

Oracle® Fusion Middleware

WebCenter WLST Command Reference



12c (12.2.1.4.0)

E95398-03

July 2021

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Copyright © 2007, 2021, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Documentation Accessibility	xii
Related Documents	xii
Conventions	xii

1 Introduction and Roadmap

1.1 Document Scope and Audience	1-1
1.2 Guide to This Document	1-1
1.3 Related Documentation	1-1
1.4 What's New and Changed in This Guide	1-2
1.4.1 New and Changed Features for 12c Release (12.2.1.3.0)	1-2

2 WebCenter Portal Custom WLST Commands

2.1 Overview of Oracle WebCenter Portal WLST Command Categories	2-2
2.2 General	2-3
2.2.1 deleteConnection	2-3
2.2.2 setWebCenterServiceFrameworkConfig	2-4
2.2.3 getWebCenterServiceFrameworkConfig	2-5
2.2.4 webcenterErrorOccurred	2-6
2.2.5 getWebCenterConnectionTypes	2-6
2.2.6 deleteWebCenterPortalCoherenceCache	2-7
2.3 Analytics	2-7
2.3.1 createAnalyticsCollectorConnection	2-8
2.3.2 setAnalyticsCollectorConnection	2-10
2.3.3 listAnalyticsCollectorConnections	2-11
2.3.4 setDefaultAnalyticsCollectorConnection	2-12
2.3.5 listDefaultAnalyticsCollectorConnection	2-13
2.3.6 setAnalyticsCollectorConfig	2-14
2.3.7 listAnalyticsCollectorConfig	2-15
2.3.8 listAnalyticsEventTypes	2-16
2.4 Activity Stream	2-17

2.4.1	archiveASByDate	2-17
2.4.2	archiveASByDeletedObjects	2-18
2.4.3	archiveASByClosedSpaces	2-19
2.4.4	archiveASByInactiveSpaces	2-19
2.4.5	restoreASByDate	2-20
2.4.6	truncateASArchive	2-21
2.4.7	archiveASBySpace	2-22
2.4.8	archiveASAllSpaces	2-23
2.4.9	archiveASByUser	2-23
2.4.10	archiveASAllUsers	2-24
2.4.11	archiveASByDeletedActors	2-25
2.4.12	showASStatistics	2-25
2.5	BPEL Server Connection	2-27
2.5.1	createBPELConnection	2-27
2.5.2	setBPELConnection	2-29
2.5.3	listBPELConnections	2-30
2.6	Content Repository	2-31
2.6.1	createContentServerConnection	2-32
2.6.2	setContentServerConnection	2-36
2.6.3	listContentServerConnections	2-40
2.6.4	listContentServerProperties	2-41
2.6.5	setContentServerProperties	2-41
2.6.6	deleteContentServerProperties	2-43
2.6.7	deleteContentServerConnection	2-43
2.6.8	exportFoldersGData	2-44
2.6.9	migrateFoldersGDataToFrameworkFolders	2-45
2.7	Discussions and Announcements	2-47
2.7.1	createDiscussionForumConnection	2-48
2.7.2	setDiscussionForumConnection	2-50
2.7.3	setDiscussionForumConnectionProperty	2-53
2.7.4	deleteDiscussionForumConnectionProperty	2-54
2.7.5	listDefaultDiscussionForumConnection	2-55
2.7.6	listDiscussionForumConnections	2-56
2.7.7	setDefaultDiscussionForumConnection	2-57
2.7.8	setDiscussionForumServiceProperty	2-58
2.7.9	removeDiscussionForumServiceProperty	2-59
2.7.10	listDiscussionForumServiceProperties	2-60
2.7.11	setAnnouncementServiceProperty	2-61
2.7.12	removeAnnouncementServiceProperty	2-62
2.7.13	listAnnouncementServiceProperties	2-63
2.7.14	addDiscussionsServerAdmin	2-63

2.7.15	addDiscussionsCategoryAdmin	2-64
2.7.16	setDiscussionsServerProperty	2-65
2.7.17	getDiscussionsServerProperty	2-66
2.7.18	removeDiscussionsServerProperty	2-67
2.8	External Applications	2-68
2.8.1	createExtAppConnection	2-69
2.8.2	setExtAppConnection	2-70
2.8.3	listExtAppConnections	2-71
2.8.4	addExtAppField	2-72
2.8.5	setExtAppField	2-73
2.8.6	removeExtAppField	2-74
2.8.7	addExtAppCredential	2-75
2.8.8	setExtAppCredential	2-76
2.8.9	removeExtAppCredential	2-77
2.9	Instant Messaging and Presence	2-78
2.9.1	createIMPConnection	2-79
2.9.2	setIMPConnection	2-81
2.9.3	setIMPConnectionProperty	2-82
2.9.4	deleteIMPConnectionProperty	2-84
2.9.5	listIMPAdapters	2-85
2.9.6	listIMPConnections	2-85
2.9.7	listDefaultIMPConnection	2-86
2.9.8	setDefaultIMPConnection	2-87
2.9.9	setIMPServiceProperty	2-88
2.9.10	removeIMPServiceProperty	2-89
2.9.11	listIMPServiceProperties	2-90
2.9.12	createIMPExtAppConnection	2-91
2.10	Mail	2-92
2.10.1	createMailConnection	2-92
2.10.2	setMailConnection	2-94
2.10.3	setMailConnectionProperty	2-97
2.10.4	deleteMailConnectionProperty	2-98
2.10.5	listMailConnections	2-98
2.10.6	listDefaultMailConnection	2-99
2.10.7	setDefaultMailConnection	2-100
2.10.8	setMailServiceProperty	2-101
2.10.9	removeMailServiceProperty	2-102
2.10.10	listMailServiceProperties	2-103
2.10.11	createMailExtApp	2-104
2.11	Notifications	2-105
2.11.1	setNotificationsConfig	2-105

2.11.2	getNotificationsConfig	2-107
2.12	People Connections	2-107
2.12.1	startSyncProfiles	2-108
2.12.2	stopSyncProfiles	2-108
2.12.3	isSyncProfilesRunning	2-109
2.12.4	syncProfile	2-110
2.12.5	setProfileConfig	2-110
2.12.6	getProfileConfig	2-111
2.12.7	listProfileConfig	2-112
2.12.8	setProfilePhotoSync	2-113
2.13	Personal Events	2-113
2.13.1	createPersonalEventConnection	2-114
2.13.2	setPersonalEventConnection	2-115
2.13.3	listPersonalEventConnections	2-116
2.14	Portlet Producers	2-117
2.14.1	registerWSRPProducer	2-118
2.14.2	setWSRPProducer	2-122
2.14.3	listWSRPProducers	2-126
2.14.4	deregisterWSRPProducer	2-128
2.14.5	listWSRPProducerRegistrationProperties	2-128
2.14.6	listWSRPProducerUserCategories	2-129
2.14.7	mapWSRPProducerUserCategory	2-130
2.14.8	registerPDKJavaProducer	2-131
2.14.9	setPDKJavaProducer	2-133
2.14.10	deregisterPDKJavaProducer	2-135
2.14.11	listPDKJavaProducers	2-136
2.14.12	refreshProducer	2-137
2.14.13	listPortletClientConfig	2-137
2.14.14	setPortletClientConfig	2-138
2.14.15	getPortletClientConfig	2-140
2.14.16	registerOOTBProducers	2-141
2.14.17	deregisterOOTBProducers	2-142
2.14.18	registerSampleProducers	2-143
2.14.19	deregisterSampleProducers	2-144
2.15	Proxy Server	2-145
2.15.1	getWebCenterProxyConfig	2-145
2.15.2	setWebCenterProxyConfig	2-146
2.15.3	unsetWebCenterProxyConfig	2-146
2.16	Search - Elasticsearch	2-147
2.16.1	createSearchConnection	2-147
2.16.2	setSearchConnection	2-148

2.16.3	listSearchConnections	2-149
2.17	WebCenter Portal Application	2-150
2.17.1	getSpacesWorkflowConnectionName	2-151
2.17.2	setSpacesWorkflowConnectionName	2-151
2.18	Identity Store	2-152
2.18.1	setWebCenterIdStoreSearchConfig	2-152
2.18.2	listWebCenterIdStoreSearchConfig	2-153
2.19	Lifecycle	2-154
2.19.1	deployWebCenterPortal	2-155
2.19.2	propagateWebCenterPortal	2-158
2.19.3	exportWebCenterPortals	2-161
2.19.4	exportWebCenterPortalTemplates	2-163
2.19.5	importWebCenterPortals	2-165
2.19.6	listWebCenterPortalArchive	2-168
2.19.7	exportWebCenterPortalConnections	2-168
2.19.8	importWebCenterPortalConnections	2-171
2.19.9	setSpaceState	2-173
2.19.10	exportWebCenterResource	2-174
2.19.11	importWebCenterResource	2-176
2.19.12	importWebCenterTranslations	2-178
2.19.13	exportWebCenterApplication	2-178
2.19.14	importWebCenterApplication	2-179
2.19.15	exportPortletClientMetadata	2-180
2.19.16	importPortletClientMetadata	2-181
2.19.17	showProducerImportFailures	2-182
2.19.18	retryAllFailedProducerImports	2-182
2.19.19	cloneWebCenterManagedServer	2-183
2.20	Upgrade	2-184
2.20.1	upgradeWebCenterPortal	2-184
2.20.2	listDeprecatedFeaturesUsage	2-187

3 Oracle WebCenter Content Custom WLST Commands

3.1	Overview of WLST WebCenter Content Command Categories	3-1
3.2	WLST WebCenter Content Help	3-2
3.3	Getter and Setter Methods Implementation	3-2
3.4	Server Configuration Commands	3-3
3.4.1	getUCMHttpServerAddress	3-3
3.4.2	getUCMServerPort	3-4
3.4.3	setUCMServerPort	3-4
3.4.4	getUCMIpAddressFilter	3-5

3.4.5	setUCMIpAddressFilter	3-5
3.4.6	getUCMUseSSL	3-6
3.5	Email Configuration Commands	3-6
3.5.1	getUCMMailServer	3-6
3.5.2	setUCMMailServer	3-7
3.5.3	getUCMSmtpPort	3-7
3.5.4	getUCMSysAdminAddress	3-8
3.5.5	setUCMSysAdminAddress	3-8
3.6	System Status Commands	3-8
3.6.1	getUCMCSVersion	3-9
3.6.2	getUCMServerUptime	3-9
3.7	General Configuration Commands	3-10
3.7.1	getUCMOverRideFormat	3-10
3.7.2	setUCMOverRideFormat	3-11
3.7.3	getUCMDownloadApplet	3-11
3.7.4	setUCMDownloadApplet	3-12
3.7.5	getUCMMultiUpload	3-12
3.7.6	setUCMMultiUpload	3-12
3.7.7	getUCMUseAccounts	3-13
3.7.8	setUCMUseAccounts	3-13
3.7.9	getUCMIsAutoNumber	3-14
3.7.10	setUCMIsAutoNumber	3-14
3.7.11	getUCMAutoNumberPrefix	3-15
3.7.12	setUCMAutoNumberPrefix	3-15
3.7.13	getUCMMajorRevLabelSeq	3-15
3.7.14	setUCMMajorRevLabelSeq	3-16
3.7.15	getUCMMinorRevLabelSeq	3-16
3.7.16	setUCMMinorRevLabelSeq	3-17
3.7.17	getUCMJspServerEnabled	3-17
3.7.18	setUCMJspServerEnabled	3-18
3.7.19	getUCMJspEnabledGroups	3-18
3.7.20	setUCMJspEnabledGroups	3-18
3.8	Content Security Configuration Commands	3-19
3.8.1	getUCMCopyAccess	3-19
3.8.2	setUCMCopyAccess	3-20
3.8.3	getUCMExclusiveCheckout	3-20
3.8.4	setUCMExclusiveCheckout	3-21
3.8.5	getUCMAuthorDelete	3-21
3.8.6	setUCMAuthorDelete	3-21
3.8.7	getUCMShowOnlyKnownAccounts	3-22
3.8.8	setUCMShowOnlyKnownAccounts	3-22

3.9	Component Manager Configuration Commands	3-23
3.9.1	getUCMComponentStatus	3-23
3.9.2	setUCMComponentStatus	3-24
3.9.3	installUCMComponent	3-24
3.9.4	uninstallUCMComponent	3-24
3.9.5	downloadUCMComponent	3-25
3.9.6	getUCMComponentConfig	3-25
3.9.7	updateUCMComponentConfig	3-26
3.10	Records Management Configuration Commands	3-26
3.10.1	getRMLevel	3-27
3.10.2	getRMConfigurationLevel	3-27
3.10.3	getRMFeatures	3-28
3.10.4	getRMDispositionActions	3-28
3.10.5	rmUpdate	3-29
3.10.6	addOutgoingProvider	3-30
3.10.7	registerSource	3-30
3.11	User Interface Commands	3-31
3.11.1	displayWccAdfConfig	3-32
3.11.2	updateWccAdfConfig	3-33
3.12	User Interface Connection Commands	3-36
3.12.1	createRIDCCConnection	3-37
3.12.2	updateRIDCCConnection	3-37
3.12.3	listRIDCCConnections	3-39
3.12.4	deleteRIDCCConnection	3-39
3.12.5	displayRIDCCConnection	3-40

4 Oracle WebCenter Content: Imaging Custom WLST Commands

4.1	Overview of Imaging WLST Command Categories	4-1
4.2	Diagnostic Commands	4-1
4.2.1	clearIPMWorkflowFaults	4-2
4.2.2	clearIPMWorkflowFaultsByDocId	4-2
4.2.3	listIPMWorkflowFaults	4-3
4.2.4	repairIPMWorkflowFaults	4-3
4.2.5	repairIPMWorkflowFaultsByDocId	4-4
4.2.6	sumIPMWorkflowFaults	4-5
4.2.7	resetIPMDMSMetrics	4-5
4.3	Imaging Configuration Commands	4-5
4.3.1	createIPMConnection	4-6
4.3.2	modifyIPMConnection	4-7
4.3.3	getIPMConfig	4-7

4.3.4	getIPMParamLimits	4-8
4.3.5	grantIPMCredAccess	4-8
4.3.6	importIPMApplication	4-9
4.3.7	importIPMInput	4-11
4.3.8	importIPMSearch	4-12
4.3.9	listIPMConfig	4-13
4.3.10	listIPMDefinitions	4-14
4.3.11	listIPMExportFile	4-14
4.3.12	refreshIPMSecurity	4-15
4.3.13	setIPMConfig	4-15
4.3.14	submitIPMToWorkflow	4-15

5 Oracle WebCenter Enterprise Capture Custom WLST Commands

5.1	Overview of WLST Oracle WebCenter Enterprise Capture Command Categories	5-1
5.2	Configuration Commands	5-2
5.2.1	listWorkspaces	5-3
5.2.2	listBatches	5-4
5.2.3	exportBatch	5-4
5.2.4	exportWorkspace	5-5
5.2.5	exportEWSEmailMessage	5-6
5.2.6	exportIMAPEmailMessage	5-7
5.2.7	importWorkspace	5-7
5.2.8	unlockBatch	5-8
5.2.9	listLockedBatches	5-9
5.2.10	listCaptureConfig	5-9
5.2.11	getCaptureConfig	5-10
5.2.12	setCaptureConfig	5-10
5.2.13	scanForClientBundles	5-10
5.2.14	deleteBatches	5-11
5.2.15	deleteBundle	5-11
5.2.16	setObjectProperty	5-12
5.2.17	getObjectProperty	5-13
5.2.18	setObjectCredentials	5-14
5.2.19	updateScript	5-15
5.2.20	cloneWorkspaceFromID	5-16
5.2.21	cloneWorkspaceFromFile	5-17
5.2.22	listImportHATokens	5-17
5.2.23	deleteImportHAToken	5-18

A Capture Object Properties

A.1	Client Profile Object Properties	A-1
A.2	Import Processor Job Object Properties	A-3
A.2.1	General Import Job Properties	A-3
A.2.2	Folder Import Job Specific Properties	A-5
A.2.3	List File Import Job Specific Properties	A-6
A.2.4	Email Job Specific Properties	A-6
A.3	Recognition Processor Job Object Properties	A-8
A.3.1	General Recognition Job Properties	A-8
A.3.2	BarcodeDefinition Class Properties	A-11
A.3.3	DocumentDefinition Class Properties	A-12
A.3.4	RecognitionJobField Class Properties	A-12
A.3.5	SeparatorDefinition Class Properties	A-12
A.4	Commit Profile Object Properties	A-12
A.4.1	General Commit Profile Properties	A-13
A.4.2	Text File Commit Profile Specific Properties	A-14
A.4.3	Webcenter Content Commit Profile Specific Properties	A-15
A.4.4	ContentAttributeMappingInfo Class Properties	A-16
A.4.5	CaptureToContentFieldMapping Class Properties	A-17
A.4.6	ContentFieldNameInfo Class Properties	A-17
A.4.7	Webcenter Imaging Commit Profile Specific Properties	A-17
A.4.8	PDF Searchable Document Output Specific Properties	A-17
A.4.9	DOCS Commit Driver Properties	A-18
A.5	Document Conversion Job Object Properties	A-19

Preface

This document describes the custom WebLogic Scripting Tool (WLST) commands that can be used to manage WebCenter Portal, WebCenter Content, and WebCenter Enterprise Capture.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

Documentation for Oracle WebCenter Portal Cloud is available at <http://docs.oracle.com/cloud/latest/webcenter-portal-cloud/index.html>.

Documentation for Oracle WebCenter Content is available from the Oracle Help Center at [Oracle WebCenter Content](#) page.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction and Roadmap

This section describes the contents and organization of this guide—*WebCenter WLST Command Reference*.

- [Document Scope and Audience](#)
- [Guide to This Document](#)

1.1 Document Scope and Audience

This document describes the custom WebLogic Scripting Tool (WLST) commands that can be used to manage WebCenter Portal, WebCenter Content, and WebCenter Enterprise Capture.



Note:

Custom WLST commands for a given Oracle Fusion Middleware component are available for use only if the component is installed in the `ORACLE_HOME` directory.

This document is written for WebLogic Server administrators and operators who deploy Java EE applications using the Java Platform, Enterprise Edition (Java EE) from Oracle. It is assumed that readers are familiar with Web technologies and the operating system and platform where WebLogic Server is installed.

1.2 Guide to This Document

This document is organized as follows:

- This chapter, "Introduction and Roadmap," introduces the organization of this guide and lists related documentation.
- [WebCenter Portal Custom WLST Commands](#), provides detailed descriptions of the custom WLST commands for Oracle WebCenter Portal.
- [Oracle WebCenter Content Custom WLST Commands](#), provides detailed descriptions of the custom WLST commands for Oracle WebCenter Content.
- [Oracle WebCenter Content: Imaging Custom WLST Commands](#), provides detailed descriptions of the custom WLST commands for Oracle WebCenter Content: Imaging.
- [Oracle WebCenter Enterprise Capture Custom WLST Commands](#), provides detailed descriptions of the custom WLST commands for Oracle WebCenter Capture.

1.3 Related Documentation

For information about how to use the WebLogic Scripting Tool, refer to *Understanding the WebLogic Scripting Tool*.

1.4 What's New and Changed in This Guide

This section introduces the new and changed Custom WLST Commands for WebCenter Content, WebCenter Enterprise Capture, and WebCenter Portal. This section contains the following topic:

- [New and Changed Features for 12c Release \(12.2.1.3.0\)](#).

1.4.1 New and Changed Features for 12c Release (12.2.1.3.0)

- Added a new section with custom WLST commands for [Oracle WebCenter Content: Imaging](#).
- Added the WLST command for creating an Elasticsearch connection for Oracle WebCenter Portal. See [Search - Elasticsearch](#).

2

WebCenter Portal Custom WLST Commands

This chapter describes WebLogic Scripting Tool (WLST) commands for Oracle WebCenter Portal. These commands enable you to configure WebCenter Portal and components from the command-line.



Note:

To use these commands, you must invoke WLST from the Oracle home (*ORACLE_HOME*) directory. See Running Oracle WebLogic Scripting Tool (WLST) Commands in *Administering Oracle WebCenter Portal*.

Most configuration changes made using these WLST commands are only effective after you restart the Managed Server on which WebCenter Portal is deployed. There are some exceptions, including WLST commands for [External Applications](#), [Portlet Producers](#), and [Lifecycle](#).

For additional details about Oracle WebCenter Portal configuration, see Introduction to Administration for Oracle WebCenter Portal in *Administering Oracle WebCenter Portal*.



Note:

Oracle WebCenter Portal has deprecated the support for Jive features (announcements and discussions). If you have upgraded from a prior release to Release 12c (12.2.1.4.0), Jive features remain available in your upgraded instance but Oracle support is not provided for these features. In the next release, Jive features will not be available even in the upgraded instances

Oracle WebCenter Portal's custom WLST commands are described in the following sections:

- [Overview of Oracle WebCenter Portal WLST Command Categories](#)
- [General](#)
- [Analytics](#)
- [Activity Stream](#)
- [BPEL Server Connection](#)
- [Content Repository](#)
- [Discussions and Announcements](#)
- [External Applications](#)
- [Instant Messaging and Presence](#)
- [Mail](#)

- [Notifications](#)
- [People Connections](#)
- [Personal Events](#)
- [Portlet Producers](#)
- [Proxy Server](#)
- [Search - Elasticsearch](#)
- [WebCenter Portal Application](#)
- [Identity Store](#)
- [Lifecycle](#)
- [Upgrade](#)

2.1 Overview of Oracle WebCenter Portal WLST Command Categories

Oracle WebCenter Portal's WLST commands are grouped into the several categories (Table 2-1).

Most configuration changes made using these WLST commands are only effective after you restart the Managed Server on which WebCenter Portal is deployed. The only exceptions are the [External Applications](#), [Portlet Producers](#), and [Lifecycle](#) WLST commands.

Table 2-1 WLST Command Categories

Command Category	Description
General	Manage connections for WebCenter Portal.
Analytics	Manage analytics collector connections and configure the analytics collector.
Activity Stream	Archive and restore activity stream data.
Content Repository	Manage content repository connections and configure document services.
Discussions and Announcements	Manage discussions server connections and configure discussions and announcements.
External Applications	Manage external application connections.
Instant Messaging and Presence	Manage instant messaging and presence server connections and configure instant messaging and presence.
Mail	Manage mail server connections and configure mail.
Notifications	Manage settings for notifications.
People Connections	Manage profile information.
Personal Events	Manage personal event server connections.
Portlet Producers	Manage portlet producers.
Proxy Server	Manage proxy settings for RSS and activity stream.
Search - Elasticsearch	Manage Elasticsearch (ES) connections.
BPEL Server Connection	Manage BPEL server connections.

Table 2-1 (Cont.) WLST Command Categories

Command Category	Description
WebCenter Portal Application	Manage WebCenter Portal workflow settings and portal metadata.
Identity Store	Configure options while searching an application's identity store.
Lifecycle	Export and import application metadata, individual portals, portal templates, portal assets, and portlet producer metadata.
Upgrade	Upgrade from a previous Oracle WebCenter Portal release.

2.2 General

Use the General commands, listed in [Table 2-2](#), to manage connections, and perform other general tasks.

Configuration changes made using these WLST commands are only effective after restarting the Managed Server on which WebCenter Portal is deployed.

Table 2-2 General WLST Commands

Use This Command...	To...	Use with WLST...
deleteConnection	Delete any connection.	Online
getWebCenterConnectionTypes	List all connection types.	Online
setWebCenterServiceFrameworkConfig	Set WebCenter Portal Service Framework configuration properties.	Online
getWebCenterServiceFrameworkConfig	Return WebCenter Portal Framework configuration properties.	Online
webcenterErrorOccurred	Return status information for the last Oracle WebCenter Portal command executed.	Online
deleteWebCenterPortalCoherenceCache	Clears the coherence cache in a coherence cluster used in the WebCenter Portal application.	Online

2.2.1 deleteConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Deletes a named connection currently configured for WebCenter Portal.

If you use `deleteConnection` to delete a WSRP or PDK-Java producer connection (instead of using `deregisterWSRPProducer` or `deregisterPDKJavaProducer`), unused secondary connections will remain, which you might want to remove. For example, when you delete a WSRP producer connection, its associated web service connection remains and when you delete a PDK-Java producer connection, its associated URL connection remains.

Syntax

```
deleteConnection(appName, name, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>name</i>	Connection name.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example deletes a connection configured for WebCenter Portal (webcenter):

```
wls:/weblogic/serverConfig> deleteConnection(appName='webcenter',
name='MyConnection')
```

2.2.2 setWebCenterServiceFrameworkConfig

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Sets configuration properties for Oracle WebCenter Portal's services framework, such as the Resource Action Handler class and display as popup properties.

Syntax

```
setWebCenterServiceFrameworkConfig(appName, [resourceActionHandlerClassName,
resourceActionHandlerDisplayInPopup, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>resourceActionHandlerClassName</i>	Optional. Class used by the Service Framework Resource Action Handler.
<i>resourceActionHandlerDisplayInPopup</i>	Optional. Indicates whether the Resource Action Handler displays resources in a popup or inline. Valid options are 1 (true) and 0 (false).
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example sets the WebCenter Portal Service Framework Resource Action Handler class to `my.company.ResourceActionHandler`:

```
wls:/wc_domain/domainRuntime> setWebCenterServiceFrameworkConfig(appName='webcenter',
resourceActionHandlerClassName='my.company.ResourceActionHandler')
```

Successfully set the WebCenter Portal service framework configuration.
Resource Action Handler class: `my.company.ResourceActionHandler`
To effect connection changes, you must restart the managed server on which the application is deployed.

The following example sets only the WebCenter Portal Service Framework Resource Action Handler display as popup value to 1 (true):

```
wls:/wc_domain/domainRuntime>
setWebCenterServiceFrameworkConfig(appName='webcenter',
resourceActionHandlerDisplayInPopup=1)
```

Successfully set the WebCenter Portal service framework configuration.
Resource Action Handler Display In Popup: `true`
To effect connection changes, you must restart the managed server on which the application is deployed.

2.2.3 getWebCenterServiceFrameworkConfig

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Returns WebCenter Portal Service Framework configuration property settings, such as:

- `resourceActionHandlerClassName`: Class currently used by the WebCenter Portal Service Framework Resource Action Handler
- `resourceActionHandlerDisplayInPopup`: Indicates whether the Resource Action Handler displays resources in a popup or inline. Valid options are 1 (true) and 0 (false).

Syntax

```
getWebCenterServiceFrameworkConfig(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example returns the service framework resource action handler class and display as popup properties, for WebCenter Portal (`webcenter`):

```
wls:/weblogic/
serverConfig>getWebCenterServiceFrameworkConfig(appName='webcenter')
Resource Action Handler Class: my.company.ResourceActionHandler
Resource Action Handler Display In Popup: true
```

2.2.4 webcenterErrorOccurred

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Returns the status of last WebCenter Portal command executed.

Use the `webcenterErrorOccurred` command to determine the status of the last WebCenter Portal command executed. The command returns 1 if an error occurred or 0 otherwise.

Syntax

```
webcenterErrorOccurred ()
```

Example

The following example returns 1 if an error occurred:

```
wls:/mydomain/serverConfig> webcenterErrorOccurred()
```

2.2.5 getWebCenterConnectionTypes

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists all the Oracle WebCenter Portal connection types.

Syntax

```
getWebCenterConnectionTypes(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example returns connection types for WebCenter Portal (`webcenter`):

```
wls:/mydomain/serverConfig>getWebCenterConnectionTypes(appName='webcenter')
```

2.2.6 deleteWebCenterPortalCoherenceCache

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Clears the coherence cache in a coherence cluster used in the WebCenter Portal application.

Syntax

```
deleteWebCenterPortalCoherenceCache(appName, server)
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.

Example

The following example clears the coherence cache for WebCenter Portal (`webcenter`):

```
wls:/weblogic/serverConfig> deleteWebCenterPortalCoherenceCache (appName='webcenter' ,  
server='WC_Portal')
```

2.3 Analytics

This section includes the commands to manage analytics collector connections and configure analytics collector for WebCenter Portal.

Analytics Collector Connections

Use the commands listed in [Table 2-3](#) to manage analytics collector connections for WebCenter Portal. Events raised in these portal applications using OpenUsage APIs can be sent to an analytics collector for use by analytics.

Connection configuration changes made using these Oracle WebCenter Portal WLST commands are only effective after you restart the Managed Server on which the application is deployed.

Table 2-3 Analytics Collector Connection WLST Commands

Use this command...	To...	Use with WLST...
createAnalyticsCollectorConnection	Create a connection to an analytics collector for WebCenter Portal.	Online

Table 2-3 (Cont.) Analytics Collector Connection WLST Commands

Use this command...	To...	Use with WLST...
setAnalyticsCollectorConnection	Edit an existing analytics collector connection.	Online
listAnalyticsCollectorConnections	List all of the analytics collector connections that are configured for WebCenter Portal.	Online
setDefaultAnalyticsCollectorConnection	Specify the default (or active) analytics collector connection for WebCenter Portal.	Online
listDefaultAnalyticsCollectorConnection	Return connection details for the analytics collector being used by WebCenter Portal.	Online

Analytics Collector and Cluster Configuration

Use the commands listed in [Table 2-4](#) to configure event collection properties for the analytics collector that is deployed on the `WC_Uilities` managed server.

If you reconfigure the analytics collector or set up clustering, you must restart the managed server on which the analytic collector is deployed (`WC_Uilities`).

Table 2-4 Analytics Collector Configuration WLST Commands

Use this command...	To...	Use with WLST...
setAnalyticsCollectorConfig	Set analytics collector options, and cluster options if operating a clustered environment.	Online
listAnalyticsCollectorConfig	Return analytics collector settings.	Online
listAnalyticsEventTypes	List events currently registered with the analytics collector.	Online

2.3.1 createAnalyticsCollectorConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Creates a connection to an analytics collector for a named application.

Events raised in WebCenter Portal using OpenUsage APIs can be sent to an analytics collector for use by analytics.

While you can register multiple analytics collector connections for an application, only one analytics collector connection is used - the default (or active) connection where `default=1`.

Syntax

```
createAnalyticsCollectorConnection(appName, connectionName, [isUnicast,
collectorhost, clusterName, collectorPort, isEnabled, timeout, default, server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the name is always <code>webcenter</code> .
<i>connectionName</i>	Connection name. The name must be unique across all connection types within application.
<i>isUnicast</i>	Optional. Specifies whether events are sent to a clustered analytics collector in multicast mode or whether a single analytics collector using unicast communication is required. Valid values are 1 (true) and 0 (false). The default value is 1 (unicast).
<i>collectorHost</i>	Optional. Host name where the analytics collector is running. The default value is <code>localhost</code> . Only required for unicast communication, that is, where <code>isUnicast=1</code> .
<i>clusterName</i>	Optional. Name of the cluster where a clustered analytics collector is running. Only required for multicast communication, that is, where <code>isUnicast=0</code> .
<i>collectorPort</i>	Optional. Port on which the analytics collector listens for events. The default value is 31314.
<i>isEnabled</i>	Optional. Specifies whether to send analytics events raised using OpenUsage APIs to the analytics collector. Valid values 1 (true) and 0 (false). The default value is 0. Analytics events are sent to the analytics collector when <code>isEnabled=1</code> and <code>default=1</code> .
<i>timeout</i>	Optional. Length of time (in seconds) to wait for a response from the analytics collector. Default value is 30. Only required for multicast communication, that is, where <code>isUnicast=0</code> .
<i>default</i>	Optional. Indicates whether this connection is the default (or active) analytics collector connection for the application. Valid values are 1 (true) and 0 (false). When set to 1, the application sends events on this connection. When set to 0, the connection is not used. The default for this argument is 0. While you can register multiple analytics collector connections for an application, only one connection is used by analytics—the default (or active) connection.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example creates a connection named `MyAnalyticsCollector` for WebCenter Portal (`webcenter`). Events are sent to a single analytics collector using *unicast* communication:

```
wls:/weblogic/serverConfig>createAnalyticsCollectorConnection(appName='webcenter',
connectionName='MyAnalyticsCollector', isUnicast=1,
collectorHost='myhost.com', collectorPort=31314, isEnabled=1, timeout=30, default=1)
```

The following example creates a connection named `MyAnalyticsCollector` for WebCenter Portal. Events are sent to a clustered analytics collector in *multicast* mode

```
wls:/weblogic/
serverConfig>createAnalyticsCollectorConnection(appName='webcenter',
connectionName='MyAnalyticsCollector', isUnicast=0, clusterName='collector-
cluster',
ccollectorPort=31314, isEnabled=1, timeout=30, default=1)
```

2.3.2 setAnalyticsCollectorConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Edits an existing analytics collector connection for a named application.

Events raised in WebCenter Portal using OpenUsage APIs can be sent to an analytics collector for use by analytics.

While you can register multiple analytics collector connections for an application, only one analytics collector connection is used - the default (or active) connection.

Syntax

```
setAnalyticsCollectorConnection(appName, connectionName, [isUnicast,
collectorHost, clusterName, collectorPort, isEnabled, timeout, default, server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the name is always <code>webcenter</code> .
<i>connectionName</i>	Connection name. The name must be unique across all connection types within application.
<i>isUnicast</i>	Optional. Specifies whether events are sent to a clustered analytics collector in multicast mode or whether a single analytics collector using unicast communication is required. Valid values are 1 (true) and 0 (false). The default value is 1 (unicast).
<i>collectorHost</i>	Optional. Host name where the analytics collector is running. The default value is <code>localhost</code> . Only required for unicast communication, that is, where <code>isUnicast=1</code> .
<i>clusterName</i>	Optional. Name of the cluster where a clustered analytics collector is running. Only required for multicast communication, that is, where <code>isUnicast=0</code> .
<i>collectorPort</i>	Optional. Port on which the analytics collector listens for events. The default value is 31314.

Argument	Definition
<i>isEnabled</i>	Optional. Specifies whether to send analytics events raised using OpenUsage APIs to the analytics collector. Valid values are 1 (true) and 0 (false). The default value is <i>false</i> . Analytics events are sent to the analytics collector when <i>isEnabled=1</i> and <i>default=1</i> .
<i>timeout</i>	Optional. Length of time (in seconds) to wait for a response from the analytics collector. Default value is 30. Only required for multicast communication, that is, where <i>isUnicast=0</i> .
<i>default</i>	Optional. Indicates whether this connection is the default (or active) analytics collector connection for the application. Valid values are 1 (true) and 0 (false). When set to 1, the application sends events on this connection. When set to 0, the connection is not used. The default for this argument is 0. While you can register multiple analytics collector connections for an application, only one connection is used by analytics—the default (or active) connection.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <i>WC_Portal</i> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example edits host and port details for an existing analytics collector connection named *MyAnalyticsCollector*. On this connection, events are sent to a single analytics collector in *unicast* mode:

```
wls:/weblogic/serverConfig>setAnalyticsCollectorConnection(appName='webcenter',
connectionName='MyAnalyticsCollector', collectorHost='myhost.com', collectorPort=31314)
```

The following example edits cluster, port, and timeout details for an existing analytics collector connection named *MyAnalyticsCollector*. On this connection, events are sent to a clustered analytics collector in *multicast* mode:

```
wls:/weblogic/serverConfig>setAnalyticsCollectorConnection(appName='webcenter',
connectionName='MyAnalyticsCollector', clusterName='collector-cluster',
collectorPort=31314, timeout=60)
```

2.3.3 listAnalyticsCollectorConnections

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists connection names and details for all analytics collector connections that are configured for a named application.

Syntax

```
listAnalyticsCollectorConnections(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example lists connection names and details for all the analytics collector connections that are currently configured for WebCenter Portal (`webcenter`):

```
wls:/weblogic/serverConfig>listAnalyticsCollectorConnections(appName='webcenter')
```

```
-----
MyAnalyticsCollector
-----
ClusterName/HostName: localhost
Port: 31314
Timeout: 30
Unicast: 1 (true)
Enabled: 1 (true)

-----
TestAnalyticsCollector
-----
ClusterName/HostName: localhost
Port: 32456
Timeout: 456
Unicast: 1 (true)
Enabled: 1 (true)
-----
```

2.3.4 setDefaultAnalyticsCollectorConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Specifies the default analytics collector connection for a named application.

The default analytics collector connection is used to send events raised in WebCenter Portal using OpenUsage APIs to an analytics collector for use by analytics.

While you can register multiple analytics collector connections for an application, only one analytics collector connection is used-- the default (or active) connection.

Syntax

```
setDefaultAnalyticsCollectorConnection(appName, name, [server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Name of an existing analytics collector connection.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example configures the connection `MyAnalyticsCollector` for events raised in WebCenter Portal (`webcenter`):

```
wls:/weblogic/serverConfig> setDefaultAnalyticsCollectorConnection
(appName='webcenter', name='MyAnalyticsCollector')
```

The following example resets the default connection name:

```
wls:/weblogic/serverConfig> setDefaultAnalyticsCollectorConnection
(appName='webcenter', name='')
```

2.3.5 listDefaultAnalyticsCollectorConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Return details about the analytics collector connection that is currently configured for a named application.

While you can register multiple analytics collector connections for an application, only one analytics collector connection is used—the default (or active) connection.

Syntax

```
listDefaultAnalyticsCollectorConnection(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example returns details about the analytics collector connection that is currently configured for a WebCenter Portal (`webcenter`):

```
wls:/weblogic/
serverConfig>listDefaultAnalyticsCollectorConnection(appName='webcenter')

-----
MyAnalyticsCollector
-----
ClusterName/HostName: localhost
Port: 31314
Timeout: 30
Unicast: 1 (true)
Enabled: 1 (true)
-----
```

2.3.6 setAnalyticsCollectorConfig

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Configure the analytics collector deployed on the `WC_Uutilities` managed server. Additionally, in a clustered environment, use this commands to set cluster settings.

