvider, 164

**O**

## Objects,display profile

- channel object, 119
- container object, 120
- lookup, 122
- property object, 121
- provider object, 121
- types, 119

OnlineHelpContainer, JSPs, 226

**P**

## PagePreferencesContainer

- JSPs, 217, 226

Portal Server:JESMF Detailed Mapping, 437

## PredefinedFrontPageFramePanelContainerProvider

- description, 292
- JSPs, 237, 245

## PredefinedFrontPageTabPanelContainerProvider

- description, 292
- JSPs, 238, 246

## PredefinedFrontPageTemplatePanelContainerProvider

- description, 292
- template files, 429, 431

## PredefinedSamplesFramePanelContainerProvider

- description, 292
- JSPs, 239, 246

## PredefinedSamplesTabPanelContainerProvider

- description, 292
- JSPs, 240, 247

## PredefinedSamplesTemplatePanelContainerProvider

- description, 292

## PredefinedToolsTemplatePanelContainerProvider

- description, 292

## Properties,display profile

- changing values, 116
- channel, 139, 171
- container, 139
  - common tab container properties, 147
  - common table container properties, 145, 147
  - general properties, 148
  - required properties, 144
- global properties, 138, 141
- provider
  - common properties, 167
  - container, 143
  - content providers, 159
  - introduction, 138, 167
  - leaf building-block providers, 149
  - service providers, 155

## type

- Boolean, 139
- Collection, 139
- ConditionalProperty, 140
- Integer, 140
- Reference, 140
- String, 140

**R**

RedirectChannel, JSPs, 249

Reference property, 140

RenderingWrappingProvider, JSPs, 204

## Reports,search

- excluded URLs attributes, 87
- popular searches attributes, 89
- robot advance reports attributes, 88
- view log files attributes, 88

## Rewriter

- default ruleset, 110
- introduction, 253, 259

## rules

- HTML content (*See also* HTML), 260, 262
- JavaScript content (*See also* JavaScript), 263, 266

Rewriter, rules (*Continued*)

- XML content (*See also* XML), 266
- ruleset DTD, 110
- service, 47
- service definition, 110
- SunPortalRewriterService, 110
- supported URLs, 257
- robot.conf file
  - editing, 303
  - user-modifiable parameters, 303, 307
- Robot,search, 66
  - advance reports attributes, 88
  - Application Functions (*See* RAFs), 301
  - crawling attributes, 65
  - edit a filter attributes, 59
  - filter
    - Data stage, 316
    - Enumeration stage, 317
    - Generation stage, 317
    - MetaData stage, 316
    - Setup stage, 315, 321
    - Shutdown stage, 317
  - filtering process, 309, 311
  - manage sites attributes, 55, 58
  - overview panel attributes, 55
  - RAFs
    - filter directives, 312
    - introduction, 301, 315
    - parameter, 319
  - schedule attributes, 83
  - simulator properties, 67
  - site probe attributes, 67
- Root Level attributes, subscriptions, 91

**S**

- SampleJSP JSP, 241
- SamplesFramePanelContainer parent, 291
- SampleSimpleWebService, JSPs, 204
- SampleSimpleWebService JSP, 242
- SampleSimpleWebServiceConfigurable, JSPs, 205
- SamplesTabPanelContainer parent, 291
- SamplesTemplatePanelContainer template files, 432

- Schema file
  - Desktop, 110
  - search, 110
- search.conf file
  - overview, 105
  - parameters, 105
- Search Provider JSPs, JSPs, 216
- Search Server Attributes
  - search
    - introduction, 51-52
- SearchProvider
  - categories attributes (*See also* Categories,search), 79, 81
  - configuration properties file, 105
  - database attributes (*See also* Database,search), 69, 84
  - JSPs, 205
  - properties,display profile, 155
  - reports attributes (*See also* Reports,search), 87
  - Robot Application Functions (RAFs) (*See* Robot,search), 301
  - robot attributes (*See also* Robot,search), 53, 84
  - schema file, 110
  - server attributes (*See also* Server,search), 51
  - service definition, 110
  - tag library
    - exceptions, 349
    - introduction, 349
    - post-search tags, 387
    - pre-search tags, 381
    - searchContext Java API tags, 379
    - tag attributes, 349
    - tags to execute the search, 385
    - types, 349
- SearchProvider/channel, reports attributes (*See also* Reports,search), 89
- SearchTabPanelContainer parent, 291
- Selected channels list, 144
- Server,search
  - log settings attributes, 89
  - settings attributes, 51
- Service Management Services DTD, 110
- SimpleWebServiceConfigurableProvider
  - JSPs, 207



- SimpleWebServiceConfigurableProvider (*Continued*)  
   properties,display profile, 165
- SimpleWebServiceProvider  
   JSPs, 207  
   properties,display profile, 164
- String property, 140
- SubscriptionsProvider  
   JSPs, 208  
   organization attributes, 92  
   properties,display profile, 158  
   scheduling profiler attributes, 98  
   service attributes, 91  
   user attributes, 94,97
- SunPortalDesktopService, 109
- SunPortalRewriterService, 110
- SunPortalSearchService, 109
- T**
- TabJSPEditContainer, JSPs, 208
- Tag libraries  
   context setup tags, 353  
   descriptors (*See also* TLDs), 350  
   hierarchy, 350  
   Instant Messaging tags (*See* IMProvider), 411  
   JSPs (*See* JSPs), 346  
   normal tags (*See* Normal tags), 357  
   search (*See* SearchProvider), 349  
   template tags (*See* HTML), 345  
   types of tags, 345  
   validator tags, 355
- Template files,desktop  
   file lookup, 416  
   in default directory, 426  
   in developersample directory, 433  
   installation location, 415  
   introduction, 415
- Theme,desktop  
   GlobalThemes (*See also* GlobalThemes), 293,297  
   introduction, 293
- TLDs  
   desktop.tld, 350,357  
   desktopContainerProviderContext.tld, 350  
   desktopProviderContext.tld, 350,361,369
- TLDs (*Continued*)  
   desktopSingle.tld, 350  
   desktopTab.tld, 351,371  
   desktopTable.tld, 351,373  
   desktopTheme.tld, 351,375  
   im.tld, 351,411  
   jr.tld, 351  
   jx.tld, 351  
   search.tld, 351,381,385,387,391
- ToolsTemplatePanelContainer template files, 433
- U**
- URLScrapperProvider  
   limitations, 254  
   properties,display profile, 150
- UWCAddressBookJSPProvider, JSPs, 209
- UWCCalendarJSPProvider, JSPs, 209
- UWCMailJSPProvider, JSPs, 210
- V**
- Validator tags, 355
- vxml, JSPs, 213
- W**
- WirelessJSPClientConfigContainer, JSPs, 210
- WirelessJSPContentChannel, JSPs, 210
- WirelessJSPDesktopProvider, JSPs, 211
- WirelessJSPLayoutChannel, JSPs, 212
- X**
- XML  
   content  
     Rewriter rules for attributes, 267  
     Rewriter rules for tag text, 266  
   files,display profile  
     dp-anon.xml, 137  
     dp-org.xml, 137

XML, files,display profile (*Continued*)

    dp-providers.xml, 137

XMLProvider, 151