Syntax

```
setAnalyticsCollectorConfig(appName, [collectorHost, defaultPort, maxPort,
broadcastType, clusterEnabled, clusterName, heartbeatFrequency, server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the analytics collector application in which to perform this operation--always <code>analytics-collector</code> .
<i>collectorHost</i>	Optional. Name of the host on which the analytics collector is running. The default value is <code>localhost</code> .
<i>defaultPort</i>	Optional. Default port number on which the analytics collector listens. The default value is <code>31314</code> .
<i>maxPort</i>	Optional. Highest port number that the analytics collector can use when allocating a listener. This property is mostly used in a clustered environment where more than one collector is running in the same box. Each collector listens for incoming UDP messages on a free port within a given port range. The range is from the default port number to the <code>maxPort</code> number.
<i>broadcastType</i>	Optional. Indicates the network channel on which the analytics collector broadcasts a 'heartbeat' to advertise its location to event producers. Valid values are <code>Broadcast</code> and <code>Multicast</code> . <ul style="list-style-type: none"> • <code>Broadcast</code> - use the standard network broadcast channel. • <code>Multicast</code> - use a special fixed multicast address.

Argument	Definition
<i>clusterEnabled</i>	Optional. Indicates whether the analytics collector is deployed in a cluster. Valid values are 1 (true) and 0 (false). If set to 1, <i>clusterName</i> must also be defined.
<i>clusterName</i>	Optional. Name of the analytics collector cluster. Only required when <i>clusterEnabled</i> =1
<i>heartbeatFrequency</i>	Optional. Broadcast analytics collector listening information every 'n' seconds. The default frequency is 10 seconds. The analytics collector periodically broadcasts a 'heartbeat' to advertise its location (<i>hostName</i>). In a clustered environment, WebCenter Portal uses the heartbeat to determine which analytics collectors are available.
<i>server</i>	Optional. Name of the managed server where the analytics collector is deployed. For example, <i>WC_Uilities</i> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example changes the default port to 31315:

```
wls:/weblogic/serverConfig>setAnalyticsCollectorConfig(appName='analytics-collector',
defaultPort=31315)
```

2.3.7 listAnalyticsCollectorConfig

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Returns analytics collector settings.

Syntax

```
listAnalyticsCollectorConfig(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the analytics collector application in which to perform this operation--always <i>analytics-collector</i> .
<i>server</i>	Optional. Name of the managed server where the analytics collector is deployed. For example, <i>WC_Uilities</i> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following command lists current settings for the analytics collector that is configured for an application named `webcenter`:

```
wls:/weblogic/serverConfig>listAnalyticsCollectorConfig(appName='analytics-collector')
```

This is sample output for a standalone analytics collector:

```
CollectorHost = localhost
CollectorDefaultPort = 31314
CollectorMaximumPort = 31314
BroadcastType = Multicast
ClusterEnabled =
ClusterName =
ClusterBroadcastFrequency = 55
```

This is sample output for an analytics collector in a clustered environment:

```
CollectorHost = localhost
CollectorDefaultPort = 31314
CollectorMaximumPort = 31318
BroadcastType = Multicast
ClusterEnabled = 1
ClusterName = myCluster
ClusterBroadcastFrequency = 55
```

2.3.8 listAnalyticsEventTypes

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists all the events currently registered with the analytics collector.

Syntax

```
listAnalyticsEventTypes(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following command lists all the events currently registered with the analytics collector for use by WebCenter Portal (`webcenter`):

```
wls:/weblogic/serverConfig>listAnalyticsEventTypes(appName='webcenter')
```

Sample output:

```
{HTTP://WWW.ORACLE.COM/ANALYTICS/WC}PAGEEDIT
{HTTP://WWW.ORACLE.COM/ANALYTICS/WC}DOCLIB_DOCUMENTCREATE
{HTTP://WWW.ORACLE.COM/ANALYTICS/WC}LOGINS
...
```

2.4 Activity Stream

Use the commands listed in [Table 2-5](#) to archive and restore activity stream data generated for WebCenter Portal.

Configuration changes made using these WLST commands are effective only after your restart the Managed Server on which the application is deployed. For details, see *Oracle Fusion Middleware Administering Oracle WebCenter Portal*.

Table 2-5 Activity Stream WLST Commands

Use this command...	To...	Use with WLST...
archiveASByDate	Archive activity stream data that is older than a specified date.	Online
archiveASByDeletedObjects	Archive activity stream data associated with deleted objects.	Online
archiveASByClosedSpaces	Archive activity stream data associated with portals that are currently closed.	Online
archiveASByInactiveSpaces	Archive activity stream data associated with portals that have been inactive since a specified date.	Online
restoreASByDate	Restore archived activity stream data from a specified date into production tables.	Online
truncateASArchive	Truncates activity stream archive data.	Online
archiveASBySpace	Archive activity stream data associated with a portal.	Online
archiveASAllSpaces	Archive activity stream data associated with all portals.	Online
archiveASByUser	Archive activity stream data associated with a user.	Online
archiveASAllUsers	Archive activity stream data associated with all users.	Online
archiveASByDeletedActors	Archive activity stream data associated with deleted actors.	Online
showASStatistics	Report activity stream statistics.	Online

2.4.1 archiveASByDate

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Archives activity stream data that is older than a specified date.

This command moves data from production tables to archive tables. Exceptions include `WC_ACTOR_DETAIL` and `WC_OBJECT_DETAIL`—data in these tables is copied to archive tables rather than moved.

Rows in `WC_OBJECT_DETAIL` that are not used by any activity element are deleted.

Syntax

```
archiveASByDate(appName, year, month, day, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>year</i>	Year before which to archive activity stream data. For example, 2009.
<i>month</i>	Month before which to archive activity stream data. For example, enter 1 for January, 2 for February, and so on.
<i>day</i>	Day of the month before which to archive activity stream data.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example archives activity stream data that is older than March 1, 2015 for WebCenter Portal (webcenter):

```
wls:/weblogic/serverConfig> archiveASByDate(appName='webcenter', year=2015, month=3, day=1)
```

2.4.2 archiveASByDeletedObjects

Module: Oracle WebCenter Portal

Description

Archives activity stream data associated with deleted objects. This command moves data from production tables to archive tables, except for WC_ACTOR_DETAIL—data in this table is copied to the archive table rather than moved.

Rows in WC_OBJECT_DETAIL that satisfy the criteria (in this case, deleted objects) are deleted.

Syntax

```
archiveASByDeletedObjects(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.

Argument	Definition
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example archives activity stream data associated with deleted objects from WebCenter Portal (*webcenter*):

```
wls:/weblogic/serverConfig> archiveASByDeletedObjects(appName='webcenter')
```

Use with WLST: Online

2.4.3 archiveASByClosedSpaces

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Archives activity stream data associated with portals that are currently closed.

This command moves data from production tables to archive tables, except for *WC_ACTOR_DETAIL*—data in this table is copied to the archive table rather than moved.

Rows in *WC_OBJECT_DETAIL* that satisfy the criteria (in this case, objects involved in activities of portals that are closed) are deleted.

Syntax

```
archiveASByClosedSpaces(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application on which to perform this operation. For WebCenter Portal, the application name is always <i>webcenter</i> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <i>WC_Portal</i> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example archives activity stream data associated with portals that are currently marked as closed in WebCenter Portal (*webcenter*):

```
wls:/weblogic/serverConfig> archiveASByClosedSpaces(appName='webcenter')
```

2.4.4 archiveASByInactiveSpaces

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Archives activity stream data associated with portals that have been inactive since a specified date. An inactive portal is an open or closed portal in which there has been no activity since the specified date.

This command moves data from production tables to archive tables, except for `WC_ACTOR_DETAIL`—data in this table is copied to the archive table rather than moved. Rows in `WC_OBJECT_DETAIL` that satisfy the criteria (in this case, objects involved in activities of portals that have been inactive since the specified date) are deleted.

Syntax

```
archiveASByInactiveSpaces(appName, year, month, day, [server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>year</i>	Year the portal became inactive. For example, 2014.
<i>month</i>	Month the portal became inactive. For example, enter 1 for January, 2 for February, and so on.
<i>day</i>	Day of the month the portal became inactive.
<i>server</i>	Optional. Name of the managed server where WebCenter Portal is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example archives activity stream data associated with portals that have been inactive (no activities have occurred, regardless of open or closed status) since October 1, 2014:

```
wls:/weblogic/serverConfig> archiveASByInactiveSpaces(appName='webcenter',
year=2014, month=10, day=1)
```

2.4.5 restoreASByDate

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Restores archived activity stream data from a specified date into production tables.

This command moves data from archive tables to production tables, except for WC_ACTOR_DETAIL—data in this table is not restored because data is not deleted from this table during the archive process.

Rows that already exist in the production tables are not changed during the restore process.

Syntax

```
restoreASByDate(appName, year, month, day, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>year</i>	Year from which to restore activity stream data. For example, 2014.
<i>month</i>	Month from which to restore activity stream data. For example, enter 1 for January, 2 for February, and so on.
<i>day</i>	Day of the month from which to restore activity stream data.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example restores activity stream data archived since October 1, 2014:

```
wls:/weblogic/serverConfig>restoreASByDate(appName='webcenter', year=2014, month=10, day=1)
```

2.4.6 truncateASArchive

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Truncates activity stream archive data.

Syntax

```
truncateASArchive(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .

Argument	Definition
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example truncates activity stream archive data:

```
wls:/weblogic/serverConfig>truncateASArchive( appName='webcenter' )
```

2.4.7 archiveASBySpace

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Archives activity stream data associated with a named portal, only keeping a fixed number of activities.

This command moves data from production tables to archive tables.

Syntax

```
archiveASBySpace(appName, space, cnt, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>space</i>	Name of the portal whose data you want to archive. For example, MySalesPortal.
<i>cnt</i>	Number of portal activities you want to keep in the production table. For example, 2000.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example keeps the last 2000 activities associated with the portal MySalesPortal and archives the rest:

```
wls:/weblogic/serverConfig> archiveASBySpace( appName='webcenter' ,  
space='MySalesPortal', cnt=2000)
```

2.4.8 archiveASAllSpaces

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Archives activity stream data for all portals, only keeping a fixed number of activities.

This command moves data from production tables to archive tables.

Syntax

```
archiveASAllSpaces(appName, cnt, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application on which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>cnt</i>	Number of portal activities you want to keep in the production table. For example, 2000.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example keeps the last 2000 activities for each portal in production tables and archives the remaining activity stream data:

```
wls:/weblogic/serverConfig> archiveASAllSpaces(appName='webcenter', cnt=2000)
```

2.4.9 archiveASByUser

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Archives activity stream data associated with a single user, only keeping a fixed number of activities.

This command moves data from production tables to archive tables.

Syntax

```
archiveASByUser(appName, actor, cnt, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>actor</i>	Name of the user whose data you want to archive. For example, <code>Monty</code> .
<i>cnt</i>	Number of user activities you want to keep in the production table. For example, <code>2000</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example keeps the last 2000 activities associated with the user `Monty` and archives the rest:

```
wls:/weblogic/serverConfig> archiveASByUser(appName='webcenter', actor='Monty', cnt=2000)
```

2.4.10 archiveASAllUsers

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Archives activity stream data for all users, only keeping a fixed number of activities.

This command moves data from production tables to archive tables.

Syntax

```
archiveASAllUsers(appName, cnt, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>cnt</i>	Number of user activities you want to keep in the production table. For example, <code>2000</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example keeps the last 2000 activities from all users in production tables and archives the remaining activity stream data:

```
wls:/weblogic/serverConfig> archiveASAllUsers(appName='webcenter', cnt=2000)
```

2.4.11 archiveASByDeletedActors

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Archives activity stream data associated with deleted users (actors).

This command moves data from production tables to archive tables. Rows in WC_AS_ACTOR_DETAIL that satisfy the criteria (in this case, deleted actors) are deleted.

Syntax

```
archiveASByDeletedActors(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example archives activity stream data associated with users deleted from WebCenter Portal (`webcenter`):

```
wls:/weblogic/serverConfig> archiveASByDeletedActors(appName='webcenter')
```

2.4.12 showASStatistics

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Reports various activity stream statistics:

- Number of activities for top "N" portals, ordered by activity count
- Number of activities for top "N" users, ordered by activity count
- Number of activities after a specific date
- Number of activities after a specific date for top "N" portals, ordered by activity count
- number of activities after a specified date for top "N" users, ordered by activity count

Syntax

```
showASStatistics(appName, year, month, day, cnt, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>year</i>	Year from which to report activity stream statistics. For example, 2014.
<i>month</i>	Month from which to report activity stream statistics. For example, enter 1 for January, 2 for February, and so on.
<i>day</i>	Day of the month from which to report activity stream statistics.
<i>cnt</i>	Number of portals or users included in the report. For example, 50.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example reports activity stream statistics for the top 50 portals and top 50 users in WebCenter Portal (webcenter) since 6/12/2014:

```
wls:/weblogic/serverConfig> showASStatistics(appName='webcenter', year=2014, month=7, day=18, cnt=3)
```

```
=====
Cutoff Date = 7/18/12 12:23 PM
Top Count = 3

1. Portals with most activity count are:
Portal Finance Count =10
Portal Photography Count =9
Portal Sport Count =7

2. Users with most activity count are:
User Monty Count =30
User Karen Count =20
User Dave Count =10

3. Total Activities after 7/18/14 12:23 PM =80

4. Portals with most activity count after specific date, are:
Portal Photography Count =9
Portal Finance Count =6
Portal Sport Count =6

5. Users with most activity count after specific date, are:
User Monty Count =10
User Dave Count =8
User Josie Count =7
```

2.5 BPEL Server Connection

Use the commands listed in [Table 2-6](#) to manage BPEL server connections to manage membership and other notifications in WebCenter Portal.

Configuration changes made using these WLST commands are only effective after you restart the Managed Server on which the application is deployed. For details, see *Oracle Fusion Middleware Administering Oracle WebCenter Portal*.

Table 2-6 Worklist Commands

Use this command...	To...	Use with WLST...
createBPELConnection	Create a connection to a BPEL server for a named application.	Online
setBPELConnection	Edit an existing BPEL server connection.	Online
listBPELConnections	List all of the BPEL server connections that are configured for a named application.	Online

2.5.1 createBPELConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Creates a connection to a BPEL server for a named application. A BPEL server connection can be used to manage memberships and notifications in WebCenter Portal. WebCenter Portal supports only a single connection to the BPEL server. Multiple BPEL server connections are not supported.

To specify the BPEL server connection that WebCenter Portal uses for its internal workflows, use the `setSpacesWorkflowConnectionName` command. See [setSpacesWorkflowConnectionName](#).

Syntax

```
createBPELConnection(appName, name, url, [policy, recipientKeyAlias, linkUrl, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Connection name. The name must be unique (across all connection types) within the application.
<i>url</i>	URL required to access the BPEL server. Use the format: <code>protocol://host:port</code> The BPEL server URL must be unique within the application.

Argument	Definition
<i>policy</i>	<p>Optional. SAML token policy this connection uses for authentication. Enter any valid policy. Valid values include:</p> <ul style="list-style-type: none"> oracle/wss10_saml_token_client_policy—use to access the BPEL server with the default, non message protected policy. oracle/wss10_saml_token_with_message_protection_client_policy—use to access the BPEL server with a message protected policy. If selected, you must configure keys stores both in your application and in the BPEL application. GPA—use if your environment supports Global Policy Attachments (GPA). <p>If you omit this argument, the connection defaults to oracle/wss10_saml_token_client_policy.</p>
<i>recipientKeyAlias</i>	<p>Optional. Recipient key alias to be used for message protected SAML policy authentication. Only required when the BPEL server connection is using a SAML token policy for authentication and the application's worklist is using multiple BPEL server connections.</p> <p>The default is null.</p> <p>See also "Configuring WS-Security" in <i>Oracle Fusion Middleware Administering Oracle WebCenter Portal</i>.</p>
<i>linkUrl</i>	<p>Optional. URL used to link to the BPEL server. Only required if it is different to the <i>url</i> argument. For example, when SSO or HTTPS is configured.</p> <p>Use the format: <i>protocol://host:port</i></p> <p>The default is null.</p> <p>For performance reasons, in an HTTPS or SSO environment, <i>linkUrl</i> specifies user access to BPEL worklist items, through HTTPS or SSO web servers, whereas <i>url</i> specifies direct access to BPEL web services, without redirection through HTTPS or SSO Web servers.</p>
<i>server</i>	<p>Optional. Name of the managed server where the application is deployed. For example, WC_Portal.</p> <p>Required when applications with the same name are deployed to different servers and also when you have a cluster.</p>
<i>applicationVersion</i>	<p>Optional. Version number of the deployed application. Required if more than one version of the application is deployed.</p>

Examples

The following example creates a connection named `WebCenter Worklist` with the default security policy:

```
wls:/weblogic/serverConfig> createBPELConnection(appName='webcenter',
name='WebCenter Worklist', url='http://myhost.com:8001',
policy='oracle/wss10_saml_token_client_policy')
```

The following example creates a connection that uses a message protected security policy, and defines a specific link URL:

```
wls:/weblogic/serverConfig> createBPELConnection(appName='webcenter',
name='WebCenter Worklist',url='http://myhost.com:8001', policy='oracle/wss10_
saml_token_with_message_protection_client_policy', recipientKeyAlias='myalias',
linkUrl='http://mySSO.com:7777')
```

The following example creates a connection to be used in an environment that supports Global Policy Attachments (GPA):

```
wls:/weblogic/serverConfig> createBPELConnection(appName='webcenter',
name='WebCenter Worklist', url='http://myhost.com:8001', policy='GPA')
```

2.5.2 setBPELConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Edits an existing BPEL server connection.

To specify the BPEL server connection used for WebCenter Portal's internal workflows, use the `setSpacesWorkflowConnectionName` command. See [setSpacesWorkflowConnectionName](#).

Syntax

```
setBPELConnection(appName, name, [url, policy, recipientKeyAlias, linkUrl, server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Existing BPEL server connection name.
<i>url</i>	Optional. URL required to access the BPEL server. Use the format: <code><protocol>://<host>:<port></code> The BPEL server URL must be unique within the application.
<i>policy</i>	Optional. SAML token policy this connection uses for authentication. Enter any valid policy. Valid values include: <ul style="list-style-type: none"> <code>oracle/wss10_saml_token_client_policy</code>—use to access the BPEL server with the default, non message protected policy. <code>oracle/wss10_saml_token_with_message_protection_client_policy</code>—use to access the BPEL server with a message protected policy. If selected, you must configure keys stores both in your application and in the BPEL application. GPA—use if your environment supports Global Policy Attachments (GPA). If you omit this argument, the connection defaults to <code>oracle/wss10_saml_token_client_policy</code> .
<i>recipientKeyAlias</i>	Optional. Recipient key alias to be used for message protected SAML policy authentication. Only required when the BPEL server connection is using a SAML token policy for authentication and the application's worklist is using multiple BPEL server connections. The default is null. See also "Configuring WS-Security" in <i>Administering Oracle WebCenter Portal</i> .

Argument	Definition
<code>linkUrl</code>	Optional. URL used to link to the BPEL server. Only required if it is different to the <code>url</code> argument. For example, when SSO or https is configured. Use the format: <i>protocol://host:port</i> For example, <code>http://mySSO.host.com:7777</code> The default is null. For performance reasons, in an HTTPS or SSO environment, the Link URL specifies user access to BPEL worklist items, through HTTPS or SSO web servers, whereas the BPEL SOAP URL specifies direct access to BPEL web services, without redirection through HTTPS or SSO web servers.
<code>server</code>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<code>applicationVersion</code>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Examples

The following example updates the BPEL server URL, security policy, recipient key alias, and link url for a connection named `WebCenter Worklist`.

```
wls:/weblogic/serverConfig> setBPELConnection(appName='webcenter',
name='WebCenter Worklist',url='http://myhost.com:6666', policy='oracle/wss10_
saml_token_with_message_protection_client_policy', recipientKeyAlias='myalias',
linkUrl='http://mySSO.com:7777')
```

The following example changes the security policy to use Global Policy Attachments (GPA):

```
wls:/weblogic/serverConfig> setBPELConnection(appName='webcenter',
name='WebCenter Worklist', policy='GPA')
```

2.5.3 listBPELConnections

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Without any arguments, this command lists all the BPEL connections that are configured for a named application. All BPEL connections are listed, even connections not currently used.

Syntax

```
listBPELConnections(appName, [verbose, name, server, applicationVersion])
```

Argument	Definition
<code>appName</code>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .

Argument	Definition
<i>verbose</i>	Optional. Displays BPEL server connection details in verbose mode. Valid options are 1 (true) and 0 (false). When set to 1, <code>listBPELConnections</code> lists all of the BPEL server connections that are configured, along with their details. When set to 0, <code>listBPELConnections</code> lists connection names only. This argument defaults to 0. If you set this argument to 0, do not specify the <code>name</code> argument.
<i>name</i>	Optional. Name of an existing BPEL server connection. You can use this argument to view details about a specific connection. To list all the connections, omit the <code>name</code> argument.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Examples

The following example lists the names of all the BPEL server connections that are configured for WebCenter Portal:

```
wls:/weblogic/serverConfig> listBPELConnections(appName='webcenter')
-----
WebCenter Worklist
-----
Human Resources Worklist
-----
```

The following example lists the names and details of all of the BPEL server connections that are configured for WebCenter Portal:

```
wls:/weblogic/serverConfig> listBPELConnections(appName='webcenter', verbose=1)
-----
WebCenter Worklist
-----
Connection Name: WebCenter Worklist
PolicyURI:oracle/wss10_saml_token_client_policy
URL:http://myhost.com:8001
-----
Human Resources Worklist
-----
Connection Name: Human Resources Worklist
PolicyURI:oracle/wss10_saml_token_client_policy
URL:http://myhost.com:8888
-----
```

2.6 Content Repository

Use the commands listed in [Table 2-7](#) to manage content repository connections and configure document services for WebCenter Portal.

Configuration changes made using these WLST commands are only effective after your restart the Managed Server on which the application is deployed. For details, see *Administering Oracle WebCenter Portal*.

Table 2-7 Content Repository WLST Commands

Use this command...	To...	Use with WLST...
createContentServerConnection	Create a connection to an Oracle WebCenter Content repository.	Online
setContentServerConnection	Edit an existing Oracle WebCenter Content repository connection.	Online
listContentServerConnections	List individual or all Oracle WebCenter Content repository connections that are configured for a named application.	Online
listContentServerProperties	List properties for the back-end Content Server that is being used by WebCenter Portal.	Online
setContentServerProperties	Modify properties for the back-end Content Server used by WebCenter Portal.	Online
deleteContentServerProperties	Delete properties for the back-end Content Server used by WebCenter Portal.	Online
deleteContentServerConnection	Delete a connection to an Oracle WebCenter Content repository.	Online
exportFoldersGData	Export Folders_g data to a specified location.	Online
migrateFoldersGDataToFrameworkFolders	Migrate Folders_g data to FrameworkFolders.	Online

2.6.1 createContentServerConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Creates a connection to an Oracle WebCenter Content repository for a named application.

Syntax

```
createContentServerConnection(appName, name, socketType, [url, serverHost,
serverPort, keystoreLocation, keystorePassword, privateKeyAlias,
privateKeyPassword, webContextRoot, clientSecurityPolicy,
cacheInvalidationInterval, binaryCacheMaxEntrySize,
adminUsername, adminPassword, extAppId, timeout, isPrimary, server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>name</i>	Connection name. The name must be unique (across all connection types) within the application.

Argument	Definition
<i>socketType</i>	<p>Specifies whether Oracle WebCenter Content's Content Server connects on the content server listener port or the web server filter, and whether the listener port is SSL enabled.</p> <p>Valid values are <code>socket</code>, <code>web</code>, <code>socketssl</code>, and <code>jaxws</code>. This option has no default.</p> <p>Choose from:</p> <ul style="list-style-type: none"> • socket—Use an <code>intradoc</code> socket connection to connect to the Content Server. The client IP address must be added to the list of authorized addresses in the Content Server. In this case, the client is the machine on which Oracle WebCenter Portal is running. • socketssl—Use an <code>intradoc</code> socket connection to connect to the Content Server that is secured using the SSL protocol. The client's certificates must be imported in the server's trust store for the connection to be allowed. This is the most secure option, and the recommended option whenever identity propagation is required (for example, in WebCenter Portal). • web—Use an HTTP(S) connection to connect to the Content Server. Note that for WebCenter Portal, this option is not suitable for the active connection, that is, the back-end Content Server repository that is being used to store portal-specific documents and Home portal documents, because it does not allow identity propagation. • jaxws—Use a Java API for XML web services connection to connect to the Content Server.
<i>url</i>	<p>Optional. Content Server URL. Required only if <code>socketType</code> is set to <code>web</code> or <code>jaxws</code>.</p> <p>URL should be in the format:</p> <pre>http://hostname:port/web_root/plugin_root</pre> <p>For example: <code>http://mycontentserver/cms/idcplg</code></p>
<i>serverHost</i>	<p>Optional. Host name of the machine where the Content Server is running. Required if <code>socketType</code> is set to <code>socket</code> or <code>socketssl</code>.</p>
<i>serverPort</i>	<p>Optional. Port on which the Content Server listens. Required if <code>socketType</code> is set to <code>socket</code> or <code>socketssl</code>:</p> <ul style="list-style-type: none"> • Socket—Port specified for the <code>incoming</code> provider in the server. • Socket SSL—Port specified for the <code>sslincoming</code> provider in the server. <p>This property corresponds to the <code>IntradocServerPort</code> setting in the Content Server configuration file, which defaults to port 4444.</p>
<i>keystoreLocation</i>	<p>Optional. Location of key store that contains the private key used to sign the security assertions. Required only if <code>socketType</code> is set to <code>socketssl</code>.</p> <p>The key store location must be an absolute path.</p>
<i>keystorePassword</i>	<p>Optional. Password required to access the key store. Required only if <code>socketType</code> is set to <code>socketssl</code>.</p>

Argument	Definition
<code>privateKeyAlias</code>	<p>Optional. Client private key alias in the key store. The key is used to sign messages to the server. The public key corresponding to this private key must be imported in the server keystore.</p> <p>Required only if <code>socketType</code> is set to <code>socketssl</code>. The value for this argument must be a string that contains neither special characters nor white space.</p>
<code>privateKeyPassword</code>	<p>Optional. Password to be used with the private key alias in the key store. Required only if <code>socketType</code> is set to <code>socketssl</code>.</p>
<code>webContextRoot</code>	<p>Optional. Web server context root for the Content Server. Use the format <code>/<context_root></code>. For example, <code>/cs</code>.</p> <p><code>webContextRoot</code> is only applicable when <code>IDENTITY_PROPAGATION</code> is used for authentication, that is, when <code>extAppId</code> is set to an empty string.</p> <p>Note: To fully enable these Oracle WebCenter Content features you must access WebCenter Portal through Oracle HTTPS Server (OHS) to expose Content Server and the application under the same host and port. Both the application and Content Server must also use single sign on. For information about setting up OHS to front-end WebCenter Portal, see "Content Server - Configuration" in <i>Oracle Fusion Middleware Administering Oracle WebCenter Portal</i>.</p> <p>If your application is connected to multiple Content Servers, Oracle recommends that each Content Server has a unique Web Server Context Root so that OHS re-direction works correctly.</p>
<code>clientSecurityPolicy</code>	<p>Optional. Client security policy to be used when the <code>socketType</code> is <code>jaxws</code>. For example: <code>oracle/wss11_saml_token_with_message_protection_service_policy</code></p> <p>Leave the field blank if your environment supports Global Policy Attachments (GPA).</p>
<code>cacheInvalidationInterval</code>	<p>Optional. Time between checks for external Content Server content changes (in minutes). WebCenter Portal automatically clears items that have changed from the cache. The <i>minimum</i> interval is 2 minutes.</p> <p>By default, cache invalidation is disabled (specified as 0) which means that no periodic checks are made for content changes.</p>
<code>binaryCacheMaxEntrySize</code>	<p>Optional. Maximum cacheable size (in bytes) for Content Server binary documents. Documents larger than this size are not cached by WebCenter Portal.</p> <p>Defaults is 102400 bytes (100K).</p> <p>Tune this value based on your machine's memory configuration and the types of binary documents that you expect to cache.</p> <p>Most documents stored in Content Server are considered binary content, that is, images, plain text, Word documents, and so on. The only exception is Site Studio content which is stored in data files (CDF files) and cache separately in a Virtual Content Repository (VCR) cache (or node cache).</p>
<code>adminUsername</code>	<p>Required when <code>socketType</code> is <code>jaxws</code>. User name with administrative rights for this Content Server instance. This user will be used to fetch content type information based on profiles and track document changes for cache invalidation purpose.</p> <p>Defaults to <code>sysadmin</code>.</p>

Argument	Definition
<i>adminPassword</i>	Optional. Password for the Content Server administrator specified in <i>adminUsername</i> . Required when <i>socketType</i> is set to <i>web</i> .
<i>extAppId</i>	Optional. External application used to authenticate users against the Content Server. This value should match the name of an existing external application connection. See also listExtAppConnections . If <i>extAppId</i> is not set, no change is made to the authentication method or external application ID. If <i>extAppId</i> is set to an empty string, the authentication method used is <code>IDENTITY_PROPAGATION</code> . With this method, the application and Content Server use the same identity store to authenticate users. Note that <i>extAppID</i> is mandatory when <i>socketType</i> is set to <i>web</i> .
<i>timeout</i>	Length of time allowed to log in to the Content Server (in ms) before issuing a connection timeout message, and the RIDD socket timeout used for all service requests for connection types <i>web</i> , <i>socket</i> and <i>socketssl</i> . If the <i>timeout</i> property is not set, the following values are used: <ul style="list-style-type: none"> Login timeout - the default concurrency timeout for the <code>oracle.webcenter.content</code> resource is used (30s or 30000ms). RIDD socket timeout - the default RIDD socket timeout (60s or 60000ms) is used for all service requests for connection types <i>socket</i>, <i>socketssl</i>, or <i>web</i>. If the <i>timeout</i> property is set and the connection type is <i>socket</i> , <i>socketssl</i> , or <i>web</i> , Oracle recommends that you do not specify a value less than 60000ms as this would reduce the RIDD socket timeout and increase the likelihood that long running requests will time out. For example, timeouts may occur during long running searches, long file uploads, or long copy operations.
<i>isPrimary</i>	Optional. Valid string values are 1 (true) and 0 (false). 1 specifies that this connection is the primary connection used for the documents tool. This argument defaults to 0. When omitted or set to 0, the primary connection used for documents does not change. In WebCenter Portal, the primary connection is used to store portal-specific content and Home portal content. Note: If you mark a connection as primary, you must run the <code>setContentServerProperties WLST</code> command to specify certain additional properties required for the primary WebCenter Content Server connection.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example creates a socket-based connection to an Oracle WebCenter Content repository running on `myhost.com` at port 4444. For authentication purposes, an existing external application named `myExtApp` is used. See also, [createExtAppConnection](#).

```
wls:/weblogic/serverConfig> createContentServerConnection(appName='webcenter',
name='myContentServerConnection', socketType='socket',
serverHost='myhost.com', serverPort=4444, extAppId='myExtApp',
isPrimary=1)
```

The following example creates an SSL socket-based connection to an Oracle WebCenter Content repository.

```
wls:/weblogic/serverConfig> createContentServerConnection(appName='webcenter',
name='myContentServerConnection', socketType='socketssl',
serverHost='myhost.com', serverPort=4444, keystoreLocation='d:/keys/here',
keystorePassword='AlphaSquad7',
privateKeyAlias='enigma', privateKeyPassword='S0larPl3x1s',
extAppId='myExtApp')
```

The following example creates a JAX-WS (Java API for XML web services) connection to an Oracle WebCenter Content repository:

```
wls:/weblogic/serverConfig> createContentServerConnection(appName='webcenter'
name='myContentServerConnection', socketType='jaxws', url='http://
myhost.com:9044/idcnativews',
adminUsername='weblogic', clientSecurityPolicy='oracle/
wss10_saml_token_client_policy')
```

2.6.2 setContentServerConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Edits an existing Oracle WebCenter Content repository connection. This command requires that you specify values for `appName` and `name`, plus one additional argument.

Syntax

```
setContentServerConnection(appName, name, [socketType, url, serverHost,
serverPort, keystoreLocation, keystorePassword, privateKeyAlias,
privateKeyPassword, webContextRoot, clientSecurityPolicy,
cacheInvalidationInterval, binaryCacheMaxEntrySize, adminUsername,
adminPassword,
extAppId, timeout, isPrimary, server, applicationVersion])
```

Argument	Definition
<code>appName</code>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<code>name</code>	Name of an existing Oracle WebCenter Content repository connection.

Argument	Definition
<i>socketType</i>	<p>Optional. Specifies whether the Oracle WebCenter Content's Content Server connects on the content server listener port or the web server filter, and whether the listener port is SSL enabled.</p> <p>Valid values are <code>socket</code>, <code>web</code>, and <code>socketssl</code>. This option has no default.</p> <p>Choose from:</p> <ul style="list-style-type: none"> • socket—Use an <code>intradoc</code> socket connection to connect to the Content Server. The client IP address must be added to the list of authorized addresses in the Content Server. In this case, the client is the machine on which Oracle WebCenter Portal is running. • socketssl—Use an <code>intradoc</code> socket connection to connect to the Content Server that is secured using the SSL protocol. The client's certificates must be imported in the server's trust store for the connection to be allowed. This is the most secure option, and the recommended option whenever identity propagation is required (for example, in WebCenter Portal). • web—Use an HTTP(S) connection to connect to the Content Server. Note that for WebCenter Portal, this option is not suitable for the active connection, that is, the back-end Content Server repository that is being used to store portal-specific documents and Home portal documents, because it does not allow identity propagation. • jaxws—Use a Java API for XML web services connection to connect to the Content Server.
<i>url</i>	<p>Optional. Content Server URL. Required only if <code>socketType</code> is set to <code>web</code> or <code>jaxws</code>.</p> <p>URL should be in the format:</p> <pre>http://hostname:port/web_root/plugin_root</pre> <p>For example: <code>http://mycontentserver/cms/idcplg</code></p>
<i>serverHost</i>	<p>Optional. Host name of the machine where the Content Server is running. Required if <code>socketType</code> is set to <code>socket</code> or <code>socketssl</code>.</p>
<i>serverPort</i>	<p>Optional. Port on which the Content Server listens. Required if <code>socketType</code> is set to <code>socket</code> or <code>socketssl</code>:</p> <ul style="list-style-type: none"> • Socket—Port specified for the <code>incoming</code> provider in the server. • Socket SSL—Port specified for the <code>sslincoming</code> provider in the server. <p>For example, 4444</p>
<i>keystoreLocation</i>	<p>Optional. Location of key store that contains the private key used to sign the security assertions. Required only if <code>socketType</code> is set to <code>socketssl</code>.</p> <p>The key store location must be an absolute path.</p>
<i>keystorePassword</i>	<p>Optional. Password required to access the key store. Required only if <code>socketType</code> is set to <code>socketssl</code>.</p>
<i>privateKeyAlias</i>	<p>Optional. Client private key alias in the key store. Required only if <code>socketType</code> is set to <code>socketssl</code>.</p> <p>The value for this argument must be a string that contains neither special characters nor white space.</p>

Argument	Definition
<code>privateKeyPassword</code>	Optional. Password to be used with the private key alias in the key store. Required only if <code>socketType</code> is set to <code>socketssl</code> .
<code>webContextRoot</code>	<p>Optional. Web server context root for the Content Server if Content Server is front-ended with Oracle HTTP Server (OHS). Use the format <code><context_root></code>. For example, <code>/cs</code>.</p> <p>Oracle recommends that you access WebCenter Portal through Oracle HTTP Server (OHS) if you want to use Content Presenter to create or edit Site Studio content. Without Oracle HTTP Server (and <code>WebContextRoot</code> configuration), it is still possible to create or edit Site Studio content from within Content Presenter, but the create and edit actions launch new browser windows (or tabs) rather than opening within the Content Presenter task flow.</p> <p>Note: To fully enable these features you must access WebCenter Portal through Oracle HTTPS Server (OHS) to expose Content Server and the application under the same host and port. In addition, both the application and the Content Server must use single sign on.</p> <p><code>webContextRoot</code> is only applicable when <code>IDENTITY_PROPAGATION</code> is used for authentication, that is, when <code>extAppId</code> is set to an empty string.</p>
<code>clientSecurityPolicy</code>	<p>Optional. Client security policy to be used when the <code>socketType</code> is <code>jaxws</code>. For example: <code>oracle/wss11_saml_token_with_message_protection_service_policy</code></p> <p>Leave the field blank if your environment supports Global Policy Attachments (GPA).</p>
<code>cacheInvalidationInterval</code>	<p>Optional. Time between checks for external Content Server content changes (in minutes). WebCenter Portal automatically clears items that have changed from the cache. The <i>minimum</i> interval is 2 minutes.</p> <p>By default, cache invalidation is disabled (specified as 0) which means that no periodic checks are made for content changes.</p>
<code>binaryCacheMaxEntrySize</code>	<p>Optional. Maximum cacheable size (in bytes) for Content Server binary documents. Documents larger than this size are not cached by WebCenter Portal. Defaults is 102400 bytes (100K).</p> <p>Tune this value based on your machine's memory configuration and the types of binary documents that you expect to cache.</p>
<code>adminUsername</code>	<p>Optional. User name with administrative rights for this Content Server instance. This user will be used to fetch content type information based on profiles and track document changes for cache invalidation purpose.</p> <p>Defaults to <code>sysadmin</code>.</p>
<code>adminPassword</code>	Optional. Password for the Content Server administrator specified in <code>adminUsername</code> . Required when <code>socketType</code> is set to <code>web</code> .
<code>extAppId</code>	<p>Optional. External application used to authenticate users against the Content Server. This value should match the name of an existing external application connection. See also listExtAppConnections. If <code>extAppId</code> is not set, no change is made to the authentication method or external application ID.</p> <p>If <code>extAppId</code> is set to an empty string, the authentication method used is <code>IDENTITY_PROPAGATION</code>. With this method, the application and Content Server use the same identity store to authenticate users.</p>

Argument	Definition
<i>timeout</i>	<p>Length of time allowed to log in to the Content Server (in ms) before issuing a connection timeout message, and the RIDC socket timeout used for all service requests for connection types <code>web</code>, <code>socket</code> and <code>socketssl</code>.</p> <p>If the <code>timeout</code> property is not set, the following values are used:</p> <ul style="list-style-type: none"> • Login timeout - the default concurrency timeout for the <code>oracle.webcenter.content</code> resource is used (30s or 30000ms). For more information, refer to Configuring Concurrency Management in <i>Tuning Performance</i>. • RIDC socket timeout - the default RIDC socket timeout (60s or 60000ms) is used for all service requests for connection types <code>socket</code>, <code>socketssl</code>, or <code>web</code>. <p>If the <code>timeout</code> property is set and the connection type is <code>socket</code>, <code>socketssl</code>, or <code>web</code>, Oracle recommends that you do not specify a value less than 60000ms as this would reduce the RIDC socket timeout and increase the likelihood that long running requests will time out. For example, timeouts may occur during long running searches, long file uploads, or long copy operations.</p>
<i>isPrimary</i>	<p>Optional. Valid string values are 1 (true) and 0 (false).</p> <p>1 specifies that this connection is the primary connection used for document services in the application.</p> <p>This argument defaults to 0. When omitted or set to 0, the primary connection used for documents does not change.</p> <p>In WebCenter Portal, the primary connection is used to store portal-specific content and Home portal content.</p>
<i>server</i>	<p>Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code>.</p> <p>Required when applications with the same name are deployed to different servers and also when you have a cluster.</p>
<i>applicationVersion</i>	<p>Optional. Version number of the deployed application. Required if more than one version of the application is deployed.</p>

Example

The following example edits a socket-based connection to an Oracle WebCenter Content repository.

```
wls:/weblogic/serverConfig>setContentServerConnection( appName='webcenter' ,
name='myContentServerConnection' , socketType='socket' ,
serverHost='myhost.com' , serverPort=4444 ,
extAppId='myExtApp' , isPrimary=1)
```

The following example edits an SSL socket-based connection to an Oracle WebCenter Content repository.

```
wls:/weblogic/serverConfig>setContentServerConnection( appName='webcenter' ,
name='myContentServerConnection' , socketType='socketssl' ,
serverHost='myhost.com' , serverPort=8443 ,
keystoreLocation='d:/keys/here' , keystorePassword='TOPS3CR3T' ,
privateKeyAlias='TekJansen' , privateKeyPassword='LadyNocturne' ,
extAppId='myExtApp' , isPrimary=1)
```

The following example edits a JAX-WS (Java API for XML web services) connection to an Oracle WebCenter Content repository:

```
wls:/weblogic/serverConfig> setContentServerConnection(appName='webcenter',
socketType='jaxws', url='http://myhost.com:9044/idcnativews',
adminUsername='weblogic',
clientSecurityPolicy='oracle/wss10_saml_token_client_policy')
```

2.6.3 listContentServerConnections

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Without any arguments, this command lists all of the Oracle WebCenter Content repository connections that are configured for a named application.

Syntax

```
listContentServerConnections(appName, [verbose, name, server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>verbose</i>	Optional. Displays content repository connection details in verbose mode. Valid options are 1 (true) and 0(false). When set to 1, <code>listJCRContentServerConnections</code> lists all Oracle WebCenter Content repository connections that are configured for the named application, along with their details. When set to 0, only connection names are listed. This argument defaults to 0.
<i>name</i>	Optional. Name of an existing Oracle WebCenter Content repository connection. When specified you can view connection details for a specific Oracle WebCenter Content repository connection. If you supply a value for <code>name</code> , you must supply a value for <code>verbose</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example lists Oracle WebCenter Content repository connections configured for WebCenter Portal (`webcenter`):

```
wls:/weblogic/serverConfig> listContentServerConnections(appName='webcenter')
```

The following example lists all properties of the Oracle WebCenter Content repository connection named `myContentServerConnection1`. The connection named

`myContentServerConnection1` must exist and be an Oracle WebCenter Content repository connection.

```
wls:/weblogic/serverConfig>listContentServerConnections(appName='webcenter',
verbose=1, name='myContentServerConnection1')
```

2.6.4 listContentServerProperties

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists properties for the back-end Oracle WebCenter Content repository that is being used by WebCenter Portal to store portal-specific documents and Home portal documents. This command is only valid for WebCenter Portal.

Syntax

```
listContentServerProperties(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example lists properties for the back-end Oracle WebCenter Content repository that is being used by a WebCenter Portal instance (`webcenter`) to store portal-specific documents and Home portal documents.

```
wls:/weblogic/serverConfig> listContentServerProperties(appName='webcenter')
```

```
The Documents Portal Server identifier is "/EnterpriseLibraries"
The Documents repository administrator is "sysadmin"
The Documents security group is "/WebCenter1109"
The Documents primary connection is "myContentServerConnection"
```

2.6.5 setContentServerProperties

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Modifies properties for the back-end Oracle WebCenter Content repository that is being used by WebCenter Portal to store portal-related data.

Syntax

```
setContentServerProperties(appName, [portalServerIdentifier, adminUserName,
securityGroup, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>portalServerIdentifier</i>	Optional. Root folder under which WebCenter Portal content is stored. The value for this argument must use the format: <code>/folderName</code> . For example, <code>/WebCenter</code> or <code>/WCPMain</code> . The <code>portalServerIdentifier</code> cannot be <code>/</code> (the root itself), and it must be unique across applications. If the folder specified does not exist it will be created for you. Note: If you change the value of this argument, you must also provide the value for the <code>adminUserName</code> and <code>securityGroup</code> arguments.
<i>adminUserName</i>	Optional. User name of a content repository administrator. For example: <code>sysadmin</code> . This user will be used to create and maintain folders for WebCenter Portal content and manage content access rights. Administrative privileges are required for this connection so that operations can be performed on behalf of WebCenter Portal users. Administrative privileges are required for this connection so that operations can be performed on behalf of WebCenter Portal users.
<i>securityGroup</i>	Optional. Unique identifier to use as the value for the security group assigned to files in Content Server that are created in WebCenter Portal. This name is used to separate data when multiple WebCenter Portal instances share the same WebCenter Content instance, and must be unique across WebCenter Portal instances. The value for this argument must begin with an alphabetical character, followed by any combination of alphanumeric characters or the underscore character. The string must be less than or equal to 30 characters. Note: If you change the value of this argument, you must also change the value of the <code>portalServerIdentifier</code> and <code>adminUserName</code> arguments.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example modifies connection properties for the back-end Oracle WebCenter Content repository that is being used by WebCenter Portal to store portal-related documents:

```
wls:/weblogic/serverConfig> setContentServerProperties(appName='webcenter',
portalServerIdentifier='/AccountingSpaces', adminUserName='admin',
securityGroup='WCAccounting')
```

The following example modifies the administrator's user name for the back-end Oracle WebCenter Content repository that is being used by WebCenter Portal to store portal-related documents:

```
wls:/weblogic/serverConfig> setContentServerProperties( appName='webcenter',
adminUserName='sysadmin' )
```

2.6.6 deleteContentServerProperties

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Deletes properties for the back-end Oracle WebCenter Content repository used by WebCenter Portal, that is the `adminUserName`, `applicationName`, and `spacesRoot`.

Syntax

```
deleteContentServerProperties(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example deletes connection properties (`adminUserName`, `applicationName`, `spacesRoot`) of the back-end Oracle WebCenter Content repository that is being used by WebCenter Portal:

```
wls:/weblogic/serverConfig> deleteContentServerProperties( appName='webcenter' )
```

2.6.7 deleteContentServerConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Deletes a named Oracle WebCenter Content repository connection.

Syntax

```
deleteContentServerConnection(appName, name)
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Name of the existing Oracle WebCenter Content repository connection that you want to delete.

Example

The following example deletes the Oracle WebCenter Content repository connection named `MyContentServerConnection`.

```
wls:/weblogic/serverConfig> deleteContentServerConnection(appName='webcenter'
name='MyContentServerConnection')
```

2.6.8 exportFoldersGData

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Exports `Folders_g` data to a specified location so you can migrate the data to `FrameworkFolders` using the WLST command `migrateFoldersGToFrameworkFolders`. After migrating `Folders_g` data to `FrameworkFolders`, you can generate comparison reports from the exported `Folders_g` data to verify that all the files and folders are migrated to `FrameworkFolders`.

Syntax

```
exportFoldersGData(appName, server,
[connectionName,directoryPath,applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Note: In a clustered environment where applications share the same content server connection (that is, applications connect to the same content server and share the same root folder and security group), run this command only against one managed server.
<i>connectionName</i>	Optional. Content Server connection name. Only required if you want to export <code>Folders_g</code> data for a non default content server connection. If this argument is not specified, the default content server connection is used, that is, the content server connection where <code>isPrimary=1</code> .
<i>directoryPath</i>	Optional. Destination directory for the <code>Folders_g</code> data that you want to migrate. If this argument is not specified, <code>Folders_g</code> data is exported to the folder named <code>FRAMEWORK_FOLDER_MIGRATION</code> under <code>WCP_ORACLE_HOME/common/wlst</code> .

Argument	Definition
applicationVersion	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example exports the Folders_g data for a WebCenter Portal application deployed to the WC_Portal managed server. WebCenter Portal's default content server connection details are used, and Folders_g data exported to the default location, the `WCP_ORACLE_HOME/common/wlst/Framework_FOLDER_MIGRATION` directory.

```
exportFoldersGData(appName='webcenter',server='WC_Portal')
```

The following example exports the Folders_g data for a WebCenter Portal application deployed to the WC_Portal managed server. The content server connection named MyContentServerConnection is used, and Folders_g data exported to the `/scratch/myTemp_Dir` directory.

```
exportFoldersGData(appName='webcenter',server='WC_Portal',connectionName='MyContentServerConnection',directoryPath='/scratch/myTemp_Dir/')
```

2.6.9 migrateFoldersGDataToFrameworkFolders

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Migrates the Folders_g data to FrameworkFolders and verifies the integrity of the migrated data.

Syntax

```
migrateFoldersGDataToFrameworkFolders(appName,server,contentDbConnectionUrl,contentDbUserName,[connectionName,directoryPath,reportMode,applicationVersion])
```

Argument	Definition
appName	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
server	Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Note: In a clustered environment where applications share the same content server connection (that is, connect to the same content server and share the same root folder and security group), run this command against only one managed server.
contentDbConnectionUrl	Connection URL for the database where WebCenter Content schema (named OCS) is present. Use the connection URL format <code>host:port:sid</code> .
contentDbUserName	User name of the WebCenter Content schema (named OCS) that you want to migrate.

Argument	Definition
connectionName	Optional. Content Server connection name. Only required if you want to migrate Folders_g data for a non default content server connection. If not specified, the default content server connection is used, that is, the content server connection where isPrimary=1.
directoryPath	Optional. Directory containing the Folders_g data that you want to migrate. If not specified, Folders_g data is migrated from <i>WCP_ORACLE_HOME/common/wlst/FRAMEWORK_FOLDER_MIGRATION</i> directory. This is the default location for Folders_g data exported using the WLST command <code>exportFoldersGData</code> .
reportMode	Optional. Indicates whether to generate a report describing what will happen during migration that you can review prior to migrating the actual data. Valid values are <code>true</code> and <code>false</code> . <code>true</code> - Generates a report that describes all the metadata changes that will be made when the Folders_g data is migrated to FrameworkFolders. No actual data migration takes place. <code>false</code> - Pre-migration report not required. Folders_g data is migrated to FrameworkFolders. The default value is <code>false</code> .
applicationVersion	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example migrates the Folders_g data from the default directory (*WCP_ORACLE_HOME/common/wlst/FRAMEWORK_FOLDER_MIGRATION*) to FrameworkFolders and validates the migrated data for the WebCenter Portal application deployed to the WC_Portal managed server. Migration is performed using the default Content Server connection details and the specified WebCenter Content database connection and user name.

```
migrateFoldersGDataToFrameworkFolders(appName='webcenter',server='WC_Portal',contentDbConnectionUrl='wccdbhost.example.com:wccdbport:wccdbsid',contentDbUserName='SCHEMA_PREFIX_OCS')
```

The following example migrates Folders_g data from the `/scratch/myTemp_Dir` directory to FrameworkFolders and validates the migrated data for the WebCenter Portal application deployed to the WC_Portal managed server. Content Server connection named `MyContentServerConnection` and the specified WebCenter Content database connection and username are used to perform the migration.

```
migrateFoldersGDataToFrameworkFolders(appName='webcenter',server='WC_Portal',contentDbConnectionUrl='wccdbhost.example.com:wccdbport:wccdbsid',contentDbUserName='SCHEMA_PREFIX_OCS',connectionName='MyContentServerConnection',directoryPath='/scratch/myTemp_Dir/)
```

The following example reports how Folders_g data will be migrated to FrameworkFolders for the WebCenter Portal application deployed to WC_Portal. `MyContentServerConnection` connection details and specified WebCenter Content

database connection and username are used to generate the migration report which is saved to the same location as the Folders_g data (/scratch/myTemp_Dir):

```
migrateFoldersGDataToFrameworkFolders(appName='webcenter',server='WC_Portal',contentDbC
onnectionUrl='wccdbhost.example.com:wccdbport:wccdbsid',
contentDbUserName='SCHEMA_PREFIX_OCS',connectionName='MyContentServerConnection',direct
oryPath='/scratch/myTemp_Dir/', reportMode='true')
```

2.7 Discussions and Announcements

Note:

Oracle WebCenter Portal has deprecated the support for Jive features (announcements and discussions). If you have upgraded from a prior release to Release 12c (12.2.1.4.0), Jive features remain available in your upgraded instance but Oracle support is not provided for these features. In the next release, Jive features will not be available even in the upgraded instances

Use the commands listed in [Table 2-8](#) to manage discussions server connections for WebCenter Portal.

Configuration changes made using these WLST commands are only effective after your restart the Managed Server on which the application is deployed. For details, see *Oracle Fusion Middleware Administering Oracle WebCenter Portal*.

Table 2-8 Discussion and Announcement WLST Commands

Use this command...	To...	Use with WLST...
createDiscussionForumConnection	Create a new discussions server connection for a named application.	Online
setDiscussionForumConnection	Edit an existing discussions server connection.	Online
setDefaultDiscussionForumConnection	Specify the default connection for discussions and announcements.	Online
listDiscussionForumConnections	List all of the discussions server connections that are configured for a named application.	Online
listDefaultDiscussionForumConnection	List the default discussions server connection for a named application.	Online
setDiscussionForumConnectionProperty	Set an additional discussions server connection property.	Online
deleteDiscussionForumConnectionProperty	Delete a discussions server connection property.	Online
setDiscussionForumServiceProperty	Specify defaults for discussions.	Online
removeDiscussionForumServiceProperty	Remove defaults for discussions.	Online
listDiscussionForumServiceProperties	List discussions settings.	Online
setAnnouncementServiceProperty	Specify defaults for announcements.	Online

Table 2-8 (Cont.) Discussion and Announcement WLST Commands

Use this command...	To...	Use with WLST...
removeAnnouncementServiceProperty	Remove defaults for announcements.	Online
listAnnouncementServiceProperties	List announcements settings.	Online
addDiscussionsServerAdmin	Grant system administrator permissions on the discussions server to a user or a group.	Online
addDiscussionsCategoryAdmin	Grant category administrator permissions on the discussions server to a user or a group.	Online
setDiscussionsServerProperty	Set discussions server properties.	Online
getDiscussionsServerProperty	Return discussions server property values.	Online
removeDiscussionsServerProperty	Remove current discussions server property values.	Online

2.7.1 createDiscussionForumConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Creates a new discussions server connection for a named application.

Discussions and announcements both require a discussions server connection. Discussions and announcements both use the same discussions server connection.

While you can register multiple discussions server connections for an application, only one connection is used for discussion and announcement services - the default (or active) connection.

Syntax

```
createDiscussionForumConnection(appName, name, url, adminUser,
[timeout, default, policyURIForAuthAccess, policyURIForPublicAccess,
recipientKeyAlias, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>name</i>	Connection name. The name must be unique (across all connection types) within the application.
<i>url</i>	URL of the discussions server hosting discussion forums and announcements. For example: <code>http://myhost:8888/owc_discussions</code> .

Argument	Definition
<code>adminUser</code>	<p>Name of the discussions server administrator. This account is used by discussions and announcements to perform administrative operations on behalf of WebCenter Portal users.</p> <p>This account is mostly used for managing discussions and announcements in WebCenter Portal. It is not necessary for this user to be a <code>super admin</code>. However, the user must have administrative privileges on the current application root category for WebCenter Portal, that is, the category (on the discussions server) under which all portal-related discussions and announcements are stored.</p>
<code>policyURIForAuthenticated</code>	<p>Optional. URI to the SAML token based policy required for authenticated access to the discussions server web service.</p> <p>The client policy specified must be compatible with the service policy that is configured for the <code>OWCDiscussionsServiceAuthenticated</code> endpoint in the discussions server. Out-of-the-box, the default service policy is WSS 1.0 SAML Token Service Policy (<code>oracle/wss10_saml_token_service_policy</code>).</p> <p>Valid client policy values include:</p> <ul style="list-style-type: none"> • <code>oracle/wss10_saml_token_client_policy</code> (WSS 1.0 SAML Token Client Policy) • <code>oracle/wss11_saml_token_with_message_protection_client_policy</code> (WSS 1.1 SAML Token with Message Protection Client Policy) • GPA (Global Policy Attachment) - Use GPA if your environment supports Global Policy Attachments. In addition, ensure that the default policy is detached from the <code>OWCDiscussionsServiceAuthenticated</code> endpoint in the discussions server using the WLST command <code>detachWebServicePolicy</code> or Enterprise Manager. <p>See also "Managing Announcements and Discussions" in <i>Administering Oracle WebCenter Portal</i>.</p>
<code>policyURIForPublicAccess</code>	<p>Optional. URI to the policy required to enforce message security and integrity for public access to the discussions server Web service.</p> <p>Default value is <code>oracle/no_authentication_client_policy</code>.</p> <p>The client policy specified must be compatible with the service policy that is configured for the <code>OWCDiscussionsServicePublic</code> endpoint in the discussions server. Out-of-the-box, a service policy is not configured for public access (<code>oracle/no_authentication_client_policy</code>).</p> <p>Valid client policy values include:</p> <ul style="list-style-type: none"> • <code>oracle/no_authentication_client_policy</code> (None) • <code>oracle/wss11_with_message_protection_client_policy</code> (WSS 1.1 Message Protection Client Policy) • GPA (Global Policy Attachment) - Use GPA if your environment supports Global Policy Attachments. In addition, you must ensure that the default policy attached to the <code>OWCDiscussionsServicePublic</code> endpoint in the discussions server is set to <code>oracle/no_authentication_service_policy</code>.

Argument	Definition
<i>recipientKeyAlias</i>	Optional. Recipient key alias to be used for message protected policies (applicable to the <code>OWCDiscussionsServicePublic</code> and <code>OWCDiscussionsServiceAuthenticated</code> endpoints). This is the alias to the certificate that contains the public key of the discussions server in the configured keystore. The default is null. See also "Configuring WS-Security" in <i>Administering Oracle WebCenter Portal</i> .
<i>timeout</i>	Optional. Length of time (in seconds) discussions waits for a response from the discussions server before issuing a connection timeout message. This argument defaults to -1. When set to -1, the service default (10 seconds) applies.
<i>default</i>	Optional. Indicates that this connection is the default connection for discussions and announcements. Valid options are 1 (true) and 0 (false). When set to 1, discussions and announcements both use this connection. When set to 0, the connection is not used. The default is 0.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example creates a discussions server connection for WebCenter Portal:

```
wls:/weblogic/serverConfig>createDiscussionForumConnection(appName='webcenter',
name='MyDiscussionServer', url='http://myhost.com:8888/owc_discussions',
adminUser='admin', policyURIForAuthAccess='oracle/
wss10_saml_token_client_policy', default=0)
```

2.7.2 setDiscussionForumConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Edits an existing discussions server connection. Use this command to update connection attributes.

The connection is created using the [createDiscussionForumConnection](#) command.

Syntax

```
setDiscussionForumConnection(appName, name, [url, adminUser,
policyURIForAuthAccess, policyURIForPublicAccess, recipientKeyAlias, timeout,
default, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Name of an existing discussions server connection.
<i>url</i>	Optional. URL to the discussions server.
<i>adminUser</i>	Optional. Name of the discussions server administrator. This account is used by discussions to perform administrative operations on behalf of WebCenter Portal users. This account is mostly used for managing discussions and announcements in WebCenter Portal. It is not necessary for this user to be a <code>super admin</code> . However, the user must have administrative privileges on the current application root category for WebCenter Portal, that is, the category (on the discussions server) under which all portal-related discussions and announcements are stored.
<i>policyURIForAuthAccess</i>	Optional. URI to the SAML token based policy required for authenticated access to the discussions server web service. The client policy specified must be compatible with the service policy that is configured for the <code>OWCDiscussionsServiceAuthenticated</code> endpoint in the discussions server. Out-of-the-box, the default service policy is WSS 1.0 SAML Token Service Policy (<code>oracle/wss10_saml_token_service_policy</code>). Valid client policy values include: <ul style="list-style-type: none">• <code>oracle/wss10_saml_token_client_policy</code> (WSS 1.0 SAML Token Client Policy)• <code>oracle/wss11_saml_token_with_message_protection_client_policy</code> (WSS 1.1 SAML Token with Message Protection Client Policy)• GPA (Global Policy Attachment) - Use GPA if your environment supports Global Policy Attachments. In addition, ensure that the default policy is detached from the <code>OWCDiscussionsServiceAuthenticated</code> endpoint in the discussions server using the WLST command <code>detachWebServicePolicy</code> or Enterprise Manager. See also "Managing Announcements and Discussions" in <i>Administering Oracle WebCenter Portal</i> .

Argument	Definition
<code>policyURIForPublicAccess</code>	<p>Optional. URI to the policy required to enforce message security and integrity for public access to the discussions server web service.</p> <p>Default value is <code>oracle/no_authentication_client_policy</code>.</p> <p>The client policy specified must be compatible with the service policy that is configured for the <code>OWCDiscussionsServicePublic</code> endpoint in the discussions server. Out-of-the-box, a service policy is not configured for public access (<code>oracle/no_authentication_client_policy</code>).</p> <p>Valid client values include:</p> <ul style="list-style-type: none"> <code>oracle/no_authentication_client_policy</code> (None) <code>oracle/wss11_with_message_protection_client_policy</code> (WSS 1.1 Message Protection Client Policy) GPA (Global Policy Attachment) - Use GPA if your environment supports Global Policy Attachments. In addition, you must ensure that the default policy attached to the <code>OWCDiscussionsServicePublic</code> endpoint in the discussions server is set to <code>oracle/no_authentication_service_policy</code>.
<code>recipientKeyAlias</code>	<p>Optional. Recipient key alias to be used for message protected policies (applicable to the <code>OWCDiscussionsServicePublic</code> and <code>OWCDiscussionsServiceAuthenticated</code> endpoints). This is the alias to the certificate that contains the public key of the discussions server in the configured keystore. The default is null.</p> <p>See also "Configuring WS-Security" in <i>Administering Oracle WebCenter Portal</i>.</p>
<code>timeout</code>	<p>Optional. Length of time (in seconds) discussions and announcements wait for a response from the discussions server before issuing a connection timeout message. This argument defaults to <code>-1</code>. When set to <code>-1</code>, the service default (10 seconds) applies.</p>
<code>default</code>	<p>Optional. Indicates that this connection is the default connection for discussions and announcements. Required only if more than one connection is defined.</p> <p>Valid options are <code>1</code> (true) and <code>0</code> (false). When set to <code>1</code>, discussions and announcements use this connection. When set to <code>0</code>, the connection is not used. The default is <code>0</code>.</p> <p>To specify that discussion and announcements use this connection, change the value from <code>0</code> to <code>1</code>.</p> <p>To disable this connection, use the <code>removeDiscussionForumServiceProperty</code> command:</p> <pre>removeDiscussionForumServiceProperty('appName='webcenter', property='selected.connection')</pre> <p>Note: While you can register multiple discussions server connections for an application, only one connection is used for discussions and announcements—the default (or active) connection.</p>
<code>server</code>	<p>Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code>.</p> <p>Required when applications with the same name are deployed to different servers and also when you have a cluster.</p>
<code>applicationVersion</code>	<p>Optional. Version number of the deployed application. Required if more than one version of the application is deployed.</p>

Example

The following example updates attributes for a secure discussions server connection named MyDiscussionsServer.

```
wls:/weblogic/serverConfig> setDiscussionForumConnection(appName='webcenter',
name='MyDiscussionServer', url='http://myhost.com:7786/owc_discussions',
adminUser='admin', policyURIForAuthAccess='oracle/wss10_saml_token_client_policy',
default=1)
```

2.7.3 setDiscussionForumConnectionProperty

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Sets a discussions server connection property. Use this command when additional parameters are required to connect to your discussions server.

This commands provides an extensible way to add any connection property using a key and a value. (You are not limited to connection properties specified by [createDiscussionForumConnection](#) and [setDiscussionForumConnection](#) .)

Note:

Do not use the [setDiscussionForumConnectionProperty](#) to set connection properties available through [createDiscussionForumConnection](#) or [setDiscussionForumConnection](#) . Attempting to do so, has no effect.

All known, additional connection properties are listed in [Table 2-9](#).

Table 2-9 Additional Discussion Server Connection Properties

Additional Connection Property	Description
<i>application.root.category.id</i>	Application root category ID on the discussions server under which all discussion forums are stored. For example, if set to 3, then all forums are stored in the category with the ID 3.
<i>linkURL</i>	URL used to link users to the discussions server's Admin Console. Only required if it is different to the <code>url</code> argument specified using the createDiscussionForumConnection or setDiscussionForumConnection command. For example, when SSO or HTTPS is configured. Use the following format to specify an alternative public external URL: protocol://host:port For example: http://example.com:7777

Syntax

```
setDiscussionForumConnectionProperty(appName, name, key, value, [secure, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>name</i>	Name of an existing discussions server connection.
<i>key</i>	Name of the connection property. See also, Table 2-9 .
<i>value</i>	Value for the property. Allows any property to be modified on the connection with a key and value.
<i>secure</i>	Optional. Indicates whether the property value must be stored securely using encryption. Valid options are 1 (true) and 0 (false). When 1, the value is encrypted. The default option is 0. Set to 1 if you are storing passwords.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example configures the category used to store discussion forums for a discussions server connection named `MyDiscussionServer`:

```
wls:/weblogic/serverConfig>setDiscussionForumConnectionProperty
(appName='webcenter', name='MyDiscussionServer',
key='application.root.category.id', value='3')
```

The following example configures an alternative, public external URL so users can access the discussions server's Admin Console:

```
wls:/weblogic/serverConfig>setDiscussionForumConnectionProperty
(appName='webcenter', name='MyDiscussionServer', key='linkURL', value='http://
example.com:7777')
```

The following example adds a custom discussions server connection property called `myProperty1` with a value `propertyValue1`:

```
wls:/weblogic/serverConfig>setDiscussionForumConnectionProperty
(appName='webcenter', name='MyDiscussionServer', key='myProperty1',
value='propertyValue1')
```

The following example adds a secured discussions server connection property called `securedProperty` with the value `secureValue`.

```
wls:/weblogic/serverConfig>setDiscussionForumConnectionProperty
(appName='webcenter', name='MyDiscussionServer', key='securedProperty',
value='secureValue', secure=1)
```

2.7.4 deleteDiscussionForumConnectionProperty

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Deletes a discussions server connection property. Take care when deleting connection properties because the connection may not work as expected if the configuration becomes invalid as a result.

This command can only delete *additional* connection properties added using the `setDiscussionForumConnectionProperty` command.

Syntax

```
deleteDiscussionForumConnectionProperty(appName, name, key, [server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Name of an existing discussions server connection.
<i>key</i>	Name of the connection property you want to delete.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example deletes a discussions server connection property named `myProperty1`.

```
wls:/weblogic/serverConfig> deleteDiscussionForumConnectionProperty
(appName='webcenter', name='MyDiscussionServer', key='myProperty1')
```

2.7.5 listDefaultDiscussionForumConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Names the discussions server connection that discussions and announcements service are using, in a named application. While you can register multiple discussions server connections for an application, discussions and announcements only uses one connection—known as the default (or active) connection.

Syntax

```
listDefaultDiscussionForumConnection(appName, [verbose, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>verbose</i>	Optional. Valid options are 1 (true) and 0 (false). When set to 1, the name and details of the discussions server connections are listed. When set to 0, only the connection name displays. This argument defaults to 0.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example names the discussions server connection that discussions and announcements are using in WebCenter Portal:

```
wls:/weblogic/
serverConfig>listDefaultDiscussionForumConnection(appName='webcenter')
```

The following example lists the name and details of the discussions server connection that discussions and announcements are using in WebCenter Portal:

```
wls:/weblogic/serverConfig>
listDefaultDiscussionForumConnection(appName='webcenter', verbose=1)
```

2.7.6 listDiscussionForumConnections

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists all the discussions server connections that are configured for a named application.

Syntax

```
listDiscussionForumConnections(appName, [verbose, name, server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.

Argument	Definition
<i>verbose</i>	Optional. Valid options are 1 (true) and 0 (false). When set to 1, <code>listDiscussionForumConnections</code> lists all of the discussions server connections that are configured for an application, along with their details. When set to 0, only connection names are listed. This argument defaults to 0.
<i>name</i>	Optional. Name of an existing discussions server connection. Use this argument to view connection details for a specific discussions server connection.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example lists the names of all of the discussions server connections that are currently configured for WebCenter Portal:

```
wls:/weblogic/serverConfig>listDiscussionForumConnections(appName='webcenter')
```

The following example lists connection names and details for all of the discussions server connections currently configured for WebCenter Portal:

```
wls:/weblogic/serverConfig>listDiscussionForumConnections(appName='webcenter', verbose=1)
```

The following example lists connection details for a discussions server connection named `myDiscussionsServer`.

```
wls:/weblogic/serverConfig>listDiscussionForumConnections(appName='webcenter', name='myDiscussionsServer')
```

2.7.7 setDefaultDiscussionForumConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Specifies the *default* discussions server connection for discussions and announcements in a named application.

While you can register multiple discussions server connections with an application, discussions and announcements only uses one connection—this is known as the default (or active) connection.

Syntax

```
setDefaultDiscussionForumConnection(appName, name, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Name of an existing discussions server connection.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example makes a connection named `myDiscussionServer` the default (or active) connection for discussions and announcements in WebCenter Portal:

```
wls:/weblogic/serverConfig> setDefaultDiscussionForumConnection
(appName='webcenter', name='myDiscussionServer')
```

2.7.8 setDiscussionForumServiceProperty

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Specifies default values for discussions, for a named application.

Configurable properties for discussions are listed in [Table 2-10](#).

Table 2-10 Discussions - Configurable Properties

Configuration Property	Description
<i>topics.fetch.size</i>	Maximum number of topics fetched by discussions and displayed in the topics view.
<i>forums.fetch.size</i>	Maximum number of forums fetched by discussions and displayed in the forums view.
<i>recentTopics.fetch.size</i>	Maximum number of topics fetched by discussions and displayed in the recent topics view.
<i>watchedTopics.fetch.size</i>	Maximum number of topics fetched by discussions and displayed in the watched topics view.
<i>watchedForums.fetch.size</i>	Maximum number of forums fetched by discussions and displayed in the watched forums view.
<i>application.root.category.id</i>	Application root category ID on the discussions server under which all discussion forums are stored. For example, if set to 3, all forums are stored inside category 3.

Table 2-10 (Cont.) Discussions - Configurable Properties

Configuration Property	Description
ForumGatewayManager.AUTO_START	<p>Communication through mail distribution lists can be published as discussion forum posts. This parameter starts or stops the gateway for this communication.</p> <p>For WebCenter Portal, the default value is 1, which means that as soon as you configure mail server settings through administration, the gateway starts. Set this to 0, and restart the managed server, to stop the gateway and disable this feature.</p>

Syntax

```
setDiscussionForumServiceProperty(appName, property, value, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>property</i>	Name of the configuration property.
<i>value</i>	Value for the property.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example changes the default number of topics displayed in topics view.

```
wls:/weblogic/serverConfig>setDiscussionForumServiceProperty  
(appName='webcenter', property='topics.fetch.size', value='30')
```

2.7.9 removeDiscussionForumServiceProperty

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Removes the current value that is set for a discussions property. Use this command to remove any of the properties listed in [Table 2-10](#).

Take care when using this command as removing values for these properties might cause unexpected behavior.

 **Note:**

Use this command syntax to disable the connection currently used for discussion and announcement services:

```
removeDiscussionForumServiceProperty('appName='webcenter',
property='selected.connection')
```

This command forces the default connection argument to 0. See also, [setDiscussionForumConnection](#) .

Syntax

```
removeDiscussionForumServiceProperty(appName, property, [server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>property</i>	Name of the configuration property.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example clears the current `topics.fetch.size` property for discussions in WebCenter Portal:

```
wls:/weblogic/serverConfig> removeDiscussionForumServiceProperty
(appName='webcenter', property='topics.fetch.size')
```

2.7.10 listDiscussionForumServiceProperties

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists all configurable properties for discussions.

Syntax

```
listDiscussionForumServiceProperties(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example lists configuration properties for discussions in WebCenter Portal:

```
wls:/weblogic/serverConfig> listDiscussionForumServiceProperties(appName='webcenter')
```

2.7.11 setAnnouncementServiceProperty

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Specifies default values for announcements in a named application.

Configurable properties for announcements are listed in [Table 2-11](#).

Table 2-11 Announcements - Configurable Properties

Configuration Property	Description
<i>miniview.page_size</i>	Maximum number of announcements displayed in the announcements mini view.
<i>mainview.page_size</i>	Maximum number of announcements displayed in the announcements main view.
<i>linksview.page_size</i>	Maximum number of announcements displayed in the announcements links view.
<i>announcements.expiration.days</i>	Number of days that announcements display and remain editable.

Syntax

```
setAnnouncementServiceProperty(appName, property, value, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>property</i>	Name of the configuration property.

Argument	Definition
<i>value</i>	Property value.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example changes the default number of days that announcements display in WebCenter Portal:

```
wls:/weblogic/serverConfig>setAnnouncementServiceProperty(appName='webcenter',
property='announcements.expiration.days', value='21')
```

2.7.12 removeAnnouncementServiceProperty

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Removes the current value that is set for an announcements property. Use this command to remove any of the properties listed in [Table 2-11](#).

Take care when using this command as removing values for these properties might cause unexpected behavior.

Syntax

```
removeAnnouncementServiceProperty(appName, property, [server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>property</i>	Name of the configuration property.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example clears the `announcements.expiration.days` property for announcements in WebCenter Portal:

```
wls:/weblogic/serverConfig>removeAnnouncementServiceProperty
(appName='webcenter', property='announcements.expiration.days')
```

2.7.13 listAnnouncementServiceProperties

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists all configurable properties for announcements, in a named application.

Syntax

```
listAnnouncementServiceProperties(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example lists configuration properties for announcements in WebCenter Portal:

```
wls:/weblogic/serverConfig>listAnnouncementServiceProperties(appName='webcenter')
```

2.7.14 addDiscussionsServerAdmin

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Grants system administrator permissions on the discussions server to a user or a group. This command is useful when you connect the discussions server to a new identity store that does not contain any of the current administrators.

Syntax

```
addDiscussionsServerAdmin(appName, name, [type, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the discussions server application in which to perform this operation. For example, <code>owc_discussions</code> .
<i>name</i>	Name of the user or group to add as an administrator on the discussions server.

Argument	Definition
<i>type</i>	Optional. Identifies the type of identity. Valid values are USER and GROUP. The default value is USER.
<i>server</i>	Optional. Name of the managed server on which the application is deployed. For example, WC_Collaboration. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example grants system administrator permissions on the discussions server to the user `weblogic`:

```
addDiscussionsServerAdmin(appName='owc_discussions', name='weblogic',
type='USER')
```

The following example grants system administrator permissions on the discussions server to all users in the `Administrators` user group:

```
addDiscussionsServerAdmin(appName='owc_discussions', name='Administrators',
type='GROUP')
```

2.7.15 addDiscussionsCategoryAdmin

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Grants category administrator permissions on the discussions server to a user or a group for a specific category ID.

Syntax

```
addDiscussionsCategoryAdmin(appName, categoryId, name, [type, server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the discussions server application in which to perform this operation. For example, <code>owc_discussions</code> .
<i>categoryId</i>	The ID (number) of a category on the discussions server.
<i>name</i>	Name of the user or group to add as an administrator for the category on the discussions server.
<i>type</i>	Optional. Identifies the type of identity. Valid values are USER and GROUP. The default value is USER.

Argument	Definition
<i>server</i>	Optional. Name of the managed server on which the application is deployed. For example, WC_Collaboration. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example grants category administrator permissions on the discussions server to the user `weblogic` for a category with an ID=2:

```
addDiscussionsCategoryAdmin(appName='owc_discussions', categoryId=2, name='weblogic',
type='USER')
```

The following example grants category administrator permissions on the discussions server to all users in the `Sales` user group for a category with an ID=2:

```
addDiscussionsCategoryAdmin(appName='owc_discussions', categoryId=2, name='Sales',
type='GROUP')
```

2.7.16 setDiscussionsServerProperty

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Sets a discussions server property. Use this command to set a system property on the discussions server.

Note:

To execute discussions server WLST commands, such as `setDiscussionsServerProperty`, the user used to connect to the Admin Server must also have administrative privileges on the discussions server.

Syntax

```
setDiscussionsServerProperty(appName, key, value, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the discussions server application in which to perform this operation. For example, <code>owc_discussions</code> .
<i>key</i>	Name of the discussions server property. For example, <code>owc_discussions.sso.mode</code> , <code>AuthFactory.className</code> , <code>UserManager.className</code> , and <code>GroupManager.className</code> .

Argument	Definition
<i>value</i>	Value for the discussions server property.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Collaboration. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example sets properties that configures the discussions server for SSO, where `example.com:8890/owc_discussions` is the base URL of the webtier on which the discussions server is deployed:

```
wls:/weblogic/
serverConfig>setDiscussionsServerProperty(appName='owc_discussions',
key='owc_discussions.sso.mode', value='true')

wls:/weblogic/
serverConfig>setDiscussionsServerProperty(appName='owc_discussions',
key='jiveURL', value='example.com:8890/owc_discussions')
```

2.7.17 getDiscussionsServerProperty

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Returns the current value of a discussions server property.

 **Note:**

To execute discussions server WLST commands, such as `getDiscussionsServerProperty`, the user used to connect to the Admin Server must also have administrative privileges on the discussions server.

Syntax

```
getDiscussionsServerProperty(appName, key, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the discussions server application in which to perform this operation. For example, <code>owc_discussions</code> .

Argument	Definition
<i>key</i>	Name of the discussions server property. For example, <code>owc_discussions.sso.mode</code> , <code>AuthFactory.className</code> , <code>UserManager.className</code> , and <code>GroupManager.className</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Collaboration</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following examples return the current value for some key discussions server properties:

```
wls:/weblogic/serverConfig>getDiscussionsServerProperty(appName='owc_discussions',
key='AuthFactory.className')
```

```
wls:/weblogic/serverConfig>getDiscussionsServerProperty(appName='owc_discussions',
key='UserManager.className')
```

```
wls:/weblogic/serverConfig>getDiscussionsServerProperty(appName='owc_discussions',
key='GroupManager.className')
```

2.7.18 removeDiscussionsServerProperty

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Removes the current value that is set for a discussions server property.

Note:

To execute discussions server WLST commands, such as `removeDiscussionsServerProperty`, the user used to connect to the Admin Server must also have administrative privileges on the discussion server.

Syntax

```
removeDiscussionsServerProperty(appName, key, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the discussions server application in which to perform this operation. For example, <code>owc_discussions</code> .

Argument	Definition
<i>key</i>	Name of the discussions server property. For example, <code>owc_discussions.sso.mode</code> , <code>AuthFactory.className</code> , <code>UserManager.className</code> , and <code>GroupManager.className</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Collaboration</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example removes the current value for the 'SSO mode' property on the discussions server:

```
wls:/weblogic/
serverConfig>removeDiscussionsServerProperty(appName='owc_discussions',
key='owc_discussions.sso.mode')
```

2.8 External Applications

Use the commands listed in [Table 2-12](#) to manage external application connections for WebCenter Portal.

Configuration changes made using these WLST commands are immediately available in the application, that is, you do not need to restart the managed server.

Table 2-12 External Application WLST Commands

Use this command...	To...	Use with WLST...
createExtAppConnection	Create an external application connection, for a named application.	Online
setExtAppConnection	Edit an existing external application connection.	Online
listExtAppConnections	List individual or all external applications that are configured for a named application.	Online
addExtAppField	Add another login field for a specific external application connection.	Online
setExtAppField	Edit the value and display-to-user setting for a specific external application login field.	Online
removeExtAppField	Remove an external application login field.	Online
addExtAppCredential	Specify shared or public credentials for an external application.	Online
setExtAppCredential	Edit shared or public credentials for an external application.	Online
removeExtAppCredential	Remove shared or public credentials currently configured for an external application.	Online

2.8.1 createExtAppConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Creates an external application connection for a named application.

Syntax

```
createExtAppConnection(appName, name, [displayName, url, authMethod,
userFieldName, pwdFieldName, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Connection name. The name must be unique (across all connection types) within WebCenter Portal.
<i>displayName</i>	Optional. External application display name. A user friendly name for the external application that WebCenter Portal users will recognize. The display name must be unique across all external applications within WebCenter Portal.
<i>url</i>	Optional. External application login URL. To determine an application's URL, navigate to the application's login page and note down the URL for that page. For example: <code>http://login.yahoo.com/config/login</code>
<i>authMethod</i>	Optional. Authentication mechanism used by the external application. Valid options are GET, POST, and BASIC. This argument defaults to POST.
<i>userFieldName</i>	Optional. Name that identifies the <i>user name</i> or <i>user ID</i> field on the external application's login form. To find this name, look at the HTML source for the login page. This argument does not specify user credentials. Mandatory if creating an automated ADF external application login.
<i>pwdFieldName</i>	Optional. Name that identifies the <i>password</i> field on the external application's login form. To find this name, look at the HTML source for the login page. This argument does not specify user credentials. Mandatory if creating an automated ADF external application login.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example creates a connection for an external application named `My Yahoo!`, in WebCenter Portal (`webcenter`).

```
wls:/weblogic/serverConfig> createExtAppConnection(appName='webcenter',
name='yahoo', displayName='My Yahoo!', url='http://login.yahoo.com/config/
login',
authMethod='POST', userFieldName='login', pwdFieldName='password')
```

2.8.2 setExtAppConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Edits an existing external application connection.

Syntax

```
setExtAppConnection(appName, name, [displayName, url, authMethod,
userFieldName, pwdFieldName, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>name</i>	Name of an existing external application connection.
<i>displayName</i>	Optional. External application display name. A user-friendly name for the external application that WebCenter Portal users will recognize. The display name must be unique across all external applications within WebCenter Portal.
<i>url</i>	Optional. External application login URL. To determine an application's URL, navigate to the application's login page and note down the URL for that page.
<i>authMethod</i>	Optional. Authentication mechanism used by the external application. Valid options are GET, POST, and BASIC. This argument defaults to POST.
<i>userFieldName</i>	Optional. Name that identifies the <i>user name</i> or <i>user ID</i> field on the external application's login form. To find this name, look at the HTML source for the login page. This argument does not specify user credentials. Mandatory if <i>authMethod</i> is GET or POST and a login URL is specified but can be left blank if BASIC authentication method is selected.
<i>pwdFieldName</i>	Optional. Name that identifies the <i>password</i> field on the external application's login form. To find this name, look at the HTML source for the login page. This argument does not specify user credentials. Mandatory if <i>authMethod</i> is GET or POST, but can be left blank if BASIC authentication method is selected.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example updates the display name attribute for an external application named yahoo.

```
wls:/weblogic/serverConfig> setExtAppConnection(appName='webcenter',
name='yahoo', displayName='My Favorite Yahoo!')
```

2.8.3 listExtAppConnections

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

When used with only the `appName` argument, this command lists the names of all the external applications currently configured for a named WebCenter Portal application.

Syntax

```
listExtAppConnections(appName, [verbose, name, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>verbose</i>	Optional. Displays external application details in verbose mode. Valid options are 1 (true) and 0 (false). When set to 1, <code>listExtAppConnections</code> lists all of the external applications that are configured for WebCenter Portal, along with their details. When set to 0, <code>listExtAppConnections</code> lists only the names of the external applications. This argument defaults to 0. If you set this argument to 0, do not specify the <code>name</code> argument.
<i>name</i>	Optional. Name of an existing external application connection. You can use this argument to view details about a specific connection.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example lists the names of all the external applications currently used by WebCenter Portal (`webcenter`):

```
wls:/weblogic/serverConfig> listExtAppConnections(appName='webcenter')
app1
app2
app3
```

The following example lists details for the external applications `app1`, `app2`, and `app3`.

```
wls:/weblogic/serverConfig> listExtAppConnections(appName='webcenter', verbose=1)
----
app1
----
Name: app1
Display Name: Application1
Login URL: http://app1
Authentication Method: POST
User Field Name: login
Password Field Name: passwd
Shared Credential: Disabled
Public Credential: Disabled
----
app2
----
Name: app2
Display Name: Application2
Login URL: http://app2
Authentication Method: POST
User Field Name: login
Password Field Name: passwd
Additional Fields: {Account1:1, Account2:DefVal:0}
Shared Credential: Disabled
Public Credential: Enabled
----
app3
----
Name: app3
Display Name: Application3
Authentication Method: POST
Shared Credential: Enabled
Public Credential: Enabled
```

The following example lists details for external application `app1` only.

```
wls:/weblogic/serverConfig> listExtAppConnections(appName='webcenter',
verbose=1, name='app1')
----
app1
----
Name: app1
Display Name: Application1
Login URL: http://app1
Authentication Method: POST
User Field Name: login
Password Field Name: passwd
Shared Credential: Disabled
Public Credential: Disabled
```

2.8.4 addExtAppField

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Adds another login field for a specific external application connection. For example, in addition to user name and password, an external application may require other login criteria such as `Host` and `MailAddress`.

Optionally, additional login fields can appear on the external application's login for a user to specify.

If you add another login field *and* the external application uses shared or public credentials, you can use the WLST commands `addExtAppCredential` and `setExtAppCredential` to update the shared/public credentials. See [addExtAppCredential](#) and [setExtAppCredential](#).

Syntax

```
addExtAppField(appName, name, fieldName, [fieldValue, displayToUser, server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Name of an existing external application connection.
<i>fieldName</i>	Login field name. The name that identifies the field on the HTML login form. This field is not applicable if the application uses BASIC authentication.
<i>fieldValue</i>	Optional. Login field value. Enter a default value for the login field or leave blank for a user to specify. This argument is blank by default.
<i>displayToUser</i>	Optional. Specifies whether the login field displays on the external application's login screen. Valid options are 1 (true) and 0 (false). This argument defaults to 0. Note that if you set this argument to 0, you must specify the <code>fieldValue</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

This example creates an additional field named `Account` with the default value `username.default.example` in an external application called `ABC`. This field will be displayed on `ABC`'s login screen.

```
wls:/weblogic/serverConfig> addExtAppField(appName='webcenter', name='ABC',
fieldName='Account', fieldValue='username.default.example', displayToUser=1)
```

2.8.5 setExtAppField

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Modifies the field value and display-to-user setting for one or more login fields currently configured for an external application. Either `fieldValue` or `displayToUser` must be specified along with the external application name and login field name. The `fieldValue` and `displayToUser` arguments are optional.

Using this command has implications on any shared or public credentials that you might have created for this external application. If you modify `displayToUser` to 1, you may also need to update existing shared user or public user credentials. See also [setExtAppCredential](#).

Syntax

```
setExtAppField(appName, name, fieldName, [fieldValue, displayToUser, server, applicationVersion])
```

Argument	Definition
<code>appName</code>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<code>name</code>	Name of an existing external application connection.
<code>fieldName</code>	Name of an existing login field.
<code>fieldValue</code>	Optional. New or changed login field value. Enter a default value for the login field or leave blank for a user to specify. This argument is blank by default.
<code>displayToUser</code>	Optional. Specifies whether the login field displays on the external application's login screen. Valid options are 1 (true) and 0 (false). If set to 0, <code>fieldValue</code> must be specified.
<code>server</code>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<code>applicationVersion</code>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example specifies a default value for a login field named `Account` and displays the field on the external application's credential provisioning screen:

```
wls:/weblogic/serverConfig> setExtAppField(appName='webcenter', name='ABC',
fieldName='Account', fieldValue='admin', displayToUser=1)
```

2.8.6 removeExtAppField

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Removes a login field from an external application connection.

This command has implications on any shared or public credentials that you may have created for this external application, that is, you may need to remove the login field from shared user or public user credentials.

You can use the `setExtAppCredential` command to remove a login field, if required. For example, external application `myApp` has an additional field called `Account` and public credentials were previously specified using:

```
addExtAppCredential(appName='webcenter', name='myApp', type='PUBLIC',
username='admin', password='mypublic.password', field='Account:admin@myhost.com')
```

If you remove the `Account` field, you can modify the credentials by running:

```
setExtAppCredential(appName='webcenter', name='myApp', type='PUBLIC',
username='admin', password='mypublic.password')
```

For details on using `setExtAppCredential`, see [setExtAppCredential](#)

Syntax

```
removeExtAppField(appName, name, fieldName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Connection name.
<i>fieldName</i>	Login field that you want to remove.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example removes the additional login field named `Account` from an external application named `ABC`.

```
wls:/weblogic/serverConfig> removeExtAppField(appName='webcenter', name='ABC',
fieldName='Account')
```

2.8.7 addExtAppCredential

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Configures shared user or public user credentials for a specific external application.

When shared credentials are specified, every user accessing WebCenter Portal, is authenticated using the user name and password defined here. WebCenter Portal users are not presented with a login form.

Public users accessing this external application through WebCenter Portal are logged in using the public credentials defined here.

If credentials already exists, a warning indicates that the `setExtAppCredential` command should be used instead.

Syntax

```
addExtAppCredential(appName, name, type, username, password, [field, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Name of an existing external application connection.
<i>type</i>	Credential type. Valid values are <code>SHARED</code> and <code>PUBLIC</code> .
<i>username</i>	Name of the shared or public user.
<i>password</i>	Password for the shared or public user.
<i>field</i>	Optional. Additional login field value. Use the format <code>FieldName:FieldValue</code> , where <code>FieldName</code> names an additional login field configured with <code>displayToUser=1</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example specifies public credentials for an external application named ABC. The public user name is `mypublic.username`, the password is `mypublic.password`, and there is one additional field named `Account`.

```
wls:/weblogic/serverConfig> addExtAppCredential(appName='webcenter', name='ABC',
type='PUBLIC', username='mypublic.username', password='mypublic.password',
field='Account:username.example')
```

2.8.8 setExtAppCredential

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Modifies shared user or public user credentials currently configured for an external application. If the credential has already not been specified, then a warning indicates that `addExtAppCredential` needs to be used instead. See [addExtAppCredential](#).

The arguments `username` and `password` are optional because `setExtAppCredential` only manipulates existing credentials. At least one of the parameters, `username`, `password` or `field`, must be specified.

You can use `setExtAppCredential` command to update passwords in systems that require changing passwords every few days.

Syntax

```
setExtAppCredential(appName, name, type, [username, password, field,
server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Name of an existing external application connection.
<i>type</i>	Credential type. Valid values are <code>SHARED</code> and <code>PUBLIC</code> .
<i>username</i>	Optional. User name of the shared or public user.
<i>password</i>	Optional. Password for the shared or public user.
<i>field</i>	Optional. Additional login field value. Use the format <code>FieldName:FieldValue</code> , where <code>FieldName</code> names an additional login field configured with <code>displayToUser=1</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example changes the public user's login credentials for an external application named `ABC`.

```
wls:/weblogic/serverConfig> setExtAppCredential(appName='webcenter',name='ABC',
type='PUBLIC', username='username.example', password='password.example',
field='Account:username.example')
```

2.8.9 removeExtAppCredential

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Removes shared user or public user credentials currently configured for an external application.

If credentials do not exist, an error displays.

Syntax

```
removeExtAppCredential(appName, name, type, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Name of an existing external application connection.
<i>type</i>	Credential type. Valid values are <code>SHARED</code> and <code>PUBLIC</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example removes shared credentials specified for an external application named ABC.

```
wls:/weblogic/serverConfig> removeExtAppCredential(appName='webcenter',
name='ABC', type='SHARED')
```

2.9 Instant Messaging and Presence

Use the commands listed in [Table 2-13](#), to manage instant messaging and presence server connections.

Configuration changes made using these WLST commands are only effective after your restart the Managed Server on which WebCenter Portal is deployed. For details, see *Oracle Fusion Middleware Administering Oracle WebCenter Portal*.

Table 2-13 Instant Messaging and Presence WLST Commands

Use this command...	To...	Use with WLST...
createIMPConnection	Create a new instant messaging and presence server connection for a named application.	Online
setIMPConnection	Edit an existing instant messaging and presence server connection.	Online
setIMPConnectionProperty	Modify instant messaging and presence server connection properties.	Online
deleteIMPConnectionProperty	Delete an instant messaging and presence server connection property.	Online

Table 2-13 (Cont.) Instant Messaging and Presence WLST Commands

Use this command...	To...	Use with WLST...
listIMPAdapters	List which presence servers a named application supports.	Online
listIMPConnections	List all of the instant messaging and presence server connections that are configured for a named application.	Online
listDefaultIMPConnection	List the default instant messaging and presence server connection that is configured for a named application.	Online
setDefaultIMPConnection	Set a specified connection as the default instant messaging and presence server connection.	Online
setIMPServiceProperty	Specify defaults for instant messaging and presence.	Online
removeIMPServiceProperty	Remove defaults for instant messaging and presence.	Online
listIMPServiceProperties	List instant messaging and presence properties.	Online
createIMPExtAppConnection	Create an external application suitable for instant messaging and presence server connection.	Online

2.9.1 createIMPConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Creates an instant messaging and presence server connection for a named application.

Use the [listIMPAdapters](#) command to find out which types of instant messaging and presence servers are supported. Out-of-the-box, WebCenter Portal supports Microsoft Office Communications Server 2007 SP1 (OCS) and Microsoft Lync 2010.

While you can register multiple presence server connections for WebCenter Portal, only one connection is used for instant messaging and presence services—the default (or active) connection.

Syntax

```
createIMPConnection(appName, name, adapter, url, [appId, poolName,
userDomain, timeout, default, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Connection name. The name must be unique (across all connection types) within the application.

Argument	Definition
<i>adapter</i>	Adapter name. Specify the adapter that matches your instant messaging and presence server. The valid value is OCS2007. Choose OCS2007 for Microsoft Office Communications Server 2007 and Microsoft Lync.
<i>url</i>	URL of the sever hosting instant messaging and presence services. For example: <code>http://myocshost.com:8888</code>
<i>domain</i>	Deprecated. Use the setIMPServiceProperty command to resolve IM addresses.
<i>appId</i>	Optional. External application associated with the presence server connection. If specified, external application credential information is used to authenticate users against the OCS or Lync server. This argument is mandatory for OCS and Lync server connections. The external application you configure for instant messaging and presence services must use <code>authMethod=POST</code> , and specify an additional field with <code>fieldName='Account'</code> and <code>displaytoUser=1</code> . If an external application does not exist yet, use the WLST command createIMPEExtAppConnection to create an external application that automatically has all the required additional fields. See also addExtAppField and setExtAppField .
<i>poolName</i>	Optional. Pool name that is required to create an OCS or Lync connection. Refer to <i>Microsoft Office Communications Server</i> , or Microsoft Lync Server documentation for details on pool names. This argument is mandatory for OCS and Lync server connections.
<i>userDomain</i>	Active Directory domain on the OCS/Lync server. This argument is mandatory for OCS/Lync server connections.
<i>timeout</i>	Optional. Length of time (in seconds) that instant messaging and presence waits for a response from the presence server before issuing a connection timeout message. This argument defaults to -1. When set to -1, the service default (10 seconds) applies.
<i>default</i>	Optional. Indicates whether this connection is the default connection for instant messaging and presence services. Valid values are 1 (true) and 0 (false). The default for this argument is 0.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example creates an instant messaging and presence server connection to a Microsoft Office Communications Server named `myOCSPresenceServer`.

```
wls:/weblogic/serverConfig>createIMPConnection(appName='webcenter',
name='myOCSPresenceServer', adapter='OCS2007', url='http://myocshost.com/owc/
ocs',
appId='OCSExtApp', userDomain='OCS', poolName='pool01.myocshost.com',
timeout=60, default=1)
```

The following example creates an instant messaging and presence server connection to a Microsoft Lync Server named `myLyncServer`.

```
wls:/weblogic/serverConfig>createIMPConnection(appName='webcenter',
name='myLyncServer', adapter='OCS2007',
url='http://mylynchost.com:8888' appId='LyncExtApp', userDomain='LYNC',
poolName='pool105.mylynchost.com', timeout=60, default=1))
```

2.9.2 setIMPConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Edits an existing instant messaging and presence server connection. Use this command to update connection attributes.

The connection is created using the [createIMPConnection](#) command.

Syntax

```
setIMPConnection(appName, name, [adapter, url, appId, poolName,
userDomain, timeout, default, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Name of an existing presence server connection.
<i>adapter</i>	Optional. Adapter name. Specify the adapter that matches your instant messaging and presence server. Valid value is <code>OCS2007</code> . Choose <code>OCS2007</code> for Microsoft Office Communications Server and Microsoft Lync Server.
<i>url</i>	Optional. URL of the server hosting instant messaging and presence services.
<i>domain</i>	Deprecated. Use the setIMPServiceProperty command to resolve IM addresses.
<i>appId</i>	External application associated with the presence server connection. If specified, external application credential information is used to authenticate users against the OCS or Lync server. This argument is mandatory for OCS and Lync server connections. The external application you configure for instant messaging and presence services must use <code>authMethod=POST</code> , and specify an additional field with <code>fieldName='Account'</code> and <code>displaytoUser=1</code> . If an external application does not exist yet, use the WLST command createIMPExtAppConnection to create an external application that automatically has all the required additional fields. See also addExtAppField and setExtAppField .
<i>poolName</i>	Pool name that is required to create an OCS or Lync connection. Refer to <i>Microsoft Office Communications Server</i> or <i>Microsoft Lync Server</i> documentation for details on pool names. This argument is mandatory for OCS, and Lync server connections.

Argument	Definition
<i>userDomain</i>	Active Directory domain on the OCS server. This argument is mandatory for OCS/Lync server connections.
<i>timeout</i>	Optional. Length of time (in seconds) that instant messaging and presence waits for a response from the presence server before issuing a connection timeout message. This argument defaults to -1. When set to -1, the service default (10 seconds) applies.
<i>default</i>	Optional. Indicates whether this connection is the default connection for instant messaging and presence services. Valid values are 1 (true) and 0 (false). The default for this argument is 0. To specify that instant messaging and presence uses this connection, change the value from 0 to 1. To disable this connection, use the removeIMPServiceProperty command: <pre>removeIMPServiceProperty('appName='webcenter', property='selected.connection')</pre> While you can register multiple presence server connections for an application, only one connection is used for instant messaging and presence services—the default (or active) connection.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example sets attributes on an existing instant messaging and presence server connection.

```
wls:/weblogic/serverConfig>setIMPConnection(appName='webcenter',  
name='myOCSPresenceServer', adapter='OCS2007', url='http://myocshost.com/owc/  
ocs',  
timeout=120, default=1)
```

2.9.3 setIMPConnectionProperty

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Sets an instant messaging and presence server connection property. Use this command if additional parameters are required to connect to your presence server. This is an extensible way to add any connection property using a key and a value. (You are not limited to connection properties specified by [createIMPConnection](#) and [setIMPConnection](#).)

All known, additional connection properties are listed in [Table 2-14](#).

Table 2-14 Additional Instant Messaging and Presence Connection Properties

Additional Connection Property	Description
<code>presence.url</code>	URL to the presence server. This must be supplied if presence is deployed on a separate server.
<code>contacts.url</code>	URL to the contact management service. This must be supplied if the contact management service is deployed on a separate server.
<code>call.url</code>	URL for the third-party call server. If no value is supplied, then this uses the same value as <code>base.connection.url</code> .
<code>call.method</code>	Supports values <code>sip</code> and <code>pstn</code> : When set to <code>sip</code> , the IMP service forwards the user's SIP address to the third-party call service. The third-party call service must decide on the routing of the call. When set to <code>pstn</code> , the user's phone number is based on the user's profile attribute (<code>BUSINESS_PHONE</code>). You can use the connection property <code>call.number.attribute</code> to change this default profile attribute (<code>BUSINESS_PHONE</code>) to any other attribute.
<code>call.domain</code>	The domain name of the <code>pstn</code> gateway. If no domain name is supplied, then this uses the domain value specified when the connection was created. Supply a domain name only when <code>call.method</code> is set to <code>pstn</code> .
<code>contact.number.attribute</code>	The attribute used to read users' phone numbers from the user profile. The default is <code>BUSINESS_PHONE</code> . Supply this attribute value only when <code>call.method</code> is set to <code>pstn</code> .
<code>primary.domain</code>	If the WebCenter user identity is qualified with a domain (for example, <code>john.doe@oracle.com</code>), and if the presence server domain is different (for example, <code>john.doe@example.com</code>) then specify the primary domain <code>oracle.com</code> here. If the user identity is qualified with a domain and the presence server uses the same <code>oracle.com</code> domain, then it is not necessary that you specify the <code>primary.domain</code> .

 **Note:**

Do not use the [setIMPConnectionProperty](#) to set connection properties available through [createIMPConnection](#) or [setIMPConnection](#). Attempting to do so has no effect.

Syntax

```
setIMPConnectionProperty(appName, name, key, value, [secure, server, applicationVersion])
```

Argument	Definition
<code>appName</code>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .

Argument	Definition
<i>name</i>	Name of an existing presence server connection.
<i>key</i>	Name of the connection property. See Table 2-14 .
<i>value</i>	Value for the property. Allows any property to be modified on the connection with a key and value.
<i>secure</i>	Optional. Indicates whether the property value must be stored securely using encryption. Valid options are 1 (true) and 0 (false). When 1, the value is encrypted. The default option is 0.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example adds a custom instant messaging and presence server connection property called `admin.user` with a default value `admin`:

```
wls:/weblogic/serverConfig>setIMPConnectionProperty(appName='webcenter',
name='MyLCSPresenceServer', key='admin.user', value='admin')
```

2.9.4 deleteIMPConnectionProperty

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Deletes an instant messaging and presence server connection property. Use caution when deleting connection properties because the connection might not work as expected if the configuration becomes invalid as a result.

This command can only delete *additional* connection properties added using the `setIMPConnectionProperty` command.

Syntax

```
deleteIMPConnectionProperty(appName, name, key, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Name of an existing presence server connection.
<i>key</i>	Name of the connection property you want to delete.

Argument	Definition
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example deletes an instant messaging and presence server connection property named `admin.user`.

```
wls:/weblogic/serverConfig>deleteIMPConnectionProperty(appName='webcenter',
name='MyOCSPresenceServer', key='admin.user')
```

2.9.5 listIMPAdapters

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists which types of instant messaging and presence servers Oracle WebCenter Portal supports. Out-of-the-box, WebCenter Portal supports Microsoft Office Communications Server 2007 SP1 (OCS) and Microsoft Lync 2010.

Syntax

```
listIMPAdapters()
```

Example

The following example lists which presence servers are supported:

```
wls:/weblogic/serverConfig>listIMPAdapters()
```

2.9.6 listIMPConnections

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists all of the instant messaging and presence server connections that are configured for a named application.

Syntax

```
listIMPConnections(appName,[verbose, name, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .

Argument	Definition
<i>verbose</i>	Optional. Displays presence server connection details in verbose mode. Valid values are 1 (true) and 0 (false). When set to 1, <code>listIMPConnections</code> lists all of the presence server connections that are configured for the named application, along with their details. When set to 0, only connection names are listed. This argument defaults to 0.
<i>name</i>	Optional. Name of an existing presence server connection. Use this argument to view connection details for a specific presence server connection. Note that if you use the <code>name</code> argument when <code>verbose</code> argument set to 1, the <code>verbose</code> argument is ignored.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example lists all the instant messaging and presence server connections that are configured for WebCenter Portal (`webcenter`):

```
wls:/weblogic/serverConfig>listIMPConnections(appName='webcenter')
```

The following example lists all the instant messaging and presence server connections that are configured for WebCenter Portal in *verbose* mode:

```
wls:/weblogic/serverConfig>listIMPConnections(appName='webcenter', verbose=1)
```

The following example lists connection details for an instant messaging and presence server connections named `impConnection1`.

```
wls:/weblogic/serverConfig> listIMPConnections(appName='webcenter',
name='impConnection1')
```

2.9.7 listDefaultIMPConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists the connection that instant messaging and presence is using, in a named application. While you can register multiple presence server connections for WebCenter Portal, instant messaging and presence only uses one connection—the default (or active) connection.

If only one presence server connection is available, that connection is assumed to be the default connection.

Syntax

```
listDefaultIMPConnection(appName, verbose, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>verbose</i>	Optional. Displays the default presence server connection in verbose mode, if available. Valid options are 1 (true) and 0 (false). When set to 1, the name and details of the presence server connection are listed. When set to 0, only the connection name displays. This argument defaults to 0.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example lists the name and details of the connection that instant messaging and presence is using in WebCenter Portal (`webcenter`):

```
wls:/weblogic/serverConfig>listDefaultIMPConnection(appName='webcenter', verbose=1)
```

2.9.8 setDefaultIMPConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Specifies the *default* connection for instant messaging and presence in a named application. While you can register multiple presence server connections with an application, instant messaging and presence only uses one connection—the default (or active) connection.

If only one presence server connection is available, that connection is assumed to be the default connection.

Syntax

```
setDefaultIMPConnection(appName, name, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Name of an existing instant messaging and presence connection.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.

Argument	Definition
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example makes a connection named `myPresenceServer` the default (or active) connection for instant messaging and presence in WebCenter Portal (`webcenter`):

```
wls:/weblogic/serverConfig>setDefaultIMPConnection(appName='webcenter',
name='myPresenceServer')
```

2.9.9 setIMPServiceProperty

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Specifies default values for instant messaging and presence.

Configurable properties for instant messaging and presence are listed in [Table 2-15](#).

Table 2-15 Instant Messaging and Presence - Configurable Properties

Configuration Property	Description
<i>selected.connection</i>	Connection used by instant messaging and presence.
<i>rtc.cache.time</i>	Cache timeout for instant messaging and presence data. The default is 60 seconds.
<i>resolve.display.name.from.user.profile</i>	Determines what to display if user display names are missing. When set to 0, and display name information is unavailable, only the user name displays in the application. When set to 1, and display name information is unavailable, display names are read from user profile data. Setting this option to 1 will impact performance. The default setting is 0. Display names are not mandatory in presence data. If WebCenter Portal does not always provide display names by default and you consider this information important, set <code>resolve.display.name.from.user.profile</code> to 1 so that display names always display.
<i>im.address.resolver.class</i>	Resolver implementation used to map user names to IM addresses and IM addresses to user names. The default setting is <code>oracle.webcenter.collab.rtc.IMAddressResolverImpl</code> . This implementation looks for IM addresses in the following places and in the order specified: <ul style="list-style-type: none"> • User Preferences • User Credentials • User Profiles

Table 2-15 (Cont.) Instant Messaging and Presence - Configurable Properties

Configuration Property	Description
<i>im.address.profile.attribute</i>	User profile attribute used to determine a user's IM address. The default setting is BUSINESS_EMAIL.

Syntax

```
setIMPServiceProperty(appName, property, value, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>property</i>	Name of the configuration property.
<i>value</i>	Value for the property.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example changes the default cache timeout for instant messaging and presence data in WebCenter Portal (webcenter):

```
wls:/weblogic/serverConfig>setIMPServiceProperty(appName='webcenter',  
property='rtc.cache.time', value='30')
```

2.9.10 removeIMPServiceProperty

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Removes the current value that is set for an instant messaging and presence property. Use this command to remove any of the properties listed in [Table 2-15](#).

Take care when using this command as removing values for these properties might cause unexpected behavior.

 **Note:**

Use this command syntax to disable the connection currently used by instant messaging and presence:

```
removeIMPServiceProperty('appName='webcenter',
property='selected.connection')
```

This command forces the default connection argument to 0. See also, [setIMPConnection](#).

Syntax

```
removeIMPServiceProperty(appName, property, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>property</i>	Name of the configuration property.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example clears the cache expiration value for instant messaging and presence in WebCenter Portal:

```
wls:/weblogic/serverConfig>removeIMPServiceProperty(appName='webcenter',
property='rtc.cache.time')
```

2.9.11 listIMPServiceProperties

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists all configurable properties for instant messaging and presence.

Syntax

```
listIMPServiceProperties(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.

Argument	Definition
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example lists configuration properties for instant messaging and presence in WebCenter Portal (webcenter):

```
wls:/weblogic/serverConfig>listIMPServiceProperties(appName='webcenter')
```

2.9.12 createIMPEExtAppConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Creates an external application suitable for instant messaging and presence server connections. The external application is configured with the required additional properties: `authMethod=POST`, and additional fields `fieldName='Account'` and `displaytoUser=1`.

Syntax

```
createIMPEExtAppConnection(appName, name, [displayName, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Connection name. The name must be unique (across all connection types) within the application.
<i>displayName</i>	Optional. External application display name. A user friendly name for the application that users will recognize. The display name must be unique across all external applications within the application.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example creates an external application named `IMPxApp` suitable for instant messaging and presence server connections:

```
wls:/weblogic/serverConfig>createIMPExtAppConnection(appName='webcenter',
name='IMPxApp', displayName='IMP Ext App')
```

2.10 Mail

Use the commands listed in [Table 2-16](#) to manage mail server connections for a named application.

WebCenter Portal supports multiple mail connections. The mail connection configured with `default=1` is the default connection for mail services in WebCenter Portal. All additional connections are offered as alternatives; WebCenter Portal users can choose which one they want to use through user preferences.

Configuration changes made using these WLST commands are only effective after you restart the Managed Server on which the application is deployed. For details, see *Oracle Fusion Middleware Administering Oracle WebCenter Portal*.

Table 2-16 Mail WLST Commands

Use this command...	To...	Use with WLST...
createMailConnection	Create a mail server connection for a named application.	Online
setMailConnection	Edit an existing mail server connection.	Online
setMailConnectionProperty	Set mail server connection properties.	Online
deleteMailConnectionProperty	Delete a mail server connection property.	Online
listMailConnections	List all of the mail server connections that are configured for a named application.	Online
listDefaultMailConnection	List the default mail server connection that is configured for a named application.	Online
setDefaultMailConnection	Set a specified connection as the default mail server connection.	Online
setMailServiceProperty	Specify defaults for mail.	Online
removeMailServiceProperty	Remove defaults for mail.	Online
listMailServiceProperties	List properties for mail.	Online
createMailExtApp	Create an external application suitable for mail connections.	Online

2.10.1 createMailConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Creates a mail server connection for a named application.

WebCenter Portal supports the Microsoft Exchange Server or any mail server that supports IMAP4 and SMTP. The most important mail server connection attributes are: `imapHost`, `imapPort`, `imapSecured`, `smtpHost`, `smtpPort`, and `smtpSecured`

You can register multiple mail server connections. WebCenter Portal supports multiple mail connections. The mail connection configured with `default=1` is the default connection for mail services in WebCenter Portal. All additional connections are offered as alternatives; WebCenter Portal users can choose which one they want to use through user preferences.

Syntax

```
createMailConnection(appName, name, [imapHost, imapPort, smtpHost, smtpPort,
imapSecured, smtpSecured, appId, timeout, default, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Connection name. The name must be unique (across all connection types) within the application.
<i>imapHost</i>	Optional. Host name of the machine on which the IMAP service is running.
<i>imapPort</i>	Optional. Port on which the IMAP service listens.
<i>smtpHost</i>	Optional. Host name of the machine where the SMTP service is running.
<i>smtpPort</i>	Optional. Port on which the SMTP service listens.
<i>imapSecured</i>	Optional. Specifies whether the mail server connection to the IMAP server is SSL-enabled. Valid values are 1 (true) and 0 (false). The default for this argument is 0.
<i>smtpSecured</i>	Optional. Specifies whether the SMTP server is secured. Valid values are 1 (true) and 0 (false). The default for this argument is 0.
<i>appId</i>	External application associated with the mail server connection. External application credential information is used to authenticate users against the IMAP and SMTP servers. The same credentials are supplied to authenticate the user on both the IMAP and SMTP servers. The external application you configure for mail must use <code>authMethod=POST</code> , and specify several additional login fields: <code>fieldName='Email Address' and displaytoUser=1</code> <code>fieldName='Your Name' and displaytoUser=1</code> <code>fieldName='Reply-To Address' and displaytoUser=1</code> If an external application does not exist yet, use the WLST command createMailExtApp to create an external application that automatically has all the required additional fields. See also createExtAppConnection .
<i>timeout</i>	Optional. Length of time (in seconds) that the mail waits to acquire a connection before terminating. This argument defaults to -1. When set to -1, the service default (10 seconds) applies.

Argument	Definition
<i>default</i>	Optional. Indicates whether this connection is the default connection for mail. Valid values are 1 (true) and 0 (false). This argument defaults to 0. WebCenter Portal supports multiple mail connections. The mail connection configured with <code>default=1</code> is the default connection for mail services in WebCenter Portal. Additional connections, configured with <code>default=0</code> , are offered as alternatives; WebCenter Portal can choose which one they want to use through user preferences.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example creates an external application suitable for a mail server connection, and then creates a mail server connection named `myMailConnection` for WebCenter Portal (`webcenter`):

```
wls:/weblogic/serverConfig>createMailExtApp(appName='webcenter',
name='extApp_Mail', displayName='Mail Ext App')

wls:/weblogic/serverConfig> reateMailConnection(appName='webcenter' ,
name='myMailConnection' , imapHost='myimaphost.com', imapPort=143 ,
smtpHost='mysmtpost.com' , smtpPort=25 , imapSecured=0, smtpSecured=0,
appId='extApp_Mail', timeout=60, default=1)
```

2.10.2 setMailConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Edits an existing mail connection. Use this command to update connection attributes.

The connection is created using the [createMailConnection](#) command.

This command enables you to set additional, optional, LDAP server attributes that cannot be set using `createMailConnection`. When LDAP details are defined, mail creates, edits, and deletes portal distribution lists in WebCenter Portal. Portal distribution lists are named after their portal (excluding non-java identifiers) and assigned a domain (derived from the `domain` attribute, for example, `@mycompany.com`). If LDAP details are not provided, portal distribution lists are not created or maintained. The mail server must be a *Microsoft Exchange Server*.

Syntax

```
setMailConnection(appName, name, [imapHost, imapPort, smtpHost, smtpPort,
imapSecured, smtpSecured, appId, default, ldapHost, ldapPort, ldapBaseDN,
ldapAdminUser, ldapAdminPassword, ldapSecured, domain, defaultUser, timeout,
server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Name of an existing mail server connection.
<i>imapHost</i>	Optional. Host name of the machine on which the IMAP service is running.
<i>imapPort</i>	Optional. Port on which the IMAP service listens.
<i>smtpHost</i>	Optional. Host name of the machine where the SMTP service is running.
<i>smtpPort</i>	Optional. Port on which the SMTP service listens.
<i>imapSecured</i>	Optional. Specifies whether the connection to the IMAP server is secured (SSL-enabled). Valid values are 1 (true) and 0 (false). The default for this argument is 0.
<i>smtpSecured</i>	Optional. Specifies whether the connection to the SMTP server is secured (SSL-enabled). Valid values are 1 (true) and 0 (false). The default for this argument is 0.
<i>appId</i>	Optional. External application associated with the mail server connection. External application credential information is used to authenticate users against the IMAP and SMTP servers. The same credentials are supplied to authenticate the user on both the IMAP and SMTP servers. The external application you configure for mail must use <code>authMethod=POST</code> , and specify several additional login fields: <code>fieldName='Email Address' and displaytoUser=1</code> <code>fieldName='Your Name' and displaytoUser=1</code> <code>fieldName='Reply-To Address' and displaytoUser=1</code> If an external application does not exist yet, use the WLST command createMailExtApp to create an external application that automatically has all the required additional fields. See also createExtAppConnection .
<i>ldapHost</i>	Optional. Host name of the machine where the LDAP directory server is running.
<i>ldapPort</i>	Optional. Port on which the LDAP directory server listens.
<i>ldapBaseDN</i>	Optional. Base distinguished name for the LDAP schema. For example, <code>CN=Users,DC=oracle,DC=com</code> .
<i>ldapAdminUser</i>	Optional. User name of the LDAP directory server administrator. A valid administrator with privileges to make entries into the LDAP schema.
<i>ldapAdminPassword</i>	Optional. Password for the LDAP directory server administrator. This password will be stored in a secured store.
<i>ldapSecured</i>	Optional. Specifies whether the connection to the LDAP server is secured (SSL enabled). Valid values are 1 (true) and 0 (false). The default for this argument is 0. Set this to 1 for all LDAP communications over SSL.

Argument	Definition
<i>domain</i>	Optional. Domain name appended to portal distribution lists. For example, if the domain attribute is set to <code>mycompany.com</code> , the Finance Project portal will maintain a distribution list named <code>FinanceProject@mycompany.com</code> .
<i>defaultUser</i>	Optional. Comma-delimited list of user names to whom you want to grant moderation capabilities. These users become members of every portal distribution list that is created. The users specified must exist in the Base LDAP schema (specified in the <code>ldapBaseDN</code> argument).
<i>timeout</i>	Optional. Length of time (in seconds) that mail waits to acquire a connection before terminating. This argument defaults to <code>-1</code> . When set to <code>-1</code> , the service default (10 seconds) applies.
<i>default</i>	Optional. Indicates whether this connection is the default (or active) connection for mail. Valid values are <code>1</code> (true) and <code>0</code> (false). This argument defaults to <code>0</code> . <code>1</code> specifies that this connection is the default connection for mail. WebCenter Portal supports multiple mail connections. The mail connection configured with <code>default=1</code> is the default connection for mail services in WebCenter Portal. Additional connections, configured with <code>default=0</code> , are offered as alternatives; WebCenter Portal users can choose which one they want to use through user preferences. A connection does not cease to be the default connection for mail if you change the <code>default</code> value from <code>0</code> to <code>1</code> . To stop using a default connection, use the removeMailServiceProperty command as follows: <pre>removeMailServiceProperty('appName='webcenter', property='selected.connection')</pre>
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example sets individual attributes for a mail server connection configured for WebCenter Portal (`webcenter`):

```
wls:/weblogic/serverConfig>setMailConnection(appName='webcenter',  
name='myMailConnection', imapHost='myimaphost.com', imapPort=143,  
smtpHost='mysmtpost.com', smtpPort=25, imapSecured=0, smtpSecured=0,  
appId='extApp_Mail', timeout=60, default=1)
```

The following example sets individual attributes for a mail server connection:

```
wls:/weblogic/serverConfig>setMailConnection(appName='webcenter',  
name='myMailConnection', imapPort=993, imapSecured=1, smtpPort=465,  
smtpSecured=1)
```

The following example sets LDAP attributes for a mail server connection:

```
wls:/weblogic/serverConfig>setMailConnection(appName='webcenter',
name='myMailConnection', domain='mycompany.com', defaultUser='admin',
imapHost='myimaphost.com', imapPort=143, smtpHost='mysmtpost.com',
imapSecured=0, smtpSecured=0, smtpPort=25, appId='extApp_Mail',
default=1, ldapHost='myldaphost.com', ldapPort=389,
ldapBaseDN='CN=Users,DC=exchange,DC=uk,DC=com', ldapAdminUser='administrator',
ldapAdminPassword='adminpswd', ldapSecured=0, timeout=60)
```

2.10.3 setMailConnectionProperty

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Sets a mail server connection property. Use this command if additional parameters are required to connect to your mail server. This is an extensible way to add any connection property using a key and a value. (You are not limited to connection properties specified by [createMailConnection](#) and [setMailConnection](#) .)

Syntax

```
setMailConnectionProperty(appName, name, key, value, [secure, server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Name of an existing mail server connection.
<i>key</i>	Name of the connection property.
<i>value</i>	Value for the property. Allows any property to be modified on the connection with a key and value.
<i>secure</i>	Optional. Indicates whether the property value must be stored securely using encryption. Valid options are 1 (true) and 0 (false) . When 1, the value is encrypted. The default option is 0.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example adds a custom mail server connection property called `myProperty1` with a default value `propertyValue1`:

```
wls:/weblogic/serverConfig> setMailConnectionProperty(appName='webcenter',
name='myMailServer', key='myProperty1', value='propertyValue1')
```

2.10.4 deleteMailConnectionProperty

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Deletes a mail server connection property. Take care when deleting connection properties because the connection may not work as expected if the configuration becomes invalid as a result.

This command can only delete *additional* connection properties added using the `setMailConnectionProperty` command.

Syntax

```
deleteMailConnectionProperty(appName, name, key, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Name of an existing mail server connection.
<i>key</i>	Name of the connection property you want to delete.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example deletes a mail server connection property named `mailProperty1`:

```
wls:/weblogic/serverConfig>deleteMailConnectionProperty(appName='webcenter',
name='myMailServer', key='mailProperty1')
```

2.10.5 listMailConnections

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists all of the mail server connections that are configured for a named application.

Syntax

```
listMailConnection(appName, [verbose, name, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>verbose</i>	Optional. Displays mail server connection details in verbose mode. Valid options are 1 (true) and 0 (false). When set to 1, <code>listMailConnections</code> lists all of the mail server connections that are configured for an application, along with their details. When set to 0, only connection names are listed. This argument defaults to 0.
<i>name</i>	Optional. Name of an existing mail server connection. Use this argument to view connection details for a specific mail server connection.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example lists the names of mail server connections that are currently configured for WebCenter Portal:

```
wls:/weblogic/serverConfig>listMailConnections(appName='webcenter')
```

The following example lists connection names and details for all of the mail server connections that are currently configured for WebCenter Portal:

```
wls:/weblogic/serverConfig>listMailConnections(appName='webcenter', verbose=1)
```

The following example lists connection details for a mail server connection named `mailConnection1`:

```
wls:/weblogic/serverConfig>listMailConnections(appName='webcenter',
name='mailConnection1')
```

2.10.6 listDefaultMailConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists the default mail server connection that mail is using in a named application.

You can register multiple mail server connections but there can only be one default connection. The mail connection configured with `default=1` is the default connection for mail services in WebCenter Portal. All additional connections are offered as alternatives; WebCenter Portal users can choose which one they want to use through user preferences.

Syntax

```
listDefaultMailConnection(appName,[verbose, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>verbose</i>	Optional. Displays the default mail server connection in verbose mode, if available. Valid options are 1 (true) and 0 (false). When set to 1, the name and details of the mail server connection are listed. When set to 0, only the connection name displays. This argument defaults to 0.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example lists the name and details of the mail server connection that mail is using in WebCenter Portal:

```
wls:/weblogic/serverConfig>listDefaultMailConnection(appName='webcenter',
verbose=1)
```

2.10.7 setDefaultMailConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Specifies the *default* mail server connection for mail in a named application.

You can register multiple mail server connections but there can only be one default connection. The mail connection configured with `default=1` is the default connection for mail services in WebCenter Portal. All additional connections are offered as alternatives; WebCenter Portal users can choose which one they want to use through user preferences.

Syntax

```
setDefaultMailConnection(appName, name, [server, applicationVersion])
```

Argument	Description
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Name of an existing mail connection.

Argument	Description
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example configures mail to use a connection named `myMailServer` for the default connection in WebCenter Portal:

```
wls:/weblogic/serverConfig>setDefaultMailConnection(appName='webcenter',
name='myMailServer')
```

2.10.8 setMailServiceProperty

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Specifies default values for mail.

Configurable properties for mail are listed in [Table 2-17](#).

Table 2-17 Mail - Configurable Properties

Configuration Property	Description
<i>address.delimiter</i>	Defines the delimiter that is used to separate multiple mail addresses. A comma is used by default. Some mail servers require mail addresses in the form <code>lastname, firstname</code> and, in such cases, a semi-colon is required.
<i>mail.emailgateway.polling.frequency</i>	The frequency, in seconds, that portal distribution lists are checked for new incoming emails. The default is 1800 seconds (30 minutes).
<i>mail.messages.fetch.size</i>	Maximum number of messages displayed in mail inboxes.
<i>resolve.email.address.to.name</i>	Determines whether user email addresses are resolved to WebCenter Portal user names when LDAP is configured. Valid values are 1 (true) and 0 (false). The default value is 0. When set to 1, WebCenter Portal user names display instead of email addresses in mail task flows. Set this property to 1 if the instant messaging and presence requires user names to obtain presence status because presence information cannot be obtained when mail provides email addresses. Setting this value to 1 does impact application performance so you must take this into consideration when setting this property.

Table 2-17 (Cont.) Mail - Configurable Properties

Configuration Property	Description
<i>mail.recipient.limit</i>	Restricts the number of recipients to a message. For example, setting this value to '500' limits the number of recipients to 500.

Syntax

```
setMailServiceProperty(appName, property, value, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>property</i>	Name of the configuration property
<i>value</i>	Value for the property.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example increases the default number of messages displayed in mail inboxes to 100, in WebCenter Portal (`webcenter`):

```
wls:/weblogic/serverConfig>setMailServiceProperty(appName='webcenter',  
property='mail.messages.fetch.size', value='100')
```

2.10.9 removeMailServiceProperty

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Removes the current value that is set for a mail property. Use this command to remove any of the properties listed in [Table 2-17](#).

Take care when using this command as removing values for these properties might cause unexpected behavior.

 **Note:**

Use this command syntax to stop mail from using the current default connection:

```
removeMailServiceProperty('appName='webcenter',
property='selected.connection')
```

This command forces the default connection argument to 0. See also, [setMailConnection](#) .

Syntax

```
removeMailServiceProperty(appName, property, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>property</i>	Name of the configuration property.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example clears the current `mail.messages.fetch.size` setting for mail, in WebCenter Portal:

```
wls:/weblogic/serverConfig>removeMailServiceProperty(appName='webcenter',
property='mail.messages.fetch.size')
```

2.10.10 listMailServiceProperties

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists all configurable properties for mail.

Syntax

```
listMailServiceProperties(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .

Argument	Definition
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example lists configurable properties for mail in WebCenter Portal:

```
wls:/weblogic/serverConfig>listMailServiceProperties(appName='webcenter')
```

2.10.11 createMailExtApp

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Creates an external application suitable for mail server connections. The external application is configured with all the required additional properties: `authMethod=POST`, and specify several additional login fields:

```
fieldName='Email Address' and displaytoUser=1
```

```
fieldName='Your Name' and displaytoUser=1
```

```
fieldName='Reply-To Address' and displaytoUser=1
```

Syntax

```
createMailExtAppConnection(appName, name, [displayName, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Connection name. The name must be unique (across all connection types) within the application.
<i>displayName</i>	Optional. External application display name. A user friendly name for the external application that portal users will recognize. The display name must be unique across all external applications within the application.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example creates an external application named `MailxApp` suitable for mail server connections.

```
wls:/weblogic/serverConfig>createMailExtAppConnection(appName='webcenter',
name='MailxApp', displayName='Mail Ext App')
```

2.11 Notifications

Use the commands listed in [Table 2-18](#) to manage settings for the notifications in a named application.

Configuration changes made using these WLST commands take effect after your restart the Managed Server where the application is deployed. For details, see *Oracle Fusion Middleware Administering Oracle WebCenter Portal*.

Table 2-18 Notifications WLST Commands

Use this command...	To...	Use with WLST...
setNotificationsConfig	Specify the connection used for routing notifications raised in a named application.	Online
getNotificationsConfig	Return details about the connection that is used to send notifications raised in a named application.	Online

2.11.1 setNotificationsConfig

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Specifies the connection used for routing notifications raised in a named application.

Use an existing mail server or BPEL server connection. If the application is connected to a BPEL server, the Oracle User Messaging Service (UMS) is available for routing notifications through multiple messaging channels, including mail and SMS. If you configure notifications to use a BPEL server connection, you may specify a sender 'From' address for each available messaging channel. That is, you can specify a sender mail address and an SMS address.

Alternatively, you can route notification messages through a mail server. If you configure notifications to use a mail server connection, the external application associated with the mail server connection must contain shared credentials. Shared credentials are required for routing application-wide notifications.

Syntax

```
setNotificationsConfig(appName, type, name, [senderMailAddress, senderSMSAddress,
server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>type</i>	Type of connection used to send notifications. Valid values are <code>MAIL</code> and <code>BPEL</code> .
<i>name</i>	Name of an <i>existing</i> connection. Consider the following: <ul style="list-style-type: none"> • Mail server connection—The external application associated with the mail server connection must contain shared credentials. • BPEL server connection—Oracle User Messaging Service (UMS) must be available on the BPEL server.
<i>senderMailAddress</i>	Optional. Mail address from which all mail notifications are sent. Use the format: <code><email_alias><<email_address>> or <email address></code> . For example, <code>WebCenter Notification<notifications@webcenter.com> or notifications@webcenter.com</code> . This argument applies to notifications routed through BPEL servers. When a BPEL server is used and UMS is configured with multiple email drivers, this address is also used to identify the appropriate email driver. When a mail server is used, the "From Address" is the same user that is specified for the associated external application's shared credentials.
<i>senderSMSAddress</i>	Optional. SMS number from which all SMS notifications are sent. Typically, the SMS address format is a 4-6 digit number (without -, spaces, or any other characters). For example, <code>28734</code> . This argument applies to notifications routed through BPEL servers. When a BPEL server is used and UMS is configured with multiple SMS drivers, this address is also used to identify the appropriate SMS driver.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example specifies that notifications in WebCenter Portal (`webcenter`) use a BPEL server connection named `WebCenter-SMS` and also defines the mail address and SMS address from which all notifications are sent:

```
wls:/weblogic/serverConfig>setNotificationsConfig(appName='webcenter',
type='BPEL',
name='WebCenter-SMS', senderMailAddress='WebCenter
Notification<notifications@webcenter.com>',
senderSMSAddress='28734')
```

2.11.2 getNotificationsConfig

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Returns details about the connection that is used to send notifications raised in a named application.

Syntax

```
getNotificationsConfig(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example returns details about the connection used by notifications service in WebCenter Portal:

```
wls:/weblogic/serverConfig>getNotificationsConfig(appName='webcenter')
```

```
ConnectionType:    BPEL
ConnectionName:   WebCenter-SMS
SenderMailAddress: notifications@webcenter.com
SenderSMSAddress: 28776
```

2.12 People Connections

Use the commands listed in [Table 2-19](#) to manage user profile information in WebCenter Portal.

Configuration changes made using these WLST commands are effective only after your restart the Managed Server on which WebCenter Portal is deployed. For details, see *Oracle Fusion Middleware Administering Oracle WebCenter Portal*.

Table 2-19 People Connection WLST Commands

Use this command...	To...	Use with WLST...
startSyncProfiles	Synchronize profile information in the LDAP store, with the WebCenter Portal database schema.	Online
syncProfile	Synchronize profile information for a specific user.	Online

Table 2-19 (Cont.) People Connection WLST Commands

Use this command...	To...	Use with WLST...
<code>stopSyncProfiles</code>	Stop the profile synchronization process.	Online
<code>isSyncProfilesRunning</code>	Check whether profile synchronization is in progress.	Online
<code>setProfileConfig</code>	Set profile properties.	Online
<code>getProfileConfig</code>	Return current profile properties.	Online
<code>listProfileConfig</code>	List current profile properties.	Online
<code>setProfilePhotoSync</code>	Specify whether to synchronize user profile photos in the WebCenter schema from the LDAP server.	Online

2.12.1 startSyncProfiles

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Synchronizes profile information in the LDAP store with the WebCenter Portal database schema.

Syntax

```
startSyncProfiles(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example synchronizes user profiles for WebCenter Portal:

```
wls:/weblogic/serverConfig>startSyncProfiles(appName='webcenter')
```

2.12.2 stopSyncProfiles

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Stops the profile synchronization process, if currently in progress.

Syntax

```
stopSyncProfiles(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example stops the profile synchronization process for WebCenter Portal:

```
wls:/weblogic/serverConfig>stopSyncProfiles(appName='webcenter')
```

2.12.3 isSyncProfilesRunning

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Checks whether profile synchronization is in progress.

Syntax

```
isSyncProfilesRunning(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example checks whether profile synchronization is in progress for WebCenter Portal:

```
wls:/weblogic/serverConfig>isSyncProfilesRunning(appName='webcenter')
```

2.12.4 syncProfile

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Synchronizes profile information for a specific user in the LDAP store with the WebCenter Portal schema.

Syntax

```
syncProfile(appName, userName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>userName</i>	Name of the user whose profile information you want to synchronize.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example synchronizes profile information for a user named chad:

```
wls:/weblogic/serverConfig>syncProfile(appName='webcenter', userName='chad')
```

2.12.5 setProfileConfig

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Sets profile properties for a named application. If you omit a parameter, the corresponding configuration remains unchanged.

Syntax

```
setProfileConfig(appName, ProfilePageVersion, ProfileSyncHourOfDay,  
ProfileSyncFrequencyInDays, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.

Argument	Definition
<i>ProfilePageVersion</i>	Optional. Specifies the profile page version to use. Valid values for ProfilePageVersion are: v1 - Use old-style profile pages v2 - Use the new profile page format The default value is v2.
<i>ProfileSyncHourOfDay</i>	Optional. Specifies when (the hour) to start profile synchronization. Any value between 0 and 23. The default value is 23, equivalent to 11pm.
<i>ProfileSyncFrequencyInDays</i>	Optional. Specifies how often profile synchronization takes place (in days). Any value greater than 0. The default value is 7.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example sets new values for the ProfilePageVersion and ProfileSyncHourOfDay configuration settings.

All other settings remain unchanged:

```
wls:/weblogic/serverConfig>setProfileConfig(appName='webcenter',ProfilePageVersion=v2,ProfileSyncHourOfDay=12)
```

2.12.6 getProfileConfig

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Returns current profile settings for a named application.

Syntax

```
getProfileConfig(appName, key, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.

Argument	Definition
<i>key</i>	Name of a profile property. Valid values include: - ProfilePageVersion - ProfileSyncHourOfDay - ProfileSyncFrequencyInDays
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following examples return current settings for various profile properties:

```
wls:/weblogic/serverConfig> getProfileConfig(appName='webcenter',
key='ProfilePageVersion')
wls:/weblogic/serverConfig> getProfileConfig(appName='webcenter',
key='ProfileSyncHourOfDay')
wls:/weblogic/serverConfig> getProfileConfig(appName='webcenter',
key='ProfileSyncFrequencyInDays')
```

2.12.7 listProfileConfig

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists the current profile settings for a named application.

Syntax

```
listProfileConfig(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example lists current profile settings for WebCenter Portal (webcenter):

```
wls:/weblogic/serverConfig>listProfileConfig(appName='webcenter')
```

2.12.8 setProfilePhotoSync

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Specifies whether to synchronize the latest user profile photos in LDAP with WebCenter Portal's profile cache. When synchronization is enabled, profile photos are synchronized when the profile cache expires. When synchronization is disabled, profile photos are not synchronized with the WebCenter Portal's profile cache.

Syntax

```
setProfilePhotoSync(appName, enablePhotoSync, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>enablePhotoSync</i>	Specifies whether to periodically synchronize user profile photos in LDAP. Valid values are 1 and 0. <ul style="list-style-type: none"> 1 - Synchronize user profile photos periodically (after WebCenter Portal's profile cache expires). 0 - Profile photos in LDAP are not automatically synchronized with WebCenter Portal's profile cache. WebCenter Portal users must manually upload new photos through their profile page. The default value is 1.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example enables profile photo synchronization in WebCenter Portal:

```
wls:/weblogic/serverConfig>setProfilePhotoSync(appName='webcenter', enablePhotoSync=1)
```

2.13 Personal Events

Use the commands listed in [Table 2-20](#) to manage personal events server connections for a named application.

Configuration changes made using these WLST commands are only effective after you restart the Managed Server on which the application is deployed. For details, see *Oracle Fusion Middleware Administering Oracle WebCenter Portal*.

Table 2-20 Personal Events WLST Commands

Use this command...	To...	Use with WLST...
createPersonalEventConnection	Create a personal events server connection for a named application.	Online
setPersonalEventConnection	Edit an existing personal events server connection.	Online
listPersonalEventConnections	List all of the personal events server connections that are configured for a named application	Online

2.13.1 createPersonalEventConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Creates a personal events server connection for a named application.

Personal events supports connections to Microsoft Exchange Server 2007, Microsoft Exchange Server 2010, and Microsoft Exchange Server 2013. While you can register multiple personal events connections for WebCenter Portal, only one connection is used for personal events services - the default (or active) connection.

Syntax

```
createPersonalEventConnection(appName, name, webServiceUrl, adapterName, appId,
[default, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>name</i>	Connection name. The name must be unique (across all connection types within the application).
<i>webServiceUrl</i>	URL of the web service exposing the event application. Use the format: <i>protocol://host:port/appWebServiceInterface/WSName</i>
<i>adapterName</i>	Specify the adapter that matches the personal events server. Valid values are MSEx2007, MSEx2010, and MSEx2013. Choose MSEx2007 for Microsoft Exchange Server 2007, MSEx2010 for Microsoft Exchange Server 2010, and MSEx2013 for Microsoft Exchange Server 2013. Each adapter has its own mechanism of authenticating and exchanging data
<i>appId</i>	External application associated with the Microsoft Exchange Server providing personal events services. If specified, external application credential information is used to authenticate users against the Microsoft Exchange Server.

Argument	Definition
<i>default</i>	Optional. Indicates whether this connection is the default connection for personal event services. Valid values are 1 (true) and 0 (false). The default for this argument is 0. To specify that personal events uses this connection, set the value to 1. While you can register multiple connections for WebCenter Portal, only one connection is used for personal event services—the default (or active) connection.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example creates a connection named `MyPEConnection` for WebCenter Portal (`webcenter`). The connection points to a Microsoft Exchange Server 2007 and is designated as the default connection for personal event services:

```
wls:/weblogic/serverConfig>createPersonalEventConnection(appName='webcenter',
name='MyPEConnection', webServiceUrl='http://myexchange.com/EWS/Services.wsdl',
adapterName='MSEx2007', appId='ExtPEApp', default=1)
```

The following example creates a connection named `MyPEConnection` for a WebCenter Portal. The connection points to Microsoft Exchange Server 2007:

```
wls:/weblogic/serverConfig>createPersonalEventConnection(appName='webcenter',
name='MyPEConnection', webServiceUrl='http://myexchange.com/ExchangeWS/
PersonalEventsWebService.asmx', adapterName='MSEx2007', appId='ExtPEApp')
```

2.13.2 setPersonalEventConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Edits an existing personal events server connection configured for a named application.

Syntax

```
setPersonalEventConnection(appName, name, [webServiceUrl, adapterName, appId, default,
server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Name of an existing personal events server connection.

Argument	Definition
<i>webServiceUrl</i>	URL of the web service exposing the event application. Use the format: <i>protocol://host:port/appWebServiceInterface/WSName</i>
<i>adapterName</i>	Optional. Specify the adapter that matches the personal events server. Valid values are MSeX2007, MSeX2010, and MSeX2013. Choose MSeX2007 for Microsoft Exchange Server 2007, MSeX2010 for Microsoft Exchange Server 2010, and MSeX2013 for Microsoft Exchange Server 2013. Each adapter has its own mechanism of authenticating and exchanging data
<i>appId</i>	Optional. External application associated with the Microsoft Exchange Server providing personal events services. If specified, external application credential information is used to authenticate users against the Microsoft Exchange Server.
<i>default</i>	Optional. Indicates whether this connection is the default connection for personal events. Valid values are 1 (true) and 0 (false) . The default for this argument is 0. To specify that personal events uses this connection, set the value to 1. While you can register multiple connections for WebCenter Portal, only one connection is used for personal event services— the default (or active) connection.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example updates the web service URL for a connection named MyPEConnection.

```
wls:/weblogic/serverConfig>setPersonalEventConnection( appName='webcenter',
name='MyPEConnection', webServiceUrl='http://myexchange.com/EWS/Services.wsdl')
```

The following example makes a connection named MyPEConnection the default connection for personal events services in WebCenter Portal.

```
wls:/weblogic/serverConfig>setPersonalEventConnection( appName='webcenter',
name='MyPEConnection', default=1)
```

2.13.3 listPersonalEventConnections

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists all of the personal events server connections that are configured for a named application.

Syntax

```
listPersonalEventConnections(appName, [verbose, name, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>verbose</i>	Optional. Displays connection details for personal events in verbose mode. Valid options are 1 (true) and 0 (false). When set to 1, <code>listPersonalEventConnections</code> lists all of the personal events server connections that are configured for a named application, along with their details. When set to 0, only connection names are listed. This argument defaults to 0. When set to 0, do not specify the <code>name</code> argument.
<i>name</i>	Optional. Name of an existing personal events connection. Use this argument to view connection details for a specific personal events server.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example lists connection names and details for all of the personal events server connections currently configured for WebCenter Portal (`webcenter`):

```
wls:/weblogic/serverConfig>listPersonalEventConnections(appName='webcenter', verbose=1)
```

The following example displays connection details for a personal events server connection named `MyPEConnection`:

```
wls:/weblogic/serverConfig>listPersonalEventConnections(appName='webcenter', verbose=1, name='MyPEConnection')
```

2.14 Portlet Producers

Use the commands listed in [Table 2-21](#) to manage portlet producers used in WebCenter Portal.

All configuration changes made using these WLST commands are immediately available in the application.

Table 2-21 Portlet Producer WLST Commands

Use this command...	To...	Use with WLST...
registerWSRPProducer	Create and register a WSRP producer.	Online
setWSRPProducer	Edit WSRP producer registration details.	Online
listWSRPProducers	List WSRP producer registration details.	Online

Table 2-21 (Cont.) Portlet Producer WLST Commands

Use this command...	To...	Use with WLST...
deregisterWSRPProducer	Deregister a WSRP producer, and delete the associated WSRP and web service connections.	Online
listWSRPProducerRegistrationProperties	List registration properties supported by a WSRP producer.	Online
listWSRPProducerUserCategories	List any user categories that the WSRP producer might support.	Online
mapWSRPProducerUserCategory	Map a role that is defined in the specified application to a user category supported by a WSRP producer.	Online
registerPDKJavaProducer	Create and register an Oracle PDK-Java producer.	Online
setPDKJavaProducer	Edit PDK-Java producer registration details.	Online
listPDKJavaProducers	List registered Oracle PDK-Java producers.	Online
deregisterPDKJavaProducer	Deregister an Oracle PDK-Java producer, deleting the associated connection.	Online
refreshProducer	Refresh the metadata stored for the named producer to reflect the portlets currently offered by that producer.	Online
listPortletClientConfig	List portlet client configuration for a named application.	Online
setPortletClientConfig	Edit portlet client configuration settings.	Online
getPortletClientConfig	Return portlet client configuration settings.	Online
registerOOTBProducers	Register out-of-the-box producers provided with Oracle WebCenter Portal.	Online
deregisterOOTBProducers	Deregister out-of-the-box producers provided with Oracle WebCenter Portal.	Online
registerSampleProducers	Register the sample producers provided with Oracle WebCenter Portal.	Online
deregisterSampleProducers	Deregister sample producers.	Online

2.14.1 registerWSRPProducer

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Creates a connection to a WSRP portlet producer and registers the WSRP producer with a named application. When you create a WSRP producer connection, a web service connection is also created named `<name>-wsconn` where `<name>` is the value specified for the name argument.

Syntax

```
registerWSRPProducer(appName, name, url, [proxyHost], [proxyPort],
[timeout], [externalApp], [registrationProperties], [tokenType], [issuer], [defUser],
[keyStorePath], [keyStorePswd], [sigKeyAlias], [sigKeyPswd], [encKeyAlias],
[encKeyPswd], [recptAlias], [enforcePolicyURI], [server], [applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Connection name. The name must be unique (across all connection types) within the application. The name you specify here will appear in Composer (under the <i>Portlets</i> folder).
<i>url</i>	Producer WSDL URL. The syntax will vary according to your WSRP implementation, for example: <code>http://host_name:port_number/context_root/portlets/wsrp2?WSDL</code> <code>http://host_name:port_number/context_root/portlets/wsrp1?WSDL</code> <code>http://host_name:port_number/context_root/portlets/?WSDL</code> (WSRP 1.0 for backward compatibility) Where: <ul style="list-style-type: none">• <code>host_name</code> is the server where your producer is deployed• <code>port_number</code> is the HTTP listener port number• <code>context_root</code> is the web application's context root• <code>portlets[/wsrp(1 2)]?WSDL</code> is static text. The text entered here depends on how the producer is deployed. For example: <code>http://myhost.com:7778/MyPortletApp/portlets/wsrp2?WSDL</code>
<i>proxyHost</i>	Optional. Host name or IP address of the proxy server. A proxy is required when the application and the remote portlet producer are separated by a firewall and an HTTP proxy is needed to communicate with the producer.
<i>proxyPort</i>	Optional. Port number on which the proxy server listens.
<i>timeout</i>	Optional. Timeout setting for communications with the producer, in seconds. For example, the maximum time the producer may take to register, deregister, or display portlets on portal pages. This argument defaults to 30. Individual portlets may define their own timeout period, which takes precedence over the value expressed here.
<i>registrationProperties</i>	Optional. A list of registration properties and their values. The format of this argument must be a comma-separated list of valid registration properties, each followed by an equals symbol and the value. For example: <code>name=Producer,key=123</code> . The registration properties for a producer can be found using <code>listWSRPProducerRegistrationProperties</code> . See listWSRPProducerRegistrationProperties .

Argument	Definition
<i>tokenType</i>	<p data-bbox="656 273 1406 327">Optional. Type of token profile to use for authentication with this WSRP producer.</p> <p data-bbox="656 338 1305 365">When the argument <code>enforcePolicyURI=1</code>, valid values are:</p> <ul data-bbox="656 375 1458 1921" style="list-style-type: none"> <li data-bbox="656 375 1458 871"> <p data-bbox="703 375 1458 575">USERNAME_WITHOUT_PASSWORD (oracle/wss10_username_id_propagation_with_msg_protection_client_policy) —This policy provides message protection (integrity and confidentiality) and identity propagation for outbound SOAP requests in accordance with the WS-Security 1.0 standard. Credentials (<i>user name</i> only) are included in outbound SOAP request messages through a WS-Security UsernameToken header. No password is included.</p> <p data-bbox="703 585 1458 695">Message protection is provided using WS-Security 1.0's Basic128 suite of asymmetric key technologies. Specifically, RSA key mechanisms for message confidentiality, SHA-1 hashing algorithm for message integrity, and AES-128 bit encryption.</p> <li data-bbox="656 705 1458 1291"> <p data-bbox="703 705 1458 871">USERNAME_WITH_PASSWORD (oracle/wss10_username_token_with_message_protection_client_policy)— This policy provides message protection (integrity and confidentiality) and authentication for outbound SOAP requests in accordance with the WS-Security v1.0 standard. Both plain text and digest mechanisms are supported.</p> <p data-bbox="703 882 1458 991">This policy uses WS-Security's Basic 128 suite of asymmetric key technologies. Specifically, RSA key mechanism for message confidentiality, SHA-1 hashing algorithm for message integrity, and AES-128 bit encryption.</p> <p data-bbox="703 1001 1458 1087">Use this token profile if the WSRP producer has a different identity store. You will need to define an external application pertaining to the producer and associate the external application with this producer.</p> <li data-bbox="656 1098 1458 1417"> <p data-bbox="703 1098 1458 1291">SAML_TOKEN_WITH_MSG_INTEGRITY (wss10_saml_token_with_message_integrity_client_policy)—This policy provides message-level integrity protection and SAML-based authentication for outbound SOAP requests in accordance with the WS-Security 1.0 standard. A SAML token, included in the SOAP message, is used in SAML-based authentication with sender vouches confirmation.</p> <p data-bbox="703 1302 1458 1356">This policy uses WS-Security's Basic 128 suite of asymmetric key technologies and SHA-1 hashing algorithm for message integrity.</p> <p data-bbox="703 1367 1458 1417">When this policy is selected, the recipient key alias (<code>recptAlias</code>) must be disabled.</p> <li data-bbox="656 1428 1458 1774"> <p data-bbox="703 1428 1458 1627">SAML_TOKEN_WITH_MSG_PROTECTION (oracle/wss10_saml_token_with_message_protection_client_policy)—This policy provides message-level protection (integrity and confidentiality) and SAML-based authentication for outbound SOAP requests in accordance with the WS-Security 1.0 standard. The web service consumer includes a SAML token in the SOAP header and the confirmation type is sender-vouches.</p> <p data-bbox="703 1638 1458 1774">This policy uses WS-Security's Basic 128 suite of asymmetric key technologies. Specifically, RSA key mechanisms for message confidentiality, SHA-1 hashing algorithm for message integrity, and AES-128 bit encryption. and SHA-1 hashing algorithm for message integrity.</p> <li data-bbox="656 1785 1458 1921"> <p data-bbox="703 1785 1458 1921">WSS11_SAML_TOKEN_WITH_MSG_PROTECTION (oracle/wss11_saml_token_with_message_protection_client_policy)—This policy provides message-level protection (integrity and confidentiality) and SAML token population for outbound SOAP requests in accordance with the WS-Security 1.1 standard. A SAML token,</p>

Argument	Definition
<i>issuer</i>	<p>included in the SOAP message, is used in SAML-based authentication with sender vouches confirmation. This policy uses the symmetric key technology for signing and encryption, and WS-Security's Basic 128 suite of asymmetric key technologies for endorsing signatures.</p> <ul style="list-style-type: none"> • WSS10_SAML_TOKEN_ONLY (oracle/wss10_saml_token_client_policy)—This policy provides SAML-based authentication for outbound SOAP request messages in accordance with the WS-Security 1.0 standard. The policy propagates user identity and is typically used in intra departmental deployments where message protection and integrity checks are not required. <p>This policy does not require any keystore configuration.</p> <p>If the argument <code>enforcePolicyURI=0</code>, you can specify any <i>valid</i> Oracle Web Services Manager (OWSM) policy URI for the <code>tokenType</code> argument.</p>
<i>issuer</i>	<p>Optional. Name of the issuer of the token. The issuer name is the entity that vouches for the verification of the subject. For example: <code>www.oracle.com</code>.</p> <p>This argument only applies when the <code>tokenType</code> is: <code>SAML_TOKEN_WITH_MSG_PROTECTION</code>, <code>SAML_TOKEN_WITH_MSG_INTEGRITY</code>, <code>WSS10_SAML_TOKEN_ONLY</code>, <code>WSS11_SAML_TOKEN_WITH_MSG_PROTECTION</code>.</p>
<i>defUser</i>	<p>Optional. User name to assert to the remote producer when the user is not authenticated with the portal application.</p> <p>When unauthenticated, the identity <i>anonymous</i> is associated with the application user. The value <i>anonymous</i> may be inappropriate for the remote producer, so you may need to specify an alternative identity here. Keep in mind though, that in this case, the application has not authenticated the user so the default user you specify should be a low privileged user in the remote producer. If the user has authenticated to the application, the user's identity is asserted rather than the default user.</p> <p>This argument only applies when the <code>tokenType</code> is: <code>USERNAME_WITHOUT_PASSWORD</code>, <code>SAML_TOKEN_WITH_MSG_PROTECTION</code>, <code>SAML_TOKEN_WITH_MSG_INTEGRITY</code>, <code>WSS10_SAML_TOKEN_ONLY</code>, <code>WSS11_SAML_TOKEN_WITH_MSG_PROTECTION</code>.</p>
<i>extApp</i>	<p>Optional. This argument applies when the <code>tokenType</code> is <code>USERNAME_WITH_PASSWORD</code>. If this producer uses an external application to store and supply user credentials for authentication, use this argument to name the associated external application.</p>
<i>keyStorePath</i>	<p>Optional. Full path to the key store that contains the certificate and the private key that is used for signing some parts of the SOAP message, such as the security token and SOAP message body. The selected file should be a key store created with the Java keytool.</p>
<i>keyStorePswd</i>	<p>Optional. Password to the key store that was set when the key store was created.</p>
<i>sigKeyAlias</i>	<p>Optional. Identifier for the certificate associated with the private key that is used for signing.</p>
<i>sigKeyPswd</i>	<p>Optional. Password for accessing the key identified by the alias that is specified using the <code>sigKeyAlias</code> argument.</p>
<i>encKeyAlias</i>	<p>Optional. Key alias to be used for encryption. A valid value is one of the key aliases that is located in the specified key store.</p>
<i>encKeyPswd</i>	<p>Optional. Password for accessing the encryption key.</p>

Argument	Definition
<i>recptAlias</i>	Optional. Key store alias that is associated with the producer's certificate. This certificate is used to encrypt the message to the producer. Do not specify a recipient key alias when the <code>tokenType</code> is <code>SAML_TOKEN_WITH_MSG_INTEGRITY</code> .
<i>enforcePolicyURI</i>	Optional. Valid values are 1 (true) and 0 (false). When set to 1, users must specify one of the following token profiles for the <code>tokenType</code> argument: <code>USERNAME_WITHOUT_PASSWORD</code> , <code>USERNAME_WITH_PASSWORD</code> , <code>SAML_TOKEN_WITH_MSG_PROTECTION</code> , <code>SAML_TOKEN_WITH_MSG_INTEGRITY</code> , <code>WSS11_SAML_TOKEN_WITH_MSG_PROTECTION</code> , <code>WSS10_SAML_TOKEN_ONLY</code> When set to 0, users can specify any Oracle Web Services Manager (OWSM) policy URI. The user must ensure that the OWSM policy specified is valid. The default value is 1.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example registers a WSRP producer named `WSRPSamples` with WebCenter Portal (`webcenter`):

```
wls:/weblogic/serverConfig> registerWSRPProducer(appName='webcenter',
name='WSRPSamples', url='http://myhost.com:9999/
portletapp/portlets/wsrp2?WSDL')
```

The following example registers a secure WSRP producer:

```
wls:/weblogic/serverConfig> registerWSRPProducer(appName='webcenter',
name='WSRPSamples2', url='http://myhost.com:8899/portletapp/portlets/wsrp2?
WSDL',
tokenType='WSS11_SAML_TOKEN_WITH_MSG_PROTECTION', issuer='www.oracle.com',
defUser='anonymous', keyStorePath='/keys/mykeystore.jks', keyStorePswd='Test1',
sigKeyAlias='mysigalias', sigKeyPswd='mysigpswd', encKeyAlias='myencalias',
encKeyPswd='myencpswd', recptAlias='myrcptalias')
```

2.14.2 setWSRPProducer

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Edits registration details for an existing WSRP producer.

Syntax

```
setWSRPProducer(appName, name, [url], [proxyHost], [proxyPort], [timeout],
[externalApp], [tokenType],[issuer], [defUser], [keyStorePath], [keyStorePswd])
```

[sigKeyAlias], [sigKeyPswd], [encKeyAlias], [encKeyPswd], [recptAlias],
[enforcePolicyURI], [server], [applicationVersion])

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>name</i>	Name of an existing WSRP producer.
<i>url</i>	Optional. WSRP producer URL. The syntax will vary according to your WSRP implementation, for example: <pre>http://host_name:port_number/context_root/portlets/wsrp2?WSDL</pre> <pre>http://host_name:port_number/context_root/portlets/wsrp1?WSDL</pre> <pre>http://host_name:port_number/context_root/portlets/?WSDL</pre> (WSRP 1.0 for backward compatibility) Where: <ul style="list-style-type: none"> • <code>host_name</code> is the server where your producer is deployed • <code>port_number</code> is the HTTP listener port number • <code>context_root</code> is the Web application's context root • <code>portlets[/wsrp(1 2)]?WSDL</code> is static text. The text entered here depends on how the producer is deployed. For example: <pre>http://myhost:7778/MyPortletApp/portlets/wsrp2?WSDL</pre>
<i>proxyHost</i>	Optional. Host name or IP address of the proxy server. A proxy is required when the application and the remote portlet producer are separated by a firewall and an HTTP proxy is needed to communicate with the producer.
<i>proxyPort</i>	Optional. Port number on which the proxy server listens.
<i>timeout</i>	Optional. Timeout setting for communications with the producer, in seconds. For example, the maximum time the producer may take to register, deregister, or display portlets on portal pages. This argument defaults to 30. Individual portlets may define their own timeout period, which takes precedence over the value expressed here.
<i>extApp</i>	Optional. This argument applies when the <code>tokenType</code> is <code>USERNAME_WITH_PASSWORD</code> . If this producer uses an external application to store and supply user credentials for authentication, use this argument to name the associated external application.

Argument	Definition
<i>tokenType</i>	<p data-bbox="656 273 1406 327">Optional. Type of token profile to use for authentication with this WSRP producer.</p> <p data-bbox="656 338 1305 365">When the argument <code>enforcePolicyURI=1</code>, valid values are:</p> <ul data-bbox="656 375 1458 1858" style="list-style-type: none"> <li data-bbox="656 375 1458 730"> <p data-bbox="703 375 1117 403">USERNAME_WITHOUT_PASSWORD</p> <p data-bbox="703 413 1458 611">(oracle/wss10_username_id_propagation_with_msg_protection_client_policy)—This policy provides message protection (integrity and confidentiality) and identity propagation for outbound SOAP requests in accordance with the WS-Security 1.0 standard. Credentials (<i>user name</i> only) are included in outbound SOAP request messages through a WS-Security UsernameToken header. No password is included.</p> <p data-bbox="703 621 1442 730">Message protection is provided using WS-Security 1.0's Basic 128 suite of asymmetric key technologies. Specifically, RSA key mechanisms for message confidentiality, SHA-1 hashing algorithm for message integrity, and AES-128 bit encryption.</p> <li data-bbox="656 741 1458 1066"> <p data-bbox="703 741 1068 768">USERNAME_WITH_PASSWORD</p> <p data-bbox="703 779 1458 940">(oracle/wss10_username_token_with_message_protection_client_policy)—This policy provides message protection (integrity and confidentiality) and authentication for outbound SOAP requests in accordance with the WS-Security v1.0 standard. Both plain text and digest mechanisms are supported.</p> <p data-bbox="703 951 1406 1066">This policy uses WS-Security's Basic 128 suite of asymmetric key technologies. Specifically, RSA key mechanism for message confidentiality, SHA-1 hashing algorithm for message integrity, and AES-128 bit encryption.</p> <p data-bbox="703 1077 1435 1157">Use this token profile if the WSRP producer has a different identity store. You will need to define an external application pertaining to the producer and associate the external application with this producer.</p> <li data-bbox="656 1167 1458 1493"> <p data-bbox="703 1167 1149 1194">SAML_TOKEN_WITH_MSG_INTEGRITY</p> <p data-bbox="703 1205 1458 1367">(wss10_saml_token_with_message_integrity_client_policy)—This policy provides message-level integrity and SAML-based authentication for outbound SOAP requests in accordance with the WS-Security 1.0 standard. A SAML token, included in the SOAP message, is used in SAML-based authentication with sender vouches confirmation.</p> <p data-bbox="703 1377 1398 1432">This policy uses WS-Security's Basic 128 suite of asymmetric key technologies and SHA-1 hashing algorithm for message integrity.</p> <p data-bbox="703 1442 1430 1493">When this policy is selected, the recipient key alias (<code>recptAlias</code>) must be disabled.</p> <li data-bbox="656 1503 1458 1858"> <p data-bbox="703 1503 1182 1530">SAML_TOKEN_WITH_MSG_PROTECTION</p> <p data-bbox="703 1541 1458 1703">(oracle/wss10_saml_token_with_message_protection_client_policy)—This policy provides message-level protection (integrity and confidentiality) and SAML-based authentication for outbound SOAP requests in accordance with the WS-Security 1.0 standard. The Web service consumer includes a SAML token in the SOAP header and the confirmation type is sender-vouches.</p> <p data-bbox="703 1713 1414 1858">This policy uses WS-Security's Basic 128 suite of asymmetric key technologies. Specifically, RSA key mechanisms for message confidentiality, SHA-1 hashing algorithm for message integrity, and AES-128 bit encryption. and SHA-1 hashing algorithm for message integrity.</p>

Argument	Definition
<i>tokenType</i> continued...	<ul style="list-style-type: none"> <li data-bbox="656 275 1446 562">• WSS11_SAML_TOKEN_WITH_MSG_PROTECTION (oracle/wss11_saml_token_with_message_protection_client_policy)—This policy enables message-level protection (integrity and confidentiality) and SAML token population for outbound SOAP requests in accordance with the WS-Security 1.1 standard. A SAML token, included in the SOAP message, is used in SAML-based authentication with sender vouches confirmation. This policy uses the symmetric key technology for signing and encryption, and WS-Security's Basic 128 suite of asymmetric key technologies for endorsing signatures. <li data-bbox="656 569 1446 747">• WSS10_SAML_TOKEN_ONLY (oracle/wss10_saml_token_client_policy)—This policy includes SAML-tokens in outbound SOAP request messages in accordance with the WS-Security 1.0 standard. The policy propagates user identity and is typically used in intra departmental deployments where message protection and integrity checks are not required. This policy does not require any keystore configuration. If the argument <code>enforcePolicyURI=0</code>, you can specify any <i>valid</i> Oracle Web Services Manager (OWSM) policy URI for the <code>tokenType</code> argument.
<i>issuer</i>	<p data-bbox="656 863 1446 915">Optional. Name of the issuer of the token. The issuer name is the entity that vouches for the verification of the subject. For example: <code>www.oracle.com</code>.</p> <p data-bbox="656 926 1446 1041">This argument only applies when the <code>tokenType</code> is: <code>SAML_TOKEN_WITH_MSG_PROTECTION</code>, <code>SAML_TOKEN_WITH_MSG_INTEGRITY</code>, <code>WSS10_SAML_TOKEN_ONLY</code>, <code>WSS11_SAML_TOKEN_WITH_MSG_PROTECTION</code>.</p>
<i>defUser</i>	<p data-bbox="656 1062 1446 1115">Optional. User name to assert to the remote producer when the user is not authenticated with the application.</p> <p data-bbox="656 1125 1446 1325">When unauthenticated, the identity <i>anonymous</i> is associated with the application user. The value <i>anonymous</i> may be inappropriate for the remote producer, so you may need to specify an alternative identity here. Keep in mind though, that in this case, the application has not authenticated the user so the default user you specify should be a low privileged user in the remote producer. If the user has authenticated to the application, the user's identity is asserted rather than the default user.</p> <p data-bbox="656 1335 1446 1451">This argument only applies when the <code>tokenType</code> is: <code>USERNAME_WITHOUT_PASSWORD</code>, <code>SAML_TOKEN_WITH_MSG_PROTECTION</code>, <code>SAML_TOKEN_WITH_MSG_INTEGRITY</code>, <code>WSS10_SAML_TOKEN_ONLY</code>, <code>WSS11_SAML_TOKEN_WITH_MSG_PROTECTION</code>.</p>
<i>keyStorePath</i>	<p data-bbox="656 1472 1446 1577">Optional. Full path to the key store that contains the certificate and the private key that is used for signing some parts of the SOAP message, such as the security token and SOAP message body. The selected file should be a key store created with the Java keytool.</p>
<i>keyStorePswd</i>	<p data-bbox="656 1598 1446 1650">Optional. Password to the key store that was set when the key store was created.</p>
<i>sigKeyAlias</i>	<p data-bbox="656 1671 1446 1724">Optional. Identifier for the certificate associated with the private key that is used for signing.</p>
<i>sigKeyPswd</i>	<p data-bbox="656 1745 1446 1797">Optional. Password for accessing the key identified by the alias that is specified using the <code>sigKeyAlias</code> argument.</p>

Argument	Definition
<i>encKeyAlias</i>	Optional. Key alias used by the producer to encrypt the return message. A valid value is one of the key aliases that is located in the specified key store. If not specified, the producer uses the signing key for encrypting the return message.
<i>encKeyPswd</i>	Optional. Password for accessing the encryption key.
<i>recptAlias</i>	Optional. Key store alias that is associated with the producer's certificate. This certificate is used to encrypt the outbound message to the producer. Do not specify a recipient key alias when the <code>tokenType</code> is <code>SAML_TOKEN_WITH_MSG_INTEGRITY</code> .
<i>enforcePolicyURI</i>	Optional. Valid values are 1 (true) and 0 (false). When set to 1, users must specify one of the following token profiles for the <code>tokenType</code> argument: <code>USERNAME_WITHOUT_PASSWORD</code> , <code>USERNAME_WITH_PASSWORD</code> , <code>SAML_TOKEN_WITH_MSG_PROTECTION</code> , <code>SAML_TOKEN_WITH_MSG_INTEGRITY</code> , <code>WSS11_SAML_TOKEN_WITH_MSG_PROTECTION</code> , <code>WSS10_SAML_TOKEN_ONLY</code> When set to 0, users can specify any Oracle Web Services Manager (OWSM) policy URI. The user must ensure that the OWSM policy specified is valid. The default value is 1.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

This example increases the timeout, for the `WSRPSamples` producer, to 60 seconds:

```
wls:/weblogic/serverConfig>setWSRPProducer (appName='webcenter',
name='WSRPSamples', timeout=60)
```

This example updates security properties on a secure WSRP producer:

```
wls:/weblogic/serverConfig>setWSRPProducer (appName='webcenter',
name='WSRPSamples2', tokenType='WSS11_SAML_TOKEN_WITH_MSG_PROTECTION',
issuer='www.oracle.com', defUser='anonymous',
keyStorePath='/keys/mykeystore.jks', keyStorePswd='Test1',
sigKeyAlias='mysigalias', sigKeyPswd='mysigpswd', encKeyAlias='myencalias',
encKeyPswd='myencpswd', recptAlias='myrcptalias')
```

This example removes all the security properties set on a secure WSRP producer:

```
wls:/weblogic/serverConfig>setWSRPProducer (appName='webcenter',
name='WSRPSamples2', tokenType='')
```

2.14.3 listWSRPProducers

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists WSRP producer registration details for a named application.

Syntax

```
listWSRPProducers(appName, [name, verbose, server, [applicationVersion]])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Optional. Name of an existing WSRP producer. If omitted, connection details for all WSRP producers configured for this application are listed.
<i>verbose</i>	Optional. Displays WSRP producer connection details in verbose mode. Valid options are 1 (true) and 0 (false). When set to 1, <code>listWSRPProducers</code> lists all connection properties. When set to 0, <code>listWSRPProducers</code> lists connection names only. This argument defaults to 1. If you set this argument to 0, do not specify the <code>names</code> argument.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example lists all the WSRP producers registered with WebCenter Portal:

```
wls:/weblogic/serverConfig> listWSRPProducers(appName='webcenter', verbose=0)
```

```
-----  
WSRPSamples-connection  
-----
```

The following example lists detailed connection information for a WSRP producer registered as `WSRPSamples-connection` with WebCenter Portal:

```
wls:/weblogic/serverConfig> listWSRPProducers(appName='webcenter', name='WSRPSamples-connection', verbose=1)
```

```
-----  
WSRPSamples-connection  
-----  
Connection Name: WSRPSamples-connection  
Web Service Connection Name: WSRPSamples-connection-wsconn  
Proxy Host: None  
Proxy Port: None  
Timeout: 0  
WSDL URL: http://example.com:7777/portletapp/portlets/wsrp2?WSDL
```

2.14.4 deregisterWSRPProducer

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Deregisters a WSRP producer, and deletes the associated WSRP and web service connections.

Syntax

```
deregisterWSRPProducer(appName, name, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>name</i>	Name of an existing WSRP producer.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example deregisters the `WSRPSamples` producer in WebCenter Portal (webcenter):

```
wls:/weblogic/serverConfig> deregisterWSRPProducer(appName='webcenter',  
name='WSRPSamples')
```

2.14.5 listWSRPProducerRegistrationProperties

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists registration properties supported by a WSRP portlet producer.

Syntax

```
listWSRPProducerRegistrationProperties(appName, url, [proxyHost, proxyPort,  
server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.

Argument	Definition
<i>url</i>	<p>WSRP producer URL. The syntax will vary according to your WSRP implementation, for example:</p> <pre>http://host_name:port_number/context_root/portlets/wsrp2?WSDL</pre> <pre>http://host_name:port_number/context_root/portlets/wsrp1?WSDL</pre> <pre>http://host_name:port_number/context_root/portlets/?WSDL (WSRP 1.0 for backward compatibility)</pre> <p>Where:</p> <ul style="list-style-type: none"> • <i>host_name</i> is the server where your producer is deployed • <i>port_number</i> is the HTTP listener port number • <i>context_root</i> is the Web application's context root • <i>portlets[wsrp(1 2)]?WSDL</i> is static text. The text entered here depends on how the producer is deployed. <p>For example:</p> <pre>http://myhost:7778/MyPortletApp/portlets/wsrp2?WSDL</pre>
<i>proxyHost</i>	<p>Optional. Host name or IP address of the proxy server.</p> <p>A proxy is required when the application and the remote portlet producer are separated by a firewall and an HTTP proxy is needed to communicate with the producer.</p>
<i>proxyPort</i>	<p>Optional. Port number on which the proxy server listens.</p>
<i>server</i>	<p>Optional. Name of the managed server where the application is deployed. For example, <i>WC_Portal</i>.</p> <p>Required when applications with the same name are deployed to different servers and also when you have a cluster.</p>
<i>applicationVersion</i>	<p>Optional. Version number of the deployed application. Required if more than one version of the application is deployed.</p>

Example

The following example lists valid registration properties for the WSRP producer with the WSDL URL provided:

```
wls:/weblogic/serverConfig> listWSRPProducerRegistrationProperties
(appName='webcenter', url='http://myhost:9999/portletapp/portlets/wsrp2?WSDL')

Registration Property hint : hint text
Registration Property label : label text
Registration Property language : en
Registration Property name : {urn:xyz:wlp:prop:reg:registration}consumerRole
Registration Property value : None
```

2.14.6 listWSRPProducerUserCategories

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists any user categories that a WSRP producer might support. WebCenter Portal users can use the WLST command [mapWSRPProducerUserCategory](#) to map application roles to a producer's user category.

Syntax

```
listWSRPProducerUserCategories(appName, name, [server, [pplicationVersion]])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>name</i>	Name of an existing WSRP producer.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example displays the categories associated with a WSRP producer named WSRPSamples:

```
wls:/weblogic/serverConfig> listWSRPProducerUserCategories( appName='webcenter' ,
name='WSRPSamples' )
```

```
User Category Name : categoryTwo
User Category Description : Custom role two.
User Category Mapped Local Roles : None
```

```
User Category Name : categoryOne
User Category Description : Custom role one.
User Category Mapped Local Roles : None
```

2.14.7 mapWSRPProducerUserCategory

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Maps a role that is defined in the named application to a user category supported by a WSRP producer. The user categories may be found using [listWSRPProducerUserCategories](#).

Syntax

```
mapWSRPProducerUserCategory(appName, name, localRole, producerUserCategory,
[server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Name of an existing WSRP producer.
<i>localRole</i>	Name of the application role to be mapped.
<i>producerUserCategory</i>	WSRP producer user category to which the WebCenter Portal role will be mapped.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example maps the application role `admin` to the WSRP user category `wrsp-admin`:

```
wls:/weblogic/serverConfig> mapWSRPProducerUserCategory(appName='webcenter',
name='WSRPProducer1', localRole='admin', producerUserCategory='wsrp-admin')
```

2.14.8 registerPDKJavaProducer

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Creates a connection to an Oracle PDK-Java portlet producer and registers the Oracle PDK-Java producer with a named application.

Syntax

```
registerPDKJavaProducer(appName, name, url, [serviceId, proxyHost, proxyPort,
subscriberId, sharedKey, timeout, establishSession, externalApp, mapUser,
server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Connection name. The name must be unique (across all connection types) within the application.

Argument	Definition
<i>url</i>	<p>URL for the Oracle PDK-Java producer. Use the following syntax: <code>http://host_name:port_number/context_root/providers</code></p> <p>Where:</p> <ul style="list-style-type: none">• <code>host_name</code> is the server where the producer is deployed• <code>port_number</code> is the HTTP Listener port number• <code>context_root</code> is the web application's context root.• <code>providers</code> is static text. The text entered here depends on how the producer is deployed. <p>For example: <code>http://myHost:7778/myEnterprisePortlets/providers</code></p>
<i>serviceId</i>	<p>Optional. Service ID of the producer.</p> <p>PDK-Java enables you to deploy multiple producers under a single adapter servlet. Producers are identified by their unique service ID. A service ID is required only if the service ID is not appended to the URL end point.</p> <p>For example, the following URL endpoint requires <code>sample</code> as the service ID: <code>http://domain.example.com:7778/xyz/providers</code></p> <p>However, the following URL endpoint, does not require a service ID: <code>http://domain.example.com:7778/xyz/providers/sample</code></p> <p>The service ID is used to look up a file called <code><service_id>.properties</code>, which defines the characteristics of the producer, such as whether to display its test page. Use any value to create the service ID.</p>
<i>proxyHost</i>	<p>Optional. Host name or IP address of the proxy server.</p> <p>A proxy is required if the application and the remote portlet producer are separated by a firewall and an HTTP proxy is needed for communication with the producer.</p>
<i>proxyPort</i>	<p>Optional. Port number on which the proxy server listens. This argument defaults to 80.</p>
<i>sharedKey</i>	<p>Optional. Shared key used for message authentication with the remote producer. Message authentication ensures that the incoming messages are sent from a host with a shared key. This argument defaults to null.</p> <p>The shared key can contain between 10 and 20 alphanumeric characters.</p>
<i>subscriberId</i>	<p>Optional. Consumer's identifier, if required.</p> <p>When a producer is registered with an application, a call is made to the producer. During the call, the consumer (WebCenter Portal in this instance) passes the value for <code>subscriberId</code> to the producer. The producer may be coded to use the <code>subscriberId</code>.</p>
<i>timeout</i>	<p>Optional. Timeout setting for communications with the producer, in seconds. For example, the maximum time the producer may take to register, deregister, or display portlets on portal pages.</p> <p>This argument defaults to 30.</p> <p>Individual portlets may define their own timeout period, which takes precedence over the value expressed here.</p>

Argument	Definition
<i>establishSession</i>	Optional. Enable a user session when executing portlets from this producer. Valid values are 1 (true) and 0 (false). The default for this argument is 0. When sessions are enabled (1), the server maintains session-specific information, such as the user name. Message authentication uses sessions, so if a shared key is specified, this option should also be enabled. For sessionless communication between the producer and the server, specify 0.
<i>externalApp</i>	Optional. Name of the external application with which to associate the producer. Required if one of this producer's portlets requires authentication.
<i>mapUser</i>	Optional. Flag indicating whether the mapped user name from the external application should be passed to the producer. Valid values are 1 (true) and 0 (false). This argument defaults to 1.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example creates and registers an Oracle PDK-Java producer named JPDKSamples in WebCenter Portal (webcenter):

```
wls:/weblogic/serverConfig> registerPDKJavaProducer(appName='webcenter',
name='JPDKSamples', url='http://myhost:9999/jpdk/providers/sample')
```

2.14.9 setPDKJavaProducer

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Edits registration details for an existing PDK-Java producer.

Syntax

```
setPDKJavaProducer(appName, name, url, [serviceId, proxyHost, proxyPort,
subscriberId, sharedKey, timeout, establishSession, externalApp, mapUser,
server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>name</i>	Name of an existing PDK-Java producer.

Argument	Definition
<i>url</i>	<p>URL for the Oracle PDK-Java producer. Use the following syntax: <code>http://host_name:port_number/context_root/providers</code></p> <p>Where:</p> <ul style="list-style-type: none">• <code>host_name</code> is the server where the producer is deployed• <code>port_number</code> is the HTTP Listener port number• <code>context_root</code> is the Web application's context root.• <code>providers</code> is static text. The text entered here depends on how the producer is deployed. <p>For example: <code>http://myHost:7778/myEnterprisePortlets/providers</code></p>
<i>serviceId</i>	<p>Optional. Service ID of the producer.</p> <p>PDK-Java enables you to deploy multiple producers under a single adapter servlet. Producers are identified by their unique service ID. A service ID is required only if the service ID is not appended to the URL end point.</p> <p>For example the following URL endpoint requires <code>sample</code> as the service ID: <code>http://domain.example.com:7778/xyz/providers</code></p> <p>However, the following URL endpoint, does not require a service ID: <code>http://domain.example.com:7778/xyz/providers/sample</code></p> <p>The service ID is used to look up a file called <code><service_id>.properties</code>, which defines the characteristics of the producer, such as whether to display its test page. Use any value to create the service ID.</p>
<i>proxyHost</i>	<p>Optional. Host name or IP address of the proxy server.</p> <p>A proxy is required if the application and the remote portlet producer are separated by a firewall and an HTTP proxy is needed for communication with the producer.</p>
<i>proxyPort</i>	<p>Optional. Port number on which the proxy server listens.</p>
<i>subscriberId</i>	<p>Optional. Consumer's identifier, if required.</p> <p>When a producer is registered with an application, a call is made to the producer. During the call, the consumer (WebCenter Portal in this instance) passes the value for Subscriber ID to the producer. If the producer does not see the expected value for Subscriber ID, it might reject the registration call.</p>
<i>sharedKey</i>	<p>Optional. The shared key is used for message authentication with the remote producer. Message authentication ensures that the incoming messages are sent from a host with a shared key. You should enable sessions using the <code>sharedKey</code> argument, as well as the <code>establishSession</code> argument.</p>
<i>timeout</i>	<p>Optional. Timeout setting for communications with the producer, in seconds. For example, the maximum time the producer may take to register, deregister, or display portlets on portal pages.</p> <p>Individual portlets may define their own timeout period, which takes precedence over the value expressed here.</p>

Argument	Definition
<i>establishSession</i>	Optional. Enable a user session when executing portlets from this producer. Valid values are 1 (true) and 0 (false). You should enable sessions using the <i>establishSession</i> argument, as well as the <i>sharedKey</i> argument. When sessions are enabled (1), the server maintains session-specific information, such as the user name. Message authentication uses sessions, so if a shared key is specified, this option should also be enabled. For sessionless communication between the producer and the server, set to 0.
<i>externalApp</i>	Optional. Name of the external application associated with this producer.
<i>mapUser</i>	Optional. Flag indicating whether the mapped user name from the external application should be passed to the producer. Valid values are 1 (true) and 0 (false).
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <i>WC_Portal</i> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example changes a PDK-Java producer registered with *MyApp* to use a proxy server:

```
wls:/weblogic/serverConfig> setPDKJavaProducer(appName='MyApp',name='MyProducer',
url='http://myhost.com/jpdk/providers/sample', proxyHost='myproxy.com', proxyPort=80)
```

2.14.10 deregisterPDKJavaProducer

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Deregisters an Oracle PDK-Java producer and deletes the associated connection, for a named application.

Syntax

```
deregisterPDKJavaProducer(appName, name, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <i>webcenter</i> .
<i>name</i>	Name of an existing PDK-Java producer.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <i>WC_Portal</i> . Required when applications with the same name are deployed to different servers and also when you have a cluster.

Argument	Definition
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example deregisters the `wc-OmniPortlet` producer, and deletes the associated connection:

```
wls:/weblogic/serverConfig> deregisterPDKJavaProducer (appName='webcenter' ,
name='wc-OmniPortlet' )
```

```
Already in Domain Runtime Tree
Producer wc-OmniPortlet has been deregistered.
Already in Domain Runtime Tree
"wc-OmniPortlet" successfully deleted
Already in Domain Runtime Tree
"wc-OmniPortlet-urlconn" successfully deleted
```

2.14.11 listPDKJavaProducers

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists details for one or more Oracle PDK-Java producers registered with a named application.

Syntax

```
listPDKJavaProducers(appName, [name, verbose, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Optional. Name of an existing PDK-Java portlet producer. If omitted, connection details for all PDK-Java producers configured for this application are listed.
<i>verbose</i>	Optional. Displays PDK-Java producer connection details in verbose mode. Valid options are 1 (true) and 0 (false). When set to 1, <code>listPDKJavaProducers</code> lists all connection properties. When set to 0, <code>listPDKJavaProducers</code> lists connection names only. This argument defaults to 1. If you set this argument to 0, do not specify the <code>name</code> argument.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example lists all the connection properties (verbose mode) for the `JPDKSamples` producer:

```
wls:/weblogic/serverConfig> listPDKJavaProducers(appName='webcenter',
name='JPDKSamples', verbose=1)

-----
wc-OmniPortlet
-----
Service Id: None
Shared Key: None
External Application Id: None
Subscriber Id: None
URL: http://myhost:9999/portalTools/omniPortlet/providers/omniPortlet
```

2.14.12 refreshProducer

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Refreshes the metadata stored for a named producer to reflect the portlets that are currently offered by that producer.

Syntax

```
refreshProducer(appName, producerName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>producerName</i>	Name of an existing producer.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example refreshes the `WSRPSamples` producer in WebCenter Portal (`webcenter`):

```
wls:/weblogic/serverConfig> refreshProducer(appName='webcenter',
producerName='WSRPSamples')
```

Producer `WSRPSamples` has been refreshed.

2.14.13 listPortletClientConfig

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists portlet client's configuration for a named application.

Syntax

```
listPortletClientConfig(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example lists portlet client configuration for WebCenter Portal:

```
wls:/weblogic/serverConfig> listPortletClientConfig(appName='webcenter')
```

```
-----
Portlet Client Configuration Settings
-----
Application Striping: 0
Content Cache Enabled: 0
Maximum Content Cache Objects: 2000
Maximum Content Cache Size: 10000000
Default Timeout: 30
Maximum IFrame Querystring Length: 0
Maximum Resource URL Length: 1500
Maximum Timeout: 60
Minimum Timeout: 2
Parallel Pool Size: 20
Parallel Queue Size: 40
Maximum Rendition Cache Objects: 1000
Resource Proxy Path: /resource-proxy
Supported Locales: en, de, fr
```

2.14.14 setPortletClientConfig

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Edits the portlet client's configuration, for a named application. If you omit a parameter, the corresponding configuration setting remains unchanged.

 **Note:**

Configuration changes made using this WLST command are only effective after your restart the Managed Server on which the application is deployed. For details, see *Oracle Fusion Middleware Administering Oracle WebCenter Portal*.

Syntax

```
setPortletClientConfig(appName, [applicationStripping, contentCacheEnabled,
contentCacheMaxObjects, contentCacheMaxSize, defaultTimeout,
maximumIframeQueryStringLength, maximumResourceUrlLength, maximumTimeout,
minimumTimeout, parallelPoolSize, parallelQueueSize,
renditionCacheMaxObjects, resourceProxyPath, supportedLocales, server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>applicationStripping</i>	Optional. Specifies whether to enable application stripping. Valid values are 1 (true) and 0 (false). The default value is 0 (false).
<i>contentCacheEnabled</i>	Optional. Specifies whether to enable the portlet client's content cache. Valid values are 1 (true) and 0 (false). The default value is 1 (true).
<i>contentCacheMaxObjects</i>	Optional. Maximum number of objects to be stored in the portlet client's content cache. The default value is 0 (unlimited).
<i>contentCacheMaxSize</i>	Optional. Maximum size of the portlet client's content cache in bytes. The default value is 0 (unlimited).
<i>defaultTimeout</i>	Optional. Default timeout period in seconds for requests made to producers. The default value is 10 seconds.
<i>maximumIframeQueryStringLength</i>	Optional. Maximum length of portlet inline frame URLs before URL shortening is applied. The default value is 0 (always shorten URLs).
<i>maximumResourceUrlLength</i>	Optional. Maximum length of portlet resource URLs. The default value is 1500.
<i>maximumTimeout</i>	Optional. Maximum timeout period in seconds for requests made to producers. The default value is 300 seconds.
<i>minimumTimeout</i>	Optional. Minimum timeout period in seconds for requests made to producers. The default value is 1.
<i>parallelPoolSize</i>	Optional. Number of threads used for parallel execution of tasks. The default value is 10.
<i>parallelQueueSize</i>	Optional. Number of tasks to allow in queue for parallel execution. The default value is 20.

Argument	Definition
<i>renditionCacheMaxObjects</i>	Optional. Maximum number of objects in the cache used by the portlet client to store portlet renditions for use when rendering portlets in inline frames. The default value is 25.
<i>resourceProxyPath</i>	Optional. The base path of the portlet resource proxy servlet, relative to the context root of the application. The default value is <code>/resourceproxy</code> .
<i>supportedLocales</i>	Optional. Specifies one or more locales supported by the portlet client. The default value is <code>en</code> (English). Separate multiple locales with a comma. For example: <code>supportedLocales='en,fr,de'</code>
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example sets new values for the `ContentCacheEnabled`, `MinimumTimeout`, `MaximumTimeout`, and `ParallelQueueSize` configuration settings. All other settings remain unchanged:

```
wls:/weblogic/serverConfig> setPortletClientConfig(appName='webcenter',
contentCacheEnabled=0, minimumTimeout=10, maximumTimeout=120,
parallelQueueSize=25)
```

The following example sets the value of the `SupportedLocales` configuration setting to English, French, and Spanish. All other settings remain unchanged:

```
wls:/weblogic/serverConfig> setPortletClientConfig(appName='webcenter',
supportedLocales='en, fr, es')
```

2.14.15 getPortletClientConfig

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Returns the value of a specific portlet client configuration setting or values of all settings for a named application.

Syntax

```
getPortletClientConfig(appName, [configSetting, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .

Argument	Definition
<i>configSetting</i>	Optional. Name of the portlet client configuration setting to return. One of the following values: ApplicationStriping ContentCacheEnabled ContentCacheMaxObjects ContentCacheMaxSize DefaultTimeout MaximumIframeQueryStringLength MaximumResourceUrlLength MaximumTimeout MinimumTimeout ParallelPoolSize ParallelQueueSize RenditionCacheMaxObjects ResourceProxyPath SupportedLocales Note: The values are case-sensitive. Omit this parameter to return the names and values of all settings.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example returns the value of the `DefaultTimeout` configuration setting for WebCenter Portal:

```
wls:/weblogic/serverConfig> defaultTimeout =
getPortletClientConfig(appName='webcenter', configSetting='DefaultTimeout')
```

The following example returns the names and values of all portlet client configuration settings for WebCenter Portal:

```
wls:/weblogic/serverConfig> settingsDict =
getPortletClientConfig(appName='webcenter')
```

2.14.16 registerOOTBProducers

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Registers several out-of-the-box producers with WebCenter Portal: OmniPortlet and WSRP Tools

Syntax

```
registerOOTBProducers(producerHost, producerPort, appName, [server,
applicationVersion])
```

Argument	Definition
<i>producerHost</i>	Host name or IP address of the server hosting out-of-the-box producers. In a cluster fronted by a load balancer, enter the host name of the load balancer.
<i>producerPort</i>	Port number for the server hosting out-of-the-box producers. In a cluster, fronted by a load balancer, enter the port number of the load balancer.
<i>appName</i>	Name of the application in which the out-of-the-box producers are to be registered. For WebCenter Portal, the application name is always webcenter.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example registers out-of-the-box producers in WebCenter Portal.

```
wls:/weblogic/serverConfig> registerOOTBProducers(producerHost='myhost.com',
producerPort=9999, appName='webcenter')
```

```
Registering Out-of-the-Box Producers
Registering producers at http://myhost.com:9999
```

```
Registering Omniportlet
Created connection wc-OmniPortlet-urlconn
Created connection wc-OmniPortlet
Producer connection wc-OmniPortlet has been registered.
```

```
Registering WSRP Tools
Created connection wc-WSRPTools-wsconn
Created connection wc-WSRPTools
Producer connection wc-WSRPTools has been registered.
```

2.14.17 deregisterOOTBProducers

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Deregisters out-of-the-box producers with WebCenter Portal: OmniPortlet and WSRP Tools

Syntax

```
deregisterOOTBProducers(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which the out-of-the-box producers are currently registered. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example deregisters out-of-the-box producers, and deletes their associated connections, in WebCenter Portal:

```
wls:/weblogic/serverConfig> deregisterOOTBProducers(appName='webcenter')
```

```
Deregistering Out-of-the-Box Producers
```

```
Deregistering Omniportlet
Producer wc-OmniPortlet has been deregistered.
wc-OmniPortlet successfully deleted
wc-OmniPortlet-urlconn successfully deleted
```

```
Deregistering WSRP Tools
Producer wc-WSRPTools has been deregistered.
wc-WSRPTools successfully deleted
wc-WSRPTools-wsconn successfully deleted
```

2.14.18 registerSampleProducers

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Registers the sample producers provided with Oracle WebCenter Portal with a named application. There are two sample producers— WSRP Samples and JPDK Samples.

Syntax

```
registerSampleProducers(producerHost, producerPort, appName, [server,
applicationVersion])
```

Argument	Definition
<i>producerHost</i>	Host name or IP address of the server hosting the sample producers.
<i>producerPort</i>	Port number for the server hosting the sample producers.

Argument	Definition
<i>appName</i>	Name of the application in which the sample producers are to be registered. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example registers the sample producers with WebCenter Portal:

```
wls:/weblogic/serverConfig> registerSampleProducers(producerHost='myhost.com',
producerPort=9999, appName='webcenter')
```

2.14.19 deregisterSampleProducers

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Deregisters Oracle WebCenter Portal's sample producers (WSRP Samples and JPDK Samples) from a named application.

Syntax

```
deregisterSampleProducers(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which the sample producers are currently registered. For WebCenter Portal, the application name is always <code>webcenter</code> . If a value is not specified, this argument defaults to <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example deregisters sample producers from WebCenter Portal:

```
wls:/weblogic/serverConfig> deregisterSampleProducers(appName='webcenter')
```

2.15 Proxy Server

Use the commands listed in [Table 2-22](#) to manage proxy server settings used by tools and services, in a named application.

Configuration changes made using these WLST commands are only effective after you restart the Managed Server on which the application is deployed. For details, see *Oracle Fusion Middleware Administering Oracle WebCenter Portal*.

Table 2-22 RSS WLST Commands

Use this command...	To...	Use with WLST...
getWebCenterProxyConfig	Return the proxy host and proxy port used by the tools and services.	Online
setWebCenterProxyConfig	Specify the proxy host and proxy port used by tools and services.	Online
unsetWebCenterProxyConfig	Delete proxy host and proxy port settings.	Online

2.15.1 getWebCenterProxyConfig

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Returns the proxy host and proxy port used by RSS news feeds and activity streams in a named application. Depending on your network configuration, proxy details may be required to display external RSS news feeds and external links in activity streams in your application.

Syntax

```
getWebCenterProxyConfig(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example returns the proxy host and proxy port used by tools and services in WebCenter Portal (`webcenter`):

```
wls:/weblogic/serverConfig> getWebCenterProxyConfig(appName='webcenter')
```

2.15.2 setWebCenterProxyConfig

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Specifies the proxy host and proxy port used by RSS news feeds and activity streams, in a named application. Depending on your network configuration, proxy details may be required to display external RSS news feeds and external links in activity streams in your application.

Syntax

```
setWebCenterProxyConfig(appName, proxyHost, proxyPort, [server,  
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>proxyHost</i>	Host name of the proxy server.
<i>proxyPort</i>	Port on which the proxy server is running.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example sets the proxy host and proxy port used by tools and services in WebCenter Portal (webcenter):

```
wls:/weblogic/serverConfig> setWebCenterProxyConfig(appName='webcenter',  
proxyHost='www-proxy.example.com', proxyPort='80')
```

2.15.3 unsetWebCenterProxyConfig

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Deletes the current proxy host and proxy port settings configured for a named application.

Syntax

```
unsetWebCenterProxyConfig(appName, [server, applicationVersion])
```

Argument	Definition
<code>appName</code>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<code>server</code>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<code>applicationVersion</code>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example deletes the proxy host and proxy port settings configured for WebCenter Portal (`webcenter`):

```
wls:/weblogic/serverConfig> unsetWebCenterProxyConfig(appName='webcenter')
```

2.16 Search - Elasticsearch

Use the commands listed in [Table 2-23](#) to manage Elasticsearch (ES) connections for WebCenter Portal.

Configuration changes made using these WLST commands are only effective after your restart the Managed Server on which the application is deployed. For details, see *Oracle Fusion Middleware Administering Oracle WebCenter Portal*.

Table 2-23 Search - Elasticsearch WLST Commands

Use this command...	To...	Use with WLST...
createSearchConnection	Create a connection to an Elasticsearch instance for a named application.	Online
setSearchConnection	Modifies a search connection,	Online
listSearchConnections	Lists the Elasticsearch connection that is configured for an application.	Online

2.16.1 createSearchConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Creates a connection to an Elasticsearch instance for a named application.

Syntax

```
createSearchConnection(appName, name, url, indexAliasName, appUser, appPassword,
[server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>name</i>	Connection name. The name must be unique (across all connection types) within the application.
<i>url</i>	URL of the Elasticsearch server. Use the format: <code>http://eshost:esport</code> where <ul style="list-style-type: none"> <code>eshost</code> is the host name of the Elasticsearch server. <code>esport</code> is the port of the Elasticsearch server.
<i>indexAliasName</i>	Name of the index alias in the Elasticsearch server. For example, <code>webcenter_portal</code> . The indices is created using the alias as the prefix, as shown in the following example: <ul style="list-style-type: none"> <code><indexAliasName>_portals</code> For example, <code>webcenter_portal_portals</code> <code><indexAliasName>_documents</code> For example, <code>webcenter_portal_documents</code> <p>Note: The name must be in lowercase alphanumeric characters and unique across all portal servers.</p>
<i>appUser</i>	User name that the application uses to authenticate itself as a trusted application to Elasticsearch so that it may perform searches on behalf of WebCenter Portal users.
<i>appPassword</i>	Password for the user name specified.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example creates a new search connection that points to the specified Elasticsearch server and sets this connection as the default search connection for WebCenter Portal, an index named `webcenter_portal` is created on the Elasticsearch server:

```
createSearchConnection(appName='webcenter', name='ESConn', url='http://
eshost:esport', indexAliasName='webcenter_portal', appUser='wpadmin',
appPassword='password')
```

2.16.2 setSearchConnection

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Modifies an existing Elasticsearch connection.

Syntax

```
setSearchConnection(appName, name, indexAliasName, appUser, appPassword, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>name</i>	Name of an existing search connection.
<i>indexAliasName</i>	Name of the index alias in the Elasticsearch server. For example, webcenter_portal. The indices is created using the alias as the prefix, as shown in the following example: <ul style="list-style-type: none"> <indexAliasName>_portals For example, webcenter_portal_portals <indexAliasName>_documents For example, webcenter_portal_documents <p>Note: The name must be in lowercase alphanumeric characters and unique across all portal servers.</p>
<i>appUser</i>	User name that the application uses to authenticate itself as a trusted application to Elasticsearch server so that it may perform searches on behalf of WebCenter Portal users.
<i>appPassword</i>	Password for the user name specified.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example modifies the *indexAliasName* of a search connection named ESConn . An index alias named webcenter_portal is created on the Elasticsearch server:

```
setSearchConnection(appName='webcenter', name='ESConn',
indexAliasName='webcenter_portal', appUser='wpadmin', appPassword='password')
```

2.16.3 listSearchConnections

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists the Elasticsearch connection that is configured for an application.

Syntax

```
listSearchConnections(appName, [verbose, name, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>verbose</i>	Optional. Displays the details of the search connection in the verbose mode. The allowed values are 1 and 0. The default value is 0. <ul style="list-style-type: none"> When the value is set to 1, the search connection that is configured for an application is listed, along with the details. When the value is set to 0, only the search connection name is displayed. <div style="border: 1px solid #0070C0; padding: 10px; margin-top: 10px;"> <p> Note: If you set the value to 0, don't specify the <code>name</code> argument.</p> </div>
<i>name</i>	Optional. Name of an existing search connection. Use this argument to view details about a specific connection.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example displays the details for the Elasticsearch connection configured for WebCenter Portal:

```
listSearchConnections(appName='webcenter', verbose=1)
```

The following example displays the connection details for the Elasticsearch connection named `ESConn`:

```
listSearchConnections(appName='webcenter', verbose=1, name='ESConn')
```

2.17 WebCenter Portal Application

Use the commands listed in [Table 2-24](#) to manage workflow settings and metadata for WebCenter Portal.

Table 2-24 WebCenter Portal Application WLST Commands

Use This Command...	To...	Use with WLST...
getSpacesWorkflowConnectionName	Return the name of the BPEL server connection that WebCenter Portal is using for internal workflows.	Online
setSpacesWorkflowConnectionName	Specify the BPEL server connection used for WebCenter Portal workflows.	Online

2.17.1 getSpacesWorkflowConnectionName

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Returns the name of the BPEL server connection that WebCenter Portal is currently using for internal workflows (portal membership notifications, portal subscription requests, and so on).

Syntax

```
getSpacesWorkflowConnectionName(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where WebCenter Portal is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of WebCenter Portal is deployed.

Example

The following example names the BPEL server connection that WebCenter Portal is currently using for internal workflows:

```
wls:/weblogic/serverConfig> getSpacesWorkflowConnectionName(appName='webcenter')  
WorkflowConfigConnectionName: WebCenter-Worklist
```

2.17.2 setSpacesWorkflowConnectionName

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Specifies the BPEL server connection that WebCenter Portal uses for internal workflows. WebCenter Portal uses a BPEL server included with the Oracle SOA Suite to host internal workflows, such as portal membership notifications, portal subscription requests, and so on. The connection name specified here must be a valid BPEL server connection.

Note:

Configuration changes made using this WLST command are only effective after you restart the Managed Server on which WebCenter Portal is deployed. For details, see *Oracle Fusion Middleware Administering Oracle WebCenter Portal*.

Syntax

```
setSpacesWorkflowConnectionName(appName, name, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>name</i>	Name of an existing BPEL server connection.
<i>server</i>	Optional. Name of the managed server where WebCenter Portal is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of WebCenter Portal is deployed.

Example

The following example specifies that WebCenter Portal uses the BPEL server connection named `WebCenter-Worklist` for its internal workflows.

```
wls:/weblogic/serverConfig>setSpacesWorkflowConnectionName(appName='webcenter',
name='WebCenter-Worklist')
```

2.18 Identity Store

Use the commands listed in [Table 2-25](#) to configure options for searching an application's identity store.

Table 2-25 WebCenter Portal Identity Store WLST Commands

Use this command...	To...	Use with WLST...
setWebCenterIdStoreSearchConfig	Modify configuration options for searching a named applications's identity store.	Online
listWebCenterIdStoreSearchConfig	List current configuration options for searching a named application's identity store.	Online

2.18.1 setWebCenterIdStoreSearchConfig

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Modifies configuration options for searching a named application's identity store. Use these settings to optimize identity store searches (for users and roles) in WebCenter Portal.

Identity store search parameters are stored in `adf-config.xml`. If a search parameter is not specified, it is not modified.

Syntax

```
setWebCenterIdStoreSearchConfig(appName,[narrowSearchTimeout, broadSearchTimeout,
maxSearchFilters, maxFetchRecords, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>narrowSearchTimeout</i>	Optional. Maximum time allowed (in ms) for small, simple searches, such as fetching a single user from the identity store. Out-of-the-box, the default is 30000ms.
<i>broadSearchTimeout</i>	Optional. Maximum time allowed (in ms) to return large result sets, such as returning users and roles that match a name pattern. Out-of-the-box, the default is 60000.
<i>maxSearchFilters</i>	Optional. Number of search filters allowed for the application's identity store. The maximum allowed, out-of-the-box, is 100. Some identity store searches are executed using search filters which are converted into LDAP search calls. If your associated LDAP server limits the search condition, you can set the <code>maxSearchFilters</code> property to match your LDAP server setting.
<i>maxFetchRecords</i>	Optional. Maximum number of records to be returned from each search query. Out-of-the-box, the default is 100. The value of this setting will impact the performance of your LDAP server so take this into consideration when increasing the search result limit. Note that the LDAP server imposes its own search result limit, so the actual limit that is used will be the lesser of these two values.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example increases both identity store search timeouts.

```
wls:/weblogic/serverConfig>setWebCenterIdStoreSearchConfig(appName='webcenter',
narrowSearchTimeout=60000, broadSearchTimeout=100000);
```

The following example limits the maximum number of records returned to 100.

```
wls:/weblogic/serverConfig>setWebCenterIdStoreSearchConfig(appName='webcenter',
maxFetchRecords=100);
```

2.18.2 listWebCenterIdStoreSearchConfig

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists current configuration options for searching the identity store for a named application (WebCenter Portal).

Identity store search parameters are stored in `adf-config.xml`.

Syntax

```
listWebCenterIdStoreSearchConfig(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example displays identity store search configuration information for WebCenter Portal (`webcenter`).

```
wls:/weblogic/serverConfig>listWebCenterIdStoreSearchConfig(appName='webcenter');

-----
User role search configuration parameters
-----
Narrow search timeout      : 30000
Broad search timeout       : 60000
Maximum search filters     : 100
Maximum records to fetch  : 200
```

2.19 Lifecycle

Use the commands listed in [Table 2-26](#) to perform lifecycle operations for WebCenter Portal.

Table 2-26 Lifecycle WLST Commands

Use this command...	To...	Use with WLST...
deployWebCenterPortal	Deploy a portal from a stage environment to a production environment.	Online
propagateWebCenterPortal	Propagate metadata for a named portal, from a stage environment to a production environment.	Online
exportWebCenterPortals	Export one or more named portals to a portal archive (<code>.par</code> file).	Online
exportWebCenterPortalTemplates	Export one or more named portal templates to a portal archive (<code>.par</code> file).	Online
importWebCenterPortals	Import one or more portals or portal templates from a portal archive (<code>.par</code> file).	Online

Table 2-26 (Cont.) Lifecycle WLST Commands

Use this command...	To...	Use with WLST...
listWebCenterPortalArchive	List the content of a portal archive and extract the portal archive to a specified location	Online or Offline
exportWebCenterPortalConnections	Export connection configuration information from a source WebCenter Portal environment to a named file.	Online
importWebCenterPortalConnections	Import new WebCenter Portal connections from a named connection properties file.	Online
setSpaceState	Take a portal offline or bring a portal online.	Online
exportWebCenterResource	Export a portal asset to an export archive (.aar file).	Online
importWebCenterResource	Import a portal asset from an export archive (.aar file)	Online
importWebCenterTranslations	Import translations for WebCenter Portal.	Online
exportWebCenterApplication	Export the WebCenter Portal application to an export archive (.par file).	Online
importWebCenterApplication	Import the WebCenter Portal application from an export archive (.par file).	Online
exportPortletClientMetadata	Export portlet client metadata and producer customizations and personalizations to an export archive.	Online
importPortletClientMetadata	Import portlet client metadata and producer customizations and personalizations from an export archive.	Online
showProducerImportFailures	Display names of producers where metadata imports have failed and reasons for those failures.	Online
retryAllFailedProducerImports	Attempt to import outstanding producer metadata.	Online
cloneWebCenterManagedServer	Clone a Managed Server used by WebCenter Portal.	Online

2.19.1 deployWebCenterPortal

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Deploys a portal from a stage environment to a production environment.

 **Note:**

- The name of the managed server must be the same in both the stage and production environments.
- A connection to the production environment must be defined on the stage instance.
- You must have at least the WebLogic `Monitor` role and the WebCenter Portal permission `Portals - Manage Security and Configuration`.

Syntax

```
deployWebCenterPortal(appName, portalName, targetConnectionName,
[deployCustomizations, deployPortalContent, deploySecurity, deployData,
deployActivities, deploySharedAssets, deployConnections, overwrite, savePortal,
deployLog, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>portalName</i>	Name of the portal that you want to deploy. For example, <code>portalName='myPortal'</code> . Note: Do not enter the portal's display name here. You must enter the portal name that appears in the portal URL. If you are not sure, obtain the portal name from the <i>About Portal</i> dialog.
<i>targetConnectionName</i>	Name of a connection on the source stage instance that specifies how to connect to the target production instance. Note: Use Enterprise Manager or the WLST command <code>adf_createURLConnection</code> to configure the connection if it does not exist.
<i>deployCustomizations</i>	Optional. This attribute is deprecated.
<i>deployPortalContent</i>	Optional. Specifies whether to deploy the portal's content folder on the target server. Valid values are 1 and 0. <ul style="list-style-type: none"> • 1 - Deploys the portal's content folder. • 0 - Excludes portal content. This argument defaults to 0.
<i>deploySecurity</i>	Optional. This attribute is deprecated.
<i>deployData</i>	Optional. This attribute is deprecated.

Argument	Definition
<i>deployActivities</i>	<p>Optional. Indicates whether to deploy activity stream messages on the target. This option is only applicable when <code>deployData=1</code>.</p> <p>Valid values are 1 and 0.</p> <ul style="list-style-type: none"> 1 - Activity stream messages from the source portal are deployed on the target. 0 - Activity stream messages from the source portal are not deployed to the target. Activity on the target (if any) is preserved. This option is useful when migrating between stage and production environments and where test activity data is not required. <p>This argument defaults to 1.</p>
<i>deploySharedAssets</i>	<p>Optional. Indicates whether to deploy non-seeded shared assets to the target. Valid values are 1 and 0.</p> <ul style="list-style-type: none"> 1 - Deploy shared assets. 0 - Do not deploy shared assets. <p>This argument defaults to 0.</p>
<i>deployConnections</i>	<p>Optional. Indicates whether to deploy connections associated with portal to the target. Valid values are 1 and 0.</p> <ul style="list-style-type: none"> 1 - Deploy connections. 0 - Do not deploy connections. <p>This argument defaults to 1.</p>
<i>overwrite</i>	<p>Optional. Specifies whether to redeploy the portal if the portal already exists in the target. Valid values are 1 and 0.</p> <ul style="list-style-type: none"> 1 - Redeploys the portal. 0 - Prevents redeployment if the portal exists. <p>This argument defaults to 0.</p>
<i>savePortal</i>	<p>Optional. Specifies whether to back up the portal on the target instance before the portal is redeployed.</p> <p>Valid values are 1 and 0.</p> <ul style="list-style-type: none"> 1 - Backs up the portal before deployment. The backup archive is saved to the temporary directory on the target machine. 0 - Does not perform a portal back up before deployment. <p>The default value is 0.</p>
<i>deployLog</i>	<p>Optional. Name and location of a local log file containing detailed information about the deploy operation.</p> <p>If not specified, a deploy log file named <code>PortalDeploy_timestamp.log</code> is generated in the temporary directory.</p>
<i>server</i>	<p>Optional. Name of the managed server where WebCenter Portal is deployed. For example, <code>WC_Portal</code>.</p> <p>Required when applications with the same name are deployed to different servers and also when you have a cluster.</p>
<i>applicationVersion</i>	<p>Optional. Version number of the deployed application. Required if more than one version of the WebCenter Portal application is deployed.</p>

Example

The following example deploys a portal named `myPortal` without its content folder:

```
wls:/weblogic/serverConfig> deployWebCenterPortal(appName='webcenter',  
portalName='myPortal', targetConnectionName='MyProductionConnection')
```

 **Note:**

You can use the `adf_createHttpURLConnection` WLST command to create `MyProductionConnection`. For example:

```
adf_createHttpURLConnection(appName='webcenter',  
name='MyProductionConnection', url='http://example.com:7777',  
user='myuser', password='mypassword', realm='ProductionRealm')
```

The following example deploys a portal named `myPortal` with all its associated content, and also specifies a name and location for the deploy log file:

```
wls:/weblogic/serverConfig> deployWebCenterPortal(appName='webcenter',  
portalName='myPortal', targetConnectionName='MyProductionConnection',  
deployPortalContent=1, deployLog='/mydeploylogs/myPortal_deploy.log')
```

The following example backs up `myPortal` on the target before redeploying the portal and all its associated content:

```
wls:/weblogic/serverConfig> deployWebCenterPortal(appName='webcenter',  
portalName='myPortal', targetConnectionName='MyProductionConnection',  
deployPortalContent=1, savePortal=1, overwrite=1)
```

2.19.2 propagateWebCenterPortal

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Propagates metadata changes for a named portal, from a stage environment to a production environment.

Metadata changes propagated:

- **Include:**
 - Portal-level customizations (metadata changes) for portal pages, system pages, portlets, assets, task flows
 - User-level customizations (metadata changes) for portal pages, portlets, task flow instances
- **Exclude:** Security, any changes to content and data, and portal state (online/offline)

 **Note:**

You can only propagate portals that were previously deployed (from stage to production) using the `deployWebCenterPortal` command.

To migrate other changes, consider using [exportWebCenterPortals](#) and [importWebCenterPortals](#), or [deployWebCenterPortal](#).



Note:

- The name of the managed server must be the same in both the stage and production environments.
- A connection to the production environment must be defined on the stage instance.
- You must have at least the WebLogic `Monitor` role and the WebCenter Portal permission `Portals - Manage Security and Configuration`.

Syntax

```
propagateWebCenterPortal(appName, portalName, targetConnectionName, [savePortal,
propagateLog, propagateSharedAssets, propagatePortalContent, server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>portalName</i>	Name of the portal that you want to propagate. For example, <code>portalName='myPortal'</code> . Note: Do not enter the portal's display name here. You must enter the portal name that appears in the portal URL. If you are not sure, obtain the portal name from the <i>About Portal</i> dialog.
<i>targetConnectionName</i>	Name of a connection on the source stage instance that specifies how to connect to the target production instance. Note: Use Enterprise Manager or the WLST command <code>adf_createHttpURLConnection</code> to configure the connection if it does not exist.
<i>savePortal</i>	Optional. Specifies whether to back up the portal on the target instance to an archive before propagating the changes for the portal. Valid values are 1 and 0: <ul style="list-style-type: none"> • 1 - Back up the target portal to an archive before propagation. If created, the backup archive is saved to the <code>/tmp</code> directory on the target machine. • 0 - Do not back up the target portal before propagating changes. The default value is 0.
<i>propagateLog</i>	Optional. Name and location of a local log file containing detailed information about the propagation operation. If not specified, a propagation log file named <code>PortalPropagation_timestamp.log</code> is generated in the temporary directory.

Argument	Definition
<i>propagateSharedAssets</i>	Optional. Specifies whether to propagate non-seeded shared assets metadata changes to the target. Valid values are 1 and 0: <ul style="list-style-type: none"> 1 - Include shared assets changes. 0 - Do not include shared assets changes. The default value is 0.
<i>propagatePortalContent</i>	Optional. Specifies whether to propagate changes to content associated with the portal to the target instance. Valid values are 1 and 0: <ul style="list-style-type: none"> 1 - Propagate portal content. 0 - Exclude portal content. The default value is 0.
<i>server</i>	Optional. Name of the managed server where WebCenter Portal is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the WebCenter Portal application is deployed.

Example

The following example propagates metadata changes for a portal named `myPortal`:

```
wls:/weblogic/serverConfig> propagateWebCenterPortal(appName='webcenter',
portalName='myPortal', targetConnectionName='MyProductionConnection')
```

Note:

You can use the `adf_createURLConnection` WLST command to create `MyProductionConnection`. For example:

```
adf_createURLConnection(appName='webcenter',
name='MyProductionConnection', url='http://example.com:7777',
user='myuser', password='mypassword', realm='ProductionRealm')
```

The following example backs up a portal named `myPortal` with all its associated content before propagating metadata changes for the portal:

```
wls:/weblogic/serverConfig> propagateWebCenterPortal(appName='webcenter',
portalName='myPortal', targetConnectionName='MyProductionConnection',
savePortal=1)
```

The following example propagates metadata changes for a portal named `myPortal`, and also specifies a name and location for the propagation log file:

```
wls:/weblogic/serverConfig> propagateWebCenterPortal(appName='webcenter',
portalName='myPortal', targetConnectionName='MyProductionConnection',
propagateLog='/mypropagationlogs/myPortal_propagation.log')
```

2.19.3 exportWebCenterPortals

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Exports one or more named portals to a portal archive (.par file), using the filename specified.

The portal archive file contains one or more portal data archives (.pdr files) -- one for each portal that you export.

The export operation continues if one of the portals fail to export properly. Status messages are shown in the WLST console indicating the success or failure of the export operation for each specified portal. Portals that could not be exported are excluded from the .par file.



Note:

To run this command you must have at least the WebLogic Monitor role and the WebCenter Portal permission Portals - Manage Security and Configuration.

If your portal is larger than 2 GB, set `exportConnections = 0` in the `exportWebCenterPortals` command. Use the `exportWebCenterPortalConnections` command to migrate connections.

Syntax

```
exportWebCenterPortals(appName, fileName, [names, offlineDuringExport,
exportPortalContent, exportConnections, exportSharedAssets, server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>fileName</i>	Name and location of the export archive (.par file). For example, <code>/myExports/myPortalExport.par</code> . If you do not specify a location, the archive is created in the same directory that you run the WLST command.
<i>names</i>	Optional. Name of one or more portals that you want to export. Separate multiple portal names with a comma. For example: <code>names='sales,finance'</code> Note: Do not enter the portal's display names here. You must enter the name that is specified in the portal URL. The portal name is available from the <i>About Portal</i> dialog. If you leave the <code>names</code> argument blank, all the portals are exported (up to a maximum of ten portals). If your instance contains more than ten portals you must name each portal individually, using the <code>names</code> argument.

Argument	Definition
<i>offlineDuringExport</i>	<p>Optional. Indicates whether the portals you want to export must be offline before starting the export process. If users are allowed to access the portal, any changes made to the portal once the export starts are not exported.</p> <p>Valid values are 1 and 0.</p> <ul style="list-style-type: none"> 1 - Take online portals (if any) offline before starting the export process and at the end of the export process, return those portals back to their original online state. To prevent data loss during the export process, Oracle recommends that you set <code>offlineDuringExport=1</code>. 0 - Export the portals in their current state. <p>The default value is 0.</p>
<i>exportPortalContent</i>	<p>Optional. Specifies whether to export each portal's content folder on WebCenter Content Server.</p> <p>A content folder is automatically created on Content Server for any portal using document services to create, manage, and store portal documents (files, folders, wikis, blogs). Only content that is stored in this folder can be exported with the portal. The export does not, for example, include web content or pages displayed through Content Presenter since this information is not stored in the portal's content folder.</p> <p>Valid values are 1 and 0.</p> <ul style="list-style-type: none"> 1 - Export the portal's content folder. 0 - Exclude portal content. <p>The default value is 0.</p>
<i>exportConnections</i>	<p>Optional. Specifies whether to export WebCenter Portal connections into a file named <code>connection.properties</code> and include it in the portal archive (<code>.par</code> file).</p> <p>In addition, a copy of <code>connection.properties</code> is included outside the archive at the same location, that is, the location specified using the <code>fileName</code> option.</p> <p>Valid values are 1 and 0.</p> <ul style="list-style-type: none"> 1 - Export WebCenter Portal connections 0 - Do not export WebCenter Portal connections. <p>The default value is 0.</p> <p>See also, "Understanding Connection Property Files" in <i>Administering Oracle WebCenter Portal</i>.</p>
<i>exportSharedAssets</i>	<p>Optional. Specifies whether to export non-seeded shared assets and include it in the portal archive (<code>.par</code>). Valid values are 1 and 0.</p> <ul style="list-style-type: none"> 1 - Export shared assets. 0 - Do not export shared assets. <p>The default value is 0.</p>
<i>server</i>	<p>Optional. Name of the managed server where WebCenter Portal is deployed. For example, <code>WC_Portal</code>.</p> <p>Required when applications with the same name are deployed to different servers and also when you have a cluster.</p>
<i>applicationVersion</i>	<p>Optional. Version number of the deployed application. Required if more than one version of the WebCenter Portal application is deployed.</p>

Example

The following example exports `myPortal1` and `myPortal2` to `myPortalExport.par`:

```
wls:/weblogic/serverConfig> exportWebCenterPortals(appName='webcenter',
fileName='myPortalExport.par', names='myPortal1, myPortal2')
```

The following example exports `myPortal1` to `myPortalExport.par`. To ensure that `myPortal1` is offline while the export operation takes place, `offlineDuringExport` is set to 1:

```
wls:/weblogic/serverConfig> exportWebCenterPortals(appName='webcenter',
fileName='myPortalOfflineExport.par', names='myPortal1', offlineDuringExport=1)
```

The following example exports all the portals, including portal content folders, to `exportAllMyPortals.par`:

```
wls:/weblogic/serverConfig> exportWebCenterPortals(appName='webcenter',
fileName='exportAllMyPortals.par', exportPortalContent=1)
```

2.19.4 exportWebCenterPortalTemplates

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Exports one or more named portal templates to a portal archive (`.par` file), using the filename specified. The portal archive file contains a portal data archive (`.pdr` file) for each portal template you export.

Each portal data archive includes template-specific data (pages, discussions, and lists), customizations, security information, and optionally, portal template content.

The export operation continues if one of the portal templates fails to export properly. Status messages are shown in the WLST console indicating the success or failure of the export operation for each specified portal. Portal templates that could not be exported are excluded from the `.par` file.



Note:

To run this command you must have at least the WebLogic Monitor role and the WebCenter Portal permission `Portal Templates - Manage All`.

Syntax

```
exportWebCenterPortalTemplates(appName, fileName, [names, exportPortalTemplateContent,
exportConnections, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>fileName</i>	Name and location of the export archive (<code>.par</code> file). For example, <code>/myExports/myPortalTemplateExport.par</code> . If you do not specify a location, the archive is created in the same directory that you run the WLST command.

Argument	Definition
<i>names</i>	<p>Optional. Name of one or more portals templates that you want to export. Separate multiple template names with a comma. For example: names= 'mySalesTemplate ,myHRTemplate '</p> <p>Note: Do not enter portal template display names here. You must enter the name that is specified in the portal template URL. The portal name is available from the <i>About Portal Template</i> dialog.</p> <p>If you leave the <i>names</i> argument blank, all the templates are exported (up to a maximum of ten templates). If your instance contains more than ten portal templates you must name each template individually, using the <i>names</i> argument.</p>
<i>exportPortalTemplateContent</i>	<p>Optional. Specifies whether to export each template's content folder on WebCenter Content Server.</p> <p>A content folder is automatically created on Content Server for any template using document services to create, manage, and store template documents (files, folders, wikis, blogs). Only content that is stored in this folder can be exported with the portal template. The export does not, for example, include web content or pages displayed through Content Presenter since this information is not stored in the portal template's content folder.</p> <p>Valid values are 1 and 0.</p> <ul style="list-style-type: none"> • 1 - Export the portal template's content folder. • 0 - Exclude portal template content. <p>The default value is 0.</p>
<i>exportConnections</i>	<p>Optional. Specifies whether to extract WebCenter Portal connections into a file named "connection.properties" and include it in the portal archive (.par). Valid values are 1 and 0.</p> <p>Valid values are 1 and 0.</p> <ul style="list-style-type: none"> • 1 - Extract WebCenter Portal connections. • 0 - Do not extract WebCenter Portal connections. <p>The default value is 1.</p>
<i>server</i>	<p>Optional. Name of the managed server where WebCenter Portal is deployed. For example, WC_Portal.</p> <p>Required when applications with the same name are deployed to different servers and also when you have a cluster.</p>
<i>applicationVersion</i>	<p>Optional. Version number of the deployed application. Required if more than one version of the WebCenter Portal application is deployed.</p>

Example

The following example exports `myPortalTemplate1` and `myPortalTemplate2` to `myPortalTemplateExport.par`:

```
wls:/weblogic/serverConfig> exportWebCenterPortalTemplates(appName='webcenter',
fileName='myPortalTemplateExport.par', names='myPortalTemplate1,
myPortalTemplate2')
```

The following example exports all the portal templates, including portal template content, to `exportAllMyPortalTemplates.par`:

```
wls:/weblogic/serverConfig> exportWebCenterPortalTemplates(appName='webcenter',
fileName='exportAllMyPortalTemplates.par', exportPortalTemplateContent=1)
```

2.19.5 importWebCenterPortals

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Imports one or more portals or portal templates from a portal archive (.par file).



Note:

To run this command you must have at least the WebLogic `Monitor` role and either the WebCenter Portal permission `Portals - Portals - Manage Security and Configuration`, or `Portal Templates - Manage All`.

Syntax

```
importWebCenterPortals(appName, fileName, [names, parentPortal, importCustomizations,
importPortalContent, importSecurity, importData,
importActivities, overwrite, savePortals, forceOffline, importLog, importConnections,
connPropertiesFile, importSharedAssets, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>fileName</i>	Name of the portal archive (.par file) containing the portals or portal templates you want to import. For example, <code>mySalesPortal.par</code> .
<i>names</i>	Optional. Name of one or more portals or portal templates that you want to import. For example: <code>names='sales'</code> . Separate multiple names with a comma. For example: <code>names='sales,finance'</code> Note: Do not enter display names here. You must enter the name that is specified in the portal or portal template URL: <ul style="list-style-type: none"> Portal names are available from the <i>About Portal</i> dialog. Portal template names are available from the <i>About Portal Template</i> dialog. Note: If a .par file has multiple portals, then <code>names</code> is used to selectively import portals; that is, portal names specified against this parameter are imported. If this is omitted, all the portals in the .par file are imported. For a .par file with a single portal, this parameter can be ignored.
<i>parentPortal</i>	Optional. Name of the parent portal under which to place portals in the archive. If specified, imported portals become subportals of the parent portal. This argument defaults to null. When no parent is specified, archived portals are imported as root portals. Note: If the archive contains portal templates, this argument is ignored.
<i>importCustomizations</i>	Optional. This attribute is deprecated.

Argument	Definition
<i>importPortalContent</i>	<p>Optional. Specifies whether to import content associated with the portal on the instance. Valid values are 1 and 0.</p> <p>Valid values are 1 and 0.</p> <ul style="list-style-type: none"> 1 - Import portal content. 0 - Exclude portal content. This option is useful when migrating between stage and production environments where test content is no longer required. <p>This argument defaults to 0.</p> <p>Note: Portal archives that contain large content folders may exceed the maximum upload size for files (2 GB by default). If necessary, you can increase this setting. For details, see "Changing the Maximum File Upload Size" in <i>Oracle Fusion Middleware Administering Oracle WebCenter Portal</i>.</p>
<i>importSecurity</i>	Optional. This attribute is deprecated.
<i>importData</i>	Optional. This attribute is deprecated.
<i>importActivities</i>	<p>Optional. Indicates whether to import activity messages from the export archive. Valid values are 1 and 0.</p> <ul style="list-style-type: none"> 1 - Activity messages stored in the portal archive are imported. 0 - Activity messages are not imported. <p>This argument defaults to 1.</p> <p>Note: If the archive contains portal templates, this argument is ignored.</p>
<i>overwrite</i>	<p>Optional. Specifies whether to overwrite portals or portal templates that exist in the target.</p> <p>Valid values are 1 and 0.</p> <ul style="list-style-type: none"> 1 - Overwrite the portal. 0 - Prevent import if the portal exists. Only import new portals (or portal templates). <p>The default is 0.</p>
<i>savePortals</i>	<p>Optional. Specifies whether to save existing portals to an archive before starting the import operation.</p> <p>Valid values are 1 and 0:</p> <ul style="list-style-type: none"> 1 - Before importing, save any of the specified portals that exist on the target to an archive. If created, the archive is copied to the <code>/tmp</code> directory on the target machine. 0 - Do not save existing portals before the import operation. . <p>The default value is 0.</p>
<i>forceOffline</i>	<p>Optional. Specifies whether to take existing portals offline before starting the import process.</p> <p>Valid values are 1 and 0.</p> <ul style="list-style-type: none"> 1 - Take the portals offline before starting the import process. 0 - Attempt to import the portals. If one or more portals are currently online, you are prompted to take the portals offline. <p>The default value is 0.</p> <p>Note: If the archive contains portal templates, this argument is ignored.</p>
<i>importLog</i>	<p>Optional. Specifies the name of a local log file containing detailed information about the import operation.</p> <p>If not specified, a log file named <code>PortalImport_ timestamp.log</code> is generated in the temporary directory.</p>

Argument	Definition
<i>importConnections</i>	Optional. Specifies whether to import connections associated with the portal on the instance. Valid values are 1 and 0. Valid values are 1 and 0. <ul style="list-style-type: none"> 1 - Import connections used by the portal. 0 - Exclude connections. The default is 1.
<i>connPropertiesFile</i>	Optional. Specifies the name and location of a connection properties file. For example: /myConnections/connection.properties Web service and URL connections are imported from the specified connection properties file. Connections that do not exist on the target are imported. Connections that exist on the target are ignored. Note: If <i>importConnections</i> is set to 0, this argument is ignored.
<i>importSharedAssets</i>	Optional. Specifies whether to import non-seeded shared assets along with the portal (.par). sharedassets directory should be present in the archive. Valid values are 1 and 0. Valid values are 1 and 0. <ul style="list-style-type: none"> 1 - Import shared assets. 0 - Do not import shared assets. The default is 0.
<i>server</i>	Optional. Name of the managed server where WebCenter Portal is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the WebCenter Portal application is deployed.

Example

The following example imports a new version of the `sales` portal archived in `myPortalExport.par` and specifies a name and location for the import log file. To ensure that the existing `sales` portal is offline while the import operation takes place, `forceOffline` is set to 1:

```
wls:/weblogic/serverConfig> importWebCenterPortals(appName='webcenter',
fileName='myPortalExport.par', names='sales', forceOffline=1,
importLog='sales_import.log')
```

The following example saves a copy of the `sales` portal before re-importing a new version of the `sales` portal from `myPortalExport.par`:

```
wls:/weblogic/serverConfig> importWebCenterPortals(appName='webcenter',
fileName='myPortalExport.par', names='sales', savePortals=1)
```

The following example imports the `newHire` portal template archived in `myPortalTemplateExport.par` and specifies a name and location for the import log file:

```
wls:/weblogic/serverConfig> importWebCenterPortals(appName='webcenter',
fileName='myPortalTemplateExport.par', names='newHire', importLog='newHire.log')
```

2.19.6 listWebCenterPortalArchive

Module: Oracle WebCenter Portal

Use with WLST: Online or Offline

Description

Lists the content of a portal archive and optionally, extracts the portal archive to a specified location.

Syntax

```
listWebCenterPortalArchive(appName, fileName, [extractDir, server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>fileName</i>	Name and location of a portal archive.
<i>extractDir</i>	Optional. Destination directory in which to extract portal archive content. If a directory is specified, the portal archive is extracted to that location. If omitted, archive content is listed but not extracted.
<i>server</i>	Optional. Name of the managed server where WebCenter Portal is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the WebCenter Portal application is deployed.

Example

The following example lists the content of `myPortal.par` and extracts archive content to the directory `/myPortalArchives/myPortalContent`:

```
wls:/weblogic/serverConfig> listWebCenterPortalArchive(appName='webcenter',
fileName='/myPortalArchives/myPortal.par', extractDir='/myPortalArchives/
myPortalContent')
```

2.19.7 exportWebCenterPortalConnections

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Exports connection configuration information from the source WebCenter Portal environment to a named file.

The following connections are exported:

- Analytics collector connections
- Content repository connections (Oracle WebCenter Content)
- Discussions server connections
- Events server connections
- External application connections
- Mail server connections
- Search connections
- Portlet producer connections (WSRP and PDK-Java)
- Presence server connections
- URL connections
- Web service connections (used by data controls)
- BPEL server connections

 **Note:**

You must have at least the WebLogic Admin role to run the `exportWebCenterPortalConnections` command.

Syntax

```
exportWebCenterPortalConnections(appName, fileName, [connectionType, [connectionName,]  
logFile, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>fileName</i>	Specifies a name and location for the connection properties file. If you do not specify a location, the file is saved in the current directory.

Argument	Definition
<i>connectionType</i>	<p>Optional. Specifies the type of connections you want to export. Separate multiple connection types with a comma. For example:</p> <pre>connectionType='wsrpProducerConnection, webServiceConnection'</pre> <p>If left blank, all connection types are exported.</p> <p>Valid connection types are:</p> <ul style="list-style-type: none"> • <code>analyticsCollectorConnection</code> (Analytics collector) • <code>webcenterContentServerConnection</code> (Oracle WebCenter Content) • <code>discussionConnection</code> (Discussions server) • <code>personalEventConnection</code> (Presence server) • <code>externalAppConnection</code> (External application) • <code>mailConnection</code> (Mail server) • <code>searchConnection</code> (Oracle Secure Enterprise Search) • <code>wsrpProducerConnection</code> (WSRP portlet producer) • <code>jpdkProducerConnection</code> (PDK-Java portlet producer) • <code>impConnection</code> (Presence server) • <code>urlConnection</code> (URL) • <code>webServiceConnection</code> (Web service) • <code>bpelConnection</code> (BPEL server) • <code>restConnection</code> <p>Note: This argument is mandatory if you specify <code>connectionName</code>.</p>
<i>connectionName</i>	<p>Optional. Names specific connections you want to export for a single <code>connectionType</code>. Separate multiple connection names with a comma.</p> <p>For example, if <code>connectionType='wsrpProducerConnection'</code>, you can specify to export one or more connections of this type:</p> <pre>connectionName='myWSRPProducer1,myWSRPProducer2'</pre> <p>When you specify a value for <code>connectionName</code>, you must specify the <code>connectionType</code> argument as well. For example:</p> <pre>exportWebCenterPortalConnections(appName='webcenter', fileName='/scratch/conn.properties', connectionType='wsrpProducerConnection', connectionName='MyWSRPConn')</pre> <p>Note: If no names are specified, all connections are exported for the specified <code>connectionType</code>.</p>
<i>logFile</i>	<p>Specifies a name of a local log file containing detailed information about the export connection operation.</p> <p>If not specified, a log file named <code>ConnectionExport_<timestamp>.log</code> is generated in the temporary directory.</p>
<i>server</i>	<p>Optional. Name of the managed server where WebCenter Portal is deployed. For example, <code>WC_Portal</code>.</p> <p>Required when applications with the same name are deployed to different servers and also when you have a cluster.</p>
<i>applicationVersion</i>	<p>Optional. Version number of the deployed application. Required if more than one version of the WebCenter Portal application is deployed.</p>

Example

The following example exports connection configuration information for all WSRP producer and web service connections to a file named `connection.properties` located at `/myConnections`:

```
wls:/weblogic/serverConfig> exportWebCenterPortalConnections(appName='webcenter',
fileName='/myConnections/connection.properties',
connectionType='wsrpProducerConnection,webServiceConnection')
```

The following example exports connection configuration information for two WSRP producer connections named `myWSRP1` and `myWSRP2` to a file named `connection.properties` located at `/myConnections`:

```
wls:/weblogic/serverConfig> exportWebCenterPortalConnections(appName='webcenter',
fileName='/myConnections/connection.properties',
connectionType='wsrpProducerConnection', connectionName='myWSRP1,myWSRP2')
```

The following example exports all connection configuration information to a file named `connection.properties` located at `/myConnections`:

```
wls:/weblogic/serverConfig> exportWebCenterPortalConnections(appName='webcenter',
fileName='/myConnections/connection.properties')
```

The following example exports connection configuration information to a file named `connection.properties` located at `\myConnections`. Detailed information about the export operation is also logged to `exportConnections.log` located at `/myExportLogs`:

```
wls:/weblogic/serverConfig> exportWebCenterPortalConnections(appName='webcenter',
fileName='/myConnections/connection.properties', logFile='/myExportLogs/
exportConnections.log')
```

2.19.8 importWebCenterPortalConnections

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Imports new WebCenter Portal connections from a named connection properties file. Connections that do not exist on the target are imported. Connections that exist on the target are ignored.

The following connections can be imported:

- Analytics collector connections
- Content repository connections (Oracle WebCenter Content)
- Discussions server connections
- Events server connections
- External application connections
- Mail server connections
- Search connections
- Portlet producer connections (WSRP and PDK-Java)
- Presence server connections
- URL connections

- Web service connections (used by data controls)
- BPEL server connections

 **Note:**

- You must have at least the `WebLogic Admin` role to run the `importWebCenterPortalConnections` command.
- You can only import connection information that was previously exported using the [exportWebCenterPortalConnections](#) command.
- Newly imported portlet producer, external application, URL, and web service connections are immediately available in the target. For the other connection types, you are prompted to restart the managed server on which WebCenter Portal is deployed to make new connections available.

Syntax

```
importWebCenterPortalConnections(appName, fileName, [promptForPassword, logFile,
server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>fileName</i>	Specifies the name and location of a connection properties file.
<i>promptForPassword</i>	Optional. Specifies whether to prompt the user for a password if credentials are required to create a new connection during the import operation. Valid values are 1 and 0: <ul style="list-style-type: none"> • 1 - Prompts the user to enter credentials if a password is required. • 0 - Do not prompt the user to enter passwords if required to create a new connection. Always set this argument to 0 if you run this command within a script. The default is 1.
<i>logFile</i>	Optional. Specifies the name of a local log file in which to record detailed information about the import connection operation. If no value is specified, a log file named <code>ConnectionImport_<timestamp>.log</code> is generated in the temporary directory.
<i>server</i>	Optional. Name of the managed server where WebCenter Portal is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the WebCenter Portal application is deployed.

Example

The following example imports connections defined in a file named `connection.properties`, located at `/myConnections`:

```
wls:/weblogic/serverConfig> importWebCenterPortalConnections(appName='webcenter',
fileName='/myConnections/connection.properties')
```

The following example imports connections defined in a file named `connection.properties`, located at `\myConnections`. Detailed information about the import operation is also logged to `importConnection.log`:

```
wls:/weblogic/serverConfig> importWebCenterPortalConnections(appName='webcenter',
fileName='/myConnections/connection.properties', logFile='importConnection.log')
```

The following example imports connections defined in a file named `connection.properties`, located at `\myConnections`. Do not prompt the user to enter credentials (if required):

```
wls:/weblogic/serverConfig> importWebCenterPortalConnections(appName='webcenter',
fileName='/myConnections/connection.properties', promptForPassword=1)
```

2.19.9 setSpaceState

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Takes a portal offline or brings a portal online.

Syntax

```
setSpaceState(appName, spaceName, offline, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the name is always <code>webcenter</code> .
<i>spaceName</i>	Name of the portal you want to take offline or bring online.
<i>offline</i>	Specifies whether to take the portal offline or bring it back online. Valid values are 1 and 0: <ul style="list-style-type: none"> 1 takes the portal offline 0 brings the portal online
<i>server</i>	Optional. Name of the managed server where WebCenter Portal is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of WebCenter Portal is deployed.

Example

The following example takes `MyPortal` offline:

```
wls:/weblogic/serverConfig> setSpaceState(appName='webcenter', spaceName='MyPortal',
offline=1)
```

2.19.10 exportWebCenterResource

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Exports a single asset, device, or device group to an export archive (.aar file), using the filename specified.

When you export an asset, you can specify either `resourceGUID` or `resourceName`.

When you export a device or device group, you can only specify `resourceName`.

Note:

- To run this command you must have at least the `WebLogic Monitor` role, as well as the appropriate `Create/Edit/Delete` permission for the type of asset, device or device group you want to export.

For more information, see "Permissions Required to Perform WebCenter Portal Life Cycle Operations" in *Administering Oracle WebCenter Portal*.

- You cannot export out-of-the-box assets, devices, or device groups.

Syntax

```
exportWebCenterResource(appName, fileName, resourceType, [resourceGUID,  
resourceName, spaceName, exportContentDirectory, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the name is always <code>webcenter</code> .
<i>fileName</i>	Name of the local file to which the export will be written.

Argument	Definition
<i>resourceType</i>	<p>Type of resource to export (an asset, device, or device group). Valid values include: <code>pageTemplate</code>, <code>contentPresenter</code>, <code>pageStyle</code>, <code>resourceCatalog</code>, <code>skin</code>, <code>layout</code>, <code>taskFlow</code>, <code>dataControl</code>, <code>device</code>, <code>deviceGroup</code>, <code>dataSource</code>, <code>sqlDataSource</code>, <code>appIntgVisualization</code>, <code>visualizationTemplate</code>.</p> <p>Where:</p> <ul style="list-style-type: none"> • <code>pageTemplate</code> - page template • <code>contentPresenter</code> - Content presenter display template • <code>pageStyle</code> - page style • <code>resourceCatalog</code> - resource catalog • <code>skin</code> - skin • <code>taskFlow</code> - task flow • <code>taskflowStyle</code> - task flow style • <code>dataControl</code> - data control • <code>device</code> - device • <code>deviceGroup</code> - device group • <code>dataSource</code> - data source • <code>sqlDataSource</code> - SQL data source • <code>appIntgVisualization</code> - Application integration visualization • <code>visualizationTemplate</code> - Visualization template <p>You cannot export or import out-of-the-box assets, devices, or device groups.</p> <p>Note: In this release, <code>taskFlowStyle</code> replaces the asset type <code>mashupStyle</code>. The <code>mashupStyle</code> option is deprecated but continues to work in this release for backward compatibility.</p>
<i>resourceGUID</i>	<p>Optional. Unique ID (GUID) of an asset to export. Internal IDs are available from the <i>About</i> dialog for the asset. This argument is not used when <code>resourceType</code> is set to <code>device</code> or <code>deviceGroup</code>.</p>
<i>resourceName</i>	<p>Optional. Display name of an asset, device, or device group to export. Asset display names are available from the <i>About</i> dialog for the asset. Device and device group names are available from the <i>Edit</i> dialog for the device or device group. Note: You cannot export out-of-the-box assets, devices, or device groups.</p>
<i>spaceName</i>	<p>Optional. Name of the portal containing the asset to export. Use this argument to export portal assets, that is, assets that are owned by a particular portal. Omit this argument if you want to export shared assets for WebCenter Portal. This argument defaults to null (shared assets are exported). This argument is not used when <code>resourceType</code> is set to <code>device</code> or <code>deviceGroup</code>.</p>
<i>exportContentDirectory</i>	<p>Deprecated. You can no longer include MDS content associated with assets in export archives. Use the standard MDS WLST command <code>exportMetadata</code> to migrate legacy MDS content, if required.</p>

Argument	Definition
<code>server</code>	Optional. Name of the managed server where the application is deployed. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<code>applicationVersion</code>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example exports a page template owned by a portal named `MyPortal` (in WebCenter Portal) to a local file named `myPageTemplateExport.aar`:

```
wls:/weblogic/serverConfig> exportWebCenterResource(appName='webcenter',
fileName='myPageTemplateExport.aar', resourceType='pageTemplate',
resourceGUID='gsr47d9a5ac_7398_439a_97d2_8b54ce905f7e', spaceName='MyPortal')
```

The following example exports the same page template owned by a portal named `MyPortal` but specifies the template's display name rather than the GUID:

```
wls:/weblogic/serverConfig> exportWebCenterResource(appName='webcenter',
fileName='myPageTemplateExport.aar', resourceType='pageTemplate',
resourceName='MyPageTemplate', spaceName='MyPortal')
```

The following example exports a device named `MyMobileDevice` from WebCenter Portal:

```
wls:/weblogic/serverConfig> exportWebCenterResource(appName='webcenter',
fileName='myDeviceExport.aar', resourceType='device',
resourceName='MyMobileDevice')
```

2.19.11 importWebCenterResource

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Imports a single asset, device, or device group, from an asset export archive (.aar file).

Note:

- To run this command you must have at least the `WebLogicMonitor` role, as well as the appropriate `Create/Edit/Delete` permission for the type of asset, device or device group you want to import.
For more information, see "Permissions Required to Perform WebCenter Portal Life Cycle Operations" in *Administering Oracle WebCenter Portal*.
- You cannot export or import out-of-the-box assets, devices, or device groups.

Syntax

```
importWebCenterResource(appName, fileName, [resourceType, spaceName, server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the name is always <code>webcenter</code> .
<i>fileName</i>	Name of the archive file that you want to import.
<i>resourceType</i>	<p>Optional. Type of resource to import (an asset, device, or device group). Valid values include: <code>pageTemplate</code>, <code>contentPresenter</code>, <code>pageStyle</code>, <code>navigation</code>, <code>resourceCatalog</code>, <code>skin</code>, <code>taskFlow</code>, <code>mashupStyle</code>, <code>dataControl</code>, <code>device</code>, <code>deviceGroup</code></p> <p>Where:</p> <ul style="list-style-type: none"> • <code>pageTemplate</code> - page template • <code>contentPresenter</code> - Content presenter display template • <code>pageStyle</code> - page style • <code>navigation</code> - navigation models • <code>resourceCatalog</code> - resource catalog • <code>skin</code> - skin • <code>taskFlow</code> - task flow • <code>taskflowStyle</code> - task flow style • <code>dataControl</code> - data control • <code>device</code> - device • <code>deviceGroup</code> - device group <p>If the archive (.aar file) contains one or more devices or device groups, then this argument is mandatory:</p> <ul style="list-style-type: none"> • When <code>resourceType='device'</code>, all devices in the archive are imported. • When <code>resourceType='deviceGroup'</code>, all device groups in the archive and their associated devices are imported. <p>Note: In this release, <code>taskFlowStyle</code> replaces the asset type <code>mashupStyle</code>. The <code>mashupStyle</code> option is deprecated but continues to work in this release for backward compatibility.</p>
<i>spaceName</i>	<p>Optional. Name of the portal into which the asset is to be imported. Omit this argument if you want to import a shared asset into WebCenter Portal.</p> <p>This argument defaults to null (import shared asset). This argument is not used when <code>resourceType</code> is set to <code>device</code> or <code>deviceGroup</code>.</p>
<i>overwriteContentDirectory</i>	<p>Deprecated.</p> <p>You can no longer include MDS content associated with assets in export archives. Use the standard MDS WLST command <code>importMetadata</code> to migrate legacy MDS content, if required.</p>
<i>server</i>	<p>Optional. Name of the managed server where the application is deployed. Required when applications with the same name are deployed to different servers and also when you have a cluster.</p>
<i>applicationVersion</i>	<p>Optional. Version number of the deployed application. Required if more than one version of the application is deployed.</p>

Example

The following example imports a page template from an archive named `myPageTemplateExport.aar` to `MyPortal` in WebCenter Portal:

```
wls:/weblogic/serverConfig>
importWebCenterResource(appName='webcenter', fileName='myPageTemplateExport.aar',
spaceName='MyPortal', resourceType='pageTemplate')
```

The following example imports a device from an archive named `myDeviceExport.aar` to WebCenter Portal:

```
wls:/weblogic/serverConfig>
importWebCenterResource(appName='webcenter', fileName='myDeviceExport.aar',
resourceType='device')
```

2.19.12 importWebCenterTranslations

Module: Oracle WebCenter

Use with WLST: Online

Description

Imports translated content (XLF files) to MDS and the WebCenter Portal repository for use in WebCenter Portal.

Syntax

```
importWebCenterTranslations(appName, server, mdsRootDir, [applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the name is always <code>webcenter</code> .
<i>server</i>	Name of the target managed server on which WebCenter Portal is deployed. For example, <code>WC_Portal</code> .
<i>mdsRootDir</i>	MDS root directory on the file system that contains translated XLF files.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of WebCenter Portal is deployed.

Example

The following example imports translated content in the directory `/scratch/shared/newmd` to MDS and the WebCenter Portal repository:

```
wls:/weblogic/serverConfig> importWebCenterTranslations(appName='webcenter',
server='WC_Portal, mdsRootDir='/scratch/shared/newmd')
```

2.19.13 exportWebCenterApplication

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Exports an entire WebCenter Portal application to an export archive (.par file) using the filename provided.

Syntax

```
exportWebCenterApplication(appName, fileName, [connectionFileName,
exportCustomizations, exportData, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the name is always <code>webcenter</code> .
<i>fileName</i>	Name of the export archive (.par file) to which you want the export to be written.
<i>connectionFileName</i>	Optional. Name of the connections file to which export of connections is performed.
<i>exportCustomizations</i>	Optional. This attribute is deprecated.
<i>exportData</i>	Optional. This attribute is deprecated.
<i>server</i>	Optional. Name of the managed server where WebCenter Portal is deployed. For example, <code>WC_Portal</code> . Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of WebCenter Portal is deployed.

Example

The following example exports WebCenter Portal to a file named `myAppExport.par`, and exports connections to the `connection.properties` file.

```
wls:/weblogic/serverConfig>exportWebCenterApplication(appName='webcenter',
fileName='myAppExport.par', connectionFileName='connection.properties')
```

The following example exports a test WebCenter Portal instance to a file named `export.par`. In this case, data created during testing (such as lists, events, links, tags, and so on) is not exported.

```
wls:/weblogic/serverConfig>exportWebCenterApplication(appName='webcenter',
fileName='export.par')
```

2.19.14 importWebCenterApplication

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Imports an entire WebCenter Portal application from an export archive file to a managed server.

After importing WebCenter Portal, you must restart the managed server on which you deployed the application.

Syntax

```
importWebCenterApplication(appName, fileName, [connectionFileName,
importConnections, server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always webcenter.
<i>fileName</i>	Name of the export archive that you want to import.
<i>connectionFileName</i>	Optional. Name of the connections file from which import of connections is performed.
<i>importConnections</i>	Optional. Specifies whether to import connections from connectionFileName parameter or from the export archive. Valid values are 1 and 0. The default value is 1. 1 - Import connections 0 - Do not import connections
<i>server</i>	Optional. Name of the managed server where WebCenter Portal is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of WebCenter Portal is deployed.

Example

The following example imports WebCenter Portal from the export archive myAppExport.par.

```
wls:/weblogic/serverConfig> importWebCenterApplication(appName='webcenter',
fileName='myAppExport.par')
```

2.19.15 exportPortletClientMetadata

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Exports portlet client metadata and producer customizations and personalizations, for WebCenter Portal. This command exports metadata for *all* the application's producers to a named export archive (.ear file). You cannot opt to export metadata for specific producers.

Syntax

```
exportPortletClientMetadata(appName, fileName, [exportPersonalizations, server,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the name is always webcenter.

Argument	Definition
<i>fileName</i>	Name of the export archive (.ear file) to which you want the export to be written.
<i>exportPersonalizations</i>	Optional. Valid values are 1 (true) and 0 (false). <ul style="list-style-type: none"> 1 - Personalizations for <i>all</i> producers are exported. 0 - Personalizations are not exported. This argument defaults to 1.
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example exports portlet client metadata and producer customizations to an export archive named `myExport.ear`. Personalizations are not exported.

```
wls:/weblogic/serverConfig> exportPortletClientMetadata(appName='myApp',
fileName='myExport.ear', exportPersonalizations=0)
```

The following example exports portlet client metadata for an application with the version number `V2.0` deployed on the server `WC_CustomPortal1`.

```
wls:/weblogic/serverConfig> exportPortletClientMetadata(appName='myApp',
fileName='myExport.ear', server='WC_CustomPortal1', applicationVersion='V2.0')
```

2.19.16 importPortletClientMetadata

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Imports portlet client metadata and producer customizations and personalizations from a named export archive.

Producer personalizations are optional on export. Producer personalizations are imported if the export archive specified includes personalizations.

Syntax

```
importPortletClientMetadata(appName, fileName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the name is always <code>webcenter</code> .
<i>fileName</i>	Name of the export archive that you want to import.

Argument	Definition
<i>server</i>	Optional. Name of the managed server where the application is deployed. For example, WC_Portal. Required when applications with the same name are deployed to different servers and also when you have a cluster.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example imports portlet client metadata and producer customizations and personalizations from an export archive named `myExport.ear`.

```
wls:/weblogic/serverConfig> importPortletClientMetadata(appName='myApp',
fileName='myExport.ear')
```

2.19.17 showProducerImportFailures

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists outstanding producer imports for a named application.

Producer import fails if a producer used by the application is not available when the application first starts after deployment or an import operation.

Syntax

```
showProducerImportFailures(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the name is always <code>webcenter</code> .
<i>server</i>	Name of the managed server on which the application is deployed.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example shows producer import failures for WebCenter Portal (`webcenter`):

```
wls:/weblogic/serverConfig> showProducerImportFailures(appName='webcenter')
```

2.19.18 retryAllFailedProducerImports

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Imports outstanding producer metadata.

Producer import can fail if a producer used by the application is not available when the application first starts after deployment or an import operation. Use this command to import metadata for any producers for which metadata import previously failed.

Syntax

```
retryAllFailedProducerImports(appName, [server, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the name is always webcenter.
<i>server</i>	Name of the managed server on which the application is deployed.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example imports missing producer metadata for WebCenter Portal (webcenter):

```
wls:/weblogic/serverConfig> retryAllFailedProducerImports(appName='webcenter')
```

Importing metadata from the following producers failed for application webcenter.

```

Producer Description : Serialised stack trace:
[[
  oracle.portlet.client.container.PortletHttpException: HTTP <unknown method>
request to URL...
...
Failure Id : /oracle/adf/portlet/producerImportFailures/producerImportFailure2
Producer Id : /oracle/adf/portlet/JSR286FilePref
Producer Name : JSR286FilePref

```

Tried to re-import producer metadata for application webcenter.

Attempt to re-import producer metadata succeeded.



Note:

Errors and exceptions that occurred during a previous attempt to import producers display so you can see which failed producers the command is attempting to re-import.

2.19.19 cloneWebCenterManagedServer

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Creates a new managed server with the same resources as a specified, base managed server.

Syntax

```
cloneWebCenterManagedServer(baseManagedServer, newManagedServer,
newManagedServerPort, [verbose])
```

Argument	Definition
<i>baseManagedServer</i>	Name of the base managed server.
<i>newManagedServer</i>	Name for the new, cloned managed server.
<i>newManagedServerPort</i>	Port number for the new managed server.
<i>verbose</i>	Optional. Creates the managed server in verbose mode. Valid values are 1 and 0. When set to 1, additional progress information displays during the creation process which is useful for diagnostic purposes. The default is 0.

Example

The following example creates a clone of the WC_Portal managed server. The new managed server is named WC_Portal2:

```
wls:/weblogic/serverConfig>
cloneWebCenterManagedServer(baseManagedServer='WC_Portal',
newManagedServer='WC_Portal2', newManagedServerPort=1234)
```

2.20 Upgrade

Use the commands listed in [Table 2-27](#) when upgrading from a previous Oracle WebCenter Portal release.

Table 2-27 Oracle WebCenter Portal Upgrade WLST Commands

Use this command...	To...	Use with WLST...
upgradeWebCenterPortal	Upgrade WebCenter Portal.	Online
listDeprecatedFeaturesUsage	List deprecated connections, taskflows and portlets used in the upgraded WebCenter Portal application.	Online

2.20.1 upgradeWebCenterPortal

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Upgrades WebCenter Portal from release 11g to release 12c.

Oracle WebCenter Portal supports the FrameworkFolders folder service on Content Server. If your existing Oracle WebCenter Portal instance is configured to use Folders_g, this command migrates WebCenter Portal and WebCenter Content Server to FrameworkFolders, and upgrades WebCenter Portal to release 12c.

For more information about upgrade, see Upgrading Oracle WebCenter Portal in *Upgrading Oracle WebCenter*.



Note:

Before running the `upgradeWebCenterPortal` command, ensure that Node Manager is up and running.

Syntax

```
upgradeWebCenterPortal(appName, server, migrationDirectory, [contentServerName,
contentDbConnectionUrl, contentDbUserName, includeFolders, applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .
<i>server</i>	Name of the managed server where the application is deployed. For example, <code>WC_Spaces</code> . In a clustered environment where applications share the same content server connection (that is, the applications connect to the same content server and share the same root folder and security group) run this command against only one managed server.
<i>migrationDirectory</i>	Absolute directory path (with write permission) where upgrade logs will be written. If WebCenter Portal is configured to use Folders_g-based Content Server, this directory will contain data related to migration of Folders_g to FrameworkFolders. In case of a multi-node setup, this must be a shared directory that can be accessed and written onto from all nodes. Also, the directory must be accessible using the same path from all nodes.
<i>contentServerName</i>	Optional. Name of the managed server where WebCenter Content Server is deployed. For example, <code>UCM_server1</code> . Specify this argument if WebCenter Portal is configured to use Content Server.
<i>contentDbConnectionUrl</i>	Optional. Connection URL for the database where WebCenter Content schema (named OCS) is present. Use the connection URL format <code>host:port:sid</code> .
<i>contentDbUserName</i>	Optional. Note: Mandatory if <i>contentDbConnectionUrl</i> is specified. User name of the WebCenter Content schema (named OCS) that you want to migrate.

Argument	Definition
<i>includeFolders</i>	Optional. Comma separated list of folders to be included for migration from Folders_g to FrameworkFolders. The PersonalSpaces folder and the WebCenter Portal root folder are migrated by default. For example, if you want to migrate Contribution Folders and Common Templates, specify <code>includeFolders='Contribution Folders,Common Templates'</code> . After migration, Contribution Folders will be migrated as /Enterprise Libraries/Contribution Folders and Common Templates will be migrated as /Enterprise Libraries/Common Templates.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example upgrades WebCenter Portal deployed to the WC_Spaces managed server. The upgrade logs are written to the directory /tmp/upgrade.

```
wls:/weblogic/serverConfig> upgradeWebCenterPortal(appName='webcenter',
server='WC_Spaces', migrationDirectory='/tmp/upgrade');
```

The following example upgrades WebCenter Portal and WebCenter Content Server to FrameworkFolders and upgrades WebCenter Portal deployed to the WC_Spaces managed server . The upgrade logs are written to the directory /tmp/upgrade. In addition to the default folders, the Contribution Folders is also migrated.

```
wls:/weblogic/serverConfig> upgradeWebCenterPortal(appName='webcenter',
server='WC_Spaces', migrationDirectory='/tmp/upgrade',
contentServerName='UCM_server1',
contentDbConnectionUrl='wccdbhost.example.com:wccdbport:wccdbsid',
contentDbUserName='SCHEMA_PREFIX_OCS', includeFolders='Contribution Folders');
```

 **Note:**

Running the `upgradeWebCenterPermissions` WLST command displays certain error messages that permissions already exist. For example:

```
Already in Domain Runtime Tree
```

```
Command FAILED, Reason: JPS-04201: Cannot grant permission(s). Grant
already
exists for grantee [GranteeEntry: codeSource=null
principals=[[AppRole:
appID=webcenter name=webcenter#-#defaultadministrator
displayName=null
description=null category=null
uniquename=cn=webcenter\#-
\#defaultadministrator,cn=Roles,cn=webcenter,cn=wc_d
omain,cn=JPSContext,cn=jpsRoot guid=66FBB210983411E49F671B16134D61DE
members=[]
classname=oracle.security.jps.service.policystore.ApplicationRole
type=JPS_APPLICATION_ROLE]]].
```

Such error messages do not affect any functionality. You can safely ignore the error messages and proceed with upgrading your WebCenter Portal instance.

2.20.2 listDeprecatedFeaturesUsage

Module: Oracle WebCenter Portal

Use with WLST: Online

Description

Lists deprecated connections, taskflows and portlets used in the upgraded WebCenter Portal application. You can run this command to list deprecated usage in a single portal, multiple portals, or the entire application. This command also generates a detailed report. For more information, see *Upgrading Oracle WebCenter Portal* in *Upgrading Oracle WebCenter*.

Syntax

```
listDeprecatedFeaturesUsage(appName, server, reportDirectory, [portal,
applicationVersion])
```

Argument	Definition
<i>appName</i>	Name of the application in which to perform this operation. For WebCenter Portal, the application name is always <code>webcenter</code> .

Argument	Definition
<i>server</i>	Name of the managed server where the application is deployed. For example, WC_Spaces. In a clustered environment where applications share the same content server connection (that is, the applications connect to the same content server and share the same root folder and security group) run this command against only one managed server.
<i>reportDirectory</i>	Absolute directory path (with write permission) where reports will be written.
<i>portal</i>	Optional. Comma separated portal names for which report needs to be generated.
<i>applicationVersion</i>	Optional. Version number of the deployed application. Required if more than one version of the application is deployed.

Example

The following example lists the deprecated connections, taskflows and portlets used in WebCenter Portal deployed to the WC_Spaces managed server. The reports are written to the directory /tmp/report.

```
wls:/weblogic/serverConfig>listDeprecatedFeaturesUsage( appName= 'webcenter' ,  
server='WC_Portal' , reportDirectory='/tmp/report' )
```

The following example lists the deprecated connections, taskflows and portlets used in the HRPortal and FinancePortal portals deployed to the WC_Portal managed server. The reports are written to the directory /tmp/report.

```
wls:/weblogic/serverConfig>listDeprecatedFeaturesUsage( appName= 'webcenter' ,  
server='WC_Portal' , reportDirectory='/tmp/report' ,  
portal='HRPortal,FinancePortal' )
```

3

Oracle WebCenter Content Custom WLST Commands

This chapter provides detailed descriptions of custom WLST commands for Oracle WebCenter Content, including command syntax, arguments and command examples. The following sections describe the custom WLST commands for Oracle WebCenter Content. These commands enable you to configure and monitor the Oracle WebCenter Content server and the Oracle WebCenter Content Server instance from the command line. Topics include:

- [Overview of WLST WebCenter Content Command Categories](#)
- [WLST WebCenter Content Help](#)
- [Getter and Setter Methods Implementation](#)
- [Server Configuration Commands](#)
- [Email Configuration Commands](#)
- [System Status Commands](#)
- [General Configuration Commands](#)
- [Content Security Configuration Commands](#)
- [Component Manager Configuration Commands](#)
- [Records Management Configuration Commands](#)
- [User Interface Commands](#)
- [User Interface Connection Commands](#)

For additional information about Oracle WebCenter Content and Content Server administration and configuration, see *Administering Oracle Fusion Middleware*.



Note:

To use the Oracle WebCenter Content custom commands, you must invoke the WLST script from the Oracle Common home in which the component has been installed.

3.1 Overview of WLST WebCenter Content Command Categories

WLST WebCenter Content commands are divided into the following categories:

Table 3-1 WLST WebCenter Content Command Categories

Command Category	Description
Server Configuration Commands	View and manage server configuration options for the Content Server instance.
Email Configuration Commands	View and manage email configuration options for the Content Server instance.
System Status Commands	View system status information for the Content Server instance.
General Configuration Commands	View and manage general configuration options for the Content Server instance.
Content Security Configuration Commands	View and manage content security configuration options for the Content Server instance.
Component Manager Configuration Commands	View and manage Component Manager configuration options for the Content Server instance.
Records Management Configuration Commands	View and manage records management configuration options for the Content Server instance.
User Interface Commands	View and manage configuration for the optional Oracle WebCenter Content user interface.
User Interface Connection Commands	View and manage connections for the Oracle WebCenter Content user interface introduced in WebCenter Content 11g Release 1 (11.1.1.8)

3.2 WLST WebCenter Content Help

To view the WebCenter Content UCM commands that can be invoked from WLST, enter the following command at the WLST prompt:

```
help('UCM')
```

To view help for a specific Oracle Webcenter Content command, specify the name of the command; for example:

```
help('getUCMServerPort')
```

```
help('wccAdfConfig')
```

3.3 Getter and Setter Methods Implementation

The WLST component for Oracle Webcenter Content uses **getter** and **setter** methods to handle a situation where multiple applications register their corresponding Mbeans on a managed server, but WLST can talk to only one application.

Getter Method

The **getter** method is designed to handle zero or one argument.

If you do not provide an argument to an WLST WebCenter Content command, then one of two things occurs:

- If only one application has registered its Mbean on the server, then the WLST WebCenter Content command should work successfully and display the output.

- If multiple applications have registered Mbeans on the server, then an error message is displayed to prompt you to enter the specific application name in the argument.

If there is one argument to an WLST WebCenter Content command, then the following occurs:

- You must enter the correct application name when entering an argument. If the name is not entered properly, then an error message is displayed to prompt you to enter the valid application name in the argument.

Setter Method

The **setter** method is designed to handle one or two arguments.

- The first argument is the *value* to which you want to set the parameter.
- The second argument is the *application name*, which can be null or a string.

3.4 Server Configuration Commands

Use the commands in [Table 3-2](#) to configure the Oracle WebCenter Content Server instance.

Before you use these custom commands, set up the initial WLST connection as follows:

1. Set the environment variable `ORACLE_HOME` to `<Middleware_Home>/Oracle_ECM1`.
2. Run the WLST script from the following location: `<middleware_home>/Oracle_ECM1/common/bin`.
3. Connect to the WebCenter Content Server instance using the `connect()` command, for example, `connect("weblogic", "password", "t3://localhost:16200")`.

Table 3-2 WLST Server Configuration Commands

Use this command...	To...	Use with WLST...
getUCMHttpServerAddress	Display the HTTP Server Address value.	Online
getUCMServerPort	Display the Intradoc Server Port configuration parameter.	Online
setUCMServerPort	Set the Intradoc Server Port configuration parameter.	Online
getUCMIpAddressFilter	Display the IP Address Filter value.	Online
setUCMIpAddressFilter	Set the IP Address Filter value.	Online
getUCMUseSSL	Display the Use SSL value.	Online

3.4.1 getUCMHttpServerAddress

Use with WLST: Online

Description

Gets the HTTP Server Address value from the config.cfg file and displays it.

Syntax

```
getUCMHttpServerAddress(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the Oracle Webcenter Content HTTP server address for the application "Oracle Universal Content Management - Content Server":

```
getUCMHttpServerAddress('Oracle Universal Content Management - Content Server')
server.example.com
```

3.4.2 getUCMServerPort

Use with WLST: Online

Description

Gets the Intradoc Server Port configuration parameter from the `config.cfg` file and displays it.

Syntax

```
getUCMServerPort(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the Intradoc Server Port value for the application "Oracle Universal Content Management - Content Server":

```
getUCMServerPort('Oracle Universal Content Management - Content Server')
4442
```

3.4.3 setUCMServerPort

Use with WLST: Online

Description

Sets the Server Port configuration parameter.

Syntax

```
setUCMServerPort(value,['appName'])
```

Argument	Definition
<i>value</i>	Server Port number. This number must be a positive integer between 0 and 65535.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command sets the Server Port configuration parameter for the application "Oracle Universal Content Management - Content Server":

```
setUCMServerPort(4442,'Oracle Universal Content Management - Content Server')
```

3.4.4 getUCMIpAddressFilter

Use with WLST: Online

Description

Gets the IP Address Filter value from the `config.cfg` file and displays it.

Syntax

```
getUCMIpAddressFilter(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the IP address filter value for the application "Oracle Universal Content Management - Content Server":

```
getUCMIpAddressFilter('Oracle Universal Content Management - Content Server')  
10.131.123.*
```

3.4.5 setUCMIpAddressFilter

Use with WLST: Online

Description

Sets the Webcenter Content IP Address Filter value.

Syntax

```
setUCMIpAddressFilter(value,['appName'])
```

Argument	Definition
<i>value</i>	WebCenter Content IP Address Filter number. This number must be of <code>"*.*.*.*" format or IPV6 Format. The value must be taken from a list of IP Addresses allowed to communicate with the Content Server instance through the Intradoc Server Port.</code>
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command sets the value for the WebCenter Content IP address filter for the application "Oracle Universal Content Management - Content Server":

```
setUCMIpAddressFilter(10.131.123.*, 'Oracle Universal Content Management - Content Server')
```

3.4.6 getUCMUseSSL

Use with WLST: Online

Description

Gets the Use SSL value from the `config.cfg` file and displays it. The value can be True or False.

Syntax

```
getUCMUseSSL(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the Use SSL value for the application "Oracle Universal Content Management - Content Server":

```
getUCMUseSSL('Oracle Universal Content Management - Content Server')
True
```

3.5 Email Configuration Commands

Use the commands in [Table 3-3](#) to configure email for the Oracle WebCenter Content Server instance.

Table 3-3 WLST E-Mail Configuration Commands

Use this command...	To...	Use with WLST...
getUCMMailServer	Display the Mail Server value.	Online
setUCMMailServer	Set the Mail Server value.	Online
getUCMSmtpPort	Display the SMTP Port value.	Online
getUCMSysAdminAddresses	Display the Admin Address value.	Online
setUCMSysAdminAddresses	Set the Admin Address value.	Online

3.5.1 getUCMMailServer

Use with WLST: Online

Description

Gets the Mail Server value from the `config.cfg` file and displays it.

Syntax

```
getUCMMailServer(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the Mail Server value for the application "Oracle Universal Content Management - Content Server":

```
getUCMMailServer('Oracle Universal Content Management - Content Server')
mymailserver.example.com
```

3.5.2 setUCMMailServer

Use with WLST: Online

Description

Sets the Mail Server value in the `config.cfg` file.

Syntax

```
setUCMMailServer(value,['appName'])
```

Argument	Definition
<i>value</i>	Value for the Mail Server. The value is the name of the mail server that the Content Server instance uses to send SMTP based email.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command sets the value for the Mail Server for the application "Oracle Universal Content Management - Content Server":

```
setUCMMailServer(mymailserver.example.com,'Oracle Universal Content Management - Content Server')
```

3.5.3 getUCMSmtpPort

Use with WLST: Online

Description

Gets the SMTP Port value in the `config.cfg` file and displays it.

Syntax

```
getUCMSmtpPort(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the SMTP port value for the application "Oracle Universal Content Management - Content Server":

```
getUCMSmtpPort('Oracle Universal Content Management - Content Server')
4055
```

3.5.4 getUCMSysAdminAddress

Use with WLST: Online

Description

Gets the Admin Address value from the `config.cfg` file and displays it. The value can be of the form `abc@xyz.def`.

Syntax

```
getUCMSysAdminAddress(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the Admin Address value for the application "Oracle Universal Content Management - Content Server":

```
getUCMSysAdminAddress('Oracle Universal Content Management - Content Server')
mymail@example.com
```

3.5.5 setUCMSysAdminAddress

Use with WLST: Online

Description

Sets the Admin Address value in the `config.cfg` file.

Syntax

```
setUCMSysAdminAddress(value, ['AppName'])
```

Argument	Definition
<i>value</i>	Value for the Admin Address. The Admin Address can be of the form <code>abc@xyz.def</code> .
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command sets the Admin Address value for the application "Oracle Universal Content Management - Content Server":

```
setUCMSysAdminAddress(mymail@example.com, 'Oracle Universal Content Management - Content Server')
```

3.6 System Status Commands

Use the commands in [Table 3-4](#) to configure additional settings to monitor the WebCenter Content Server instance.

Table 3-4 WLST Additional Configuration Commands

Use this command...	To...	Use with WLST...
getUCMCSVersion	Display the version number.	Online
getUCMServerUptime	Display the uptime value.	Online

3.6.1 getUCMCSVersion

Use with WLST: Online

Description

Gets the version number of the Content Server running instance.

Syntax

```
getUCMCSVersion(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the version number of the active instance of the application "Oracle Universal Content Management - Content Server":

```
getUCMCSVersion('Oracle Universal Content Management - Content Server')
11g R1
```

3.6.2 getUCMServerUptime

Use with WLST: Online

Description

Gets the amount of time the Content Server instance has been up.

Syntax

```
getUCMServerUptime(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the amount of time the application "Oracle Universal Content Management - Content Server" has been up:

```
getUCMServerUptime('Oracle Universal Content Management - Content Server')
00H:01 Min:12 Sec
```

3.7 General Configuration Commands

Use the commands in [Table 3-5](#) to configure general configuration options for the Oracle WebCenter Content Server instance.

Table 3-5 WLST General Configuration Options Commands

Use this command...	To...	Use with WLST...
getUCMOverrideFormat	Display the <code>OverrideFormat</code> value.	Online
setUCMOverrideFormat	Set the <code>OverrideFormat</code> value.	Online
getUCMDownloadApplet	Display the <code>DownloadApplet</code> value.	Online
setUCMDownloadApplet	Set the <code>DownloadApplet</code> value.	Online
getUCMMultiUpload	Display the <code>MultiUpload</code> value.	Online
setUCMMultiUpload	Set the <code>MultiUpload</code> value.	Online
getUCMUseAccounts	Display the <code>UseAccount</code> value.	Online
setUCMUseAccounts	Set the <code>UseAccount</code> value.	Online
getUCMIsAutoNumber	Display the <code>AutoNumber</code> value.	Online
setUCMIsAutoNumber	Set the <code>AutoNumber</code> value.	Online
getUCMAutoNumberPrefix	Display the <code>AutoNumberPrefix</code> value.	Online
setUCMAutoNumberPrefix	Set the <code>AutoNumberPrefix</code> value.	Online
getUCMMajorRevLabelSeq	Display the <code>MajorRevLabelSeq</code> value.	Online
setUCMMajorRevLabelSeq	Set the <code>MajorRevLabelSeq</code> value.	Online
getUCMMinorRevLabelSeq	Display the <code>MinorRevLabelSeq</code> value.	Online
setUCMMinorRevLabelSeq	Set the <code>MinorRevLabelSeq</code> value.	Online
getUCMJspServerEnabled	Display the <code>JspServerEnabled</code> value	Online
setUCMJspServerEnabled	Set the <code>JspServerEnabled</code> value.	Online
getUCMJspEnabledGroups	Display the <code>JspEnabledGroups</code> value.	Online
setUCMJspEnabledGroups	Set the <code>JspEnabledGroups</code> value.	Online

3.7.1 getUCMOverrideFormat

Use with WLST: Online

Description

Gets the value from the `config.cfg` file, indicating whether the `OverrideFormat` parameter is set. The `OverrideFormat` parameter enables users to choose the application format of their content items.

Syntax

```
getUCMOverrideFormat(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the `OverRideFormat` parameter value:

```
getUCMOverRideFormat()
true
```

3.7.2 setUCMOverRideFormat

Use with WLST: Online

Description

Sets the `OverrideFormat` parameter to enable or disable the ability for users to choose the application format of their content items.

Syntax

```
getUCMOverRideFormat('value', ['appName'])
```

Argument	Definition
<i>value</i>	Specifies whether to enable or disable the <code>OverrideFormat</code> option. Values can be: Yes, No, True, False, 1, 0.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command enables the `OverRideFormat` parameter:

```
getUCMOverRideFormat('True')
```

3.7.3 getUCMDownloadApplet

Use with WLST: Online

Description

Gets the value from the `config.cfg` file indicating whether the `DownloadApplet` parameter is set. `DownloadApplet` enables users to download multiple files from a search results page.

Syntax

```
getUCMDownloadApplet(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command gets the value for the `DownloadApplet` parameter:

```
getUCMDownloadApplet()
true
```

3.7.4 setUCMDownloadApplet

Use with WLST: Online

Description

Sets the DownloadApplet parameter value to enable or disable the ability for users to download multiple files from a search results page.

Syntax

```
setUCMDownloadApplet('value', ['appName'])
```

Argument	Definition
<i>value</i>	Specifies whether to enable or disable the DownloadApplet option. Values can be: Yes, No, True, False, 1, 0.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command sets the value for the DownloadApplet to enable the functionality:

```
setUCMDownloadApplet('Yes')
```

3.7.5 getUCMMultiUpload

Use with WLST: Online

Description

Gets the value from the `config.cfg` file indicating whether the MultiUpload parameter is set. MultiUpload allows multiple files to be zipped and checked in as a single content item.

Syntax

```
getUCMMultiUpload(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the value for the MultiUpload parameter:

```
getUCMMultiUpload()
true
```

3.7.6 setUCMMultiUpload

Use with WLST: Online

Description

Sets the MultiUpload parameter value to allow or disallow multiple files to be zipped and checked in as a single content item.

Syntax

```
setUCMMultiUpload('value', ['appName'])
```

Argument	Definition
<i>value</i>	Specifies whether to enable or disable the MultiUpload option. Values can be: Yes, No, True, False, 1, 0.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command sets MultiUpload to allow multiple files to be zipped and check in as a single content item:

```
setUCMMultiUpload('1')
```

3.7.7 getUCMUseAccounts

Use with WLST: Online

Description

Gets the value from the `config.cfg` file indicating whether the UseAccounts parameter is set. UseAccounts enables the use of accounts in Oracle WebCenter Content.

Syntax

```
getUCMUseAccounts(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the value for the UseAccounts option:

```
getUCMUseAccounts()  
True
```

3.7.8 setUCMUseAccounts

Use with WLST: Online

Description

Sets the UseAccounts parameter value to enable to disable the use of accounts in Oracle WebCenter Content.

Syntax

```
setUCMUseAccounts('value', ['appName'])
```

Argument	Definition
<i>value</i>	Specifies whether to enable or disable the UseAccounts option. Values can be: Yes, No, True, False, 1, 0.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command sets UseAccounts to enable accounts in Oracle WebCenter Content:

```
setUCMUseAccounts('True')
```

3.7.9 getUCMIsAutoNumber

Use with WLST: Online

Description

Gets the value from the `config.cfg` file indicating whether the IsAutoNumber parameter is set. The IsAutoNumber parameter enables automatic numbering of Content IDs.

Syntax

```
getUCMIsAutoNumber(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the value for the IsAutoNumber parameter:

```
getUCMIsAutoNumber()  
True
```

3.7.10 setUCMIsAutoNumber

Use with WLST: Online

Description

Sets the IsAutoNumber parameter value to enable or disable automatic numbering of Content IDs.

Syntax

```
setUCMIsAutoNumber('value', ['appName'])
```

Argument	Definition
<i>value</i>	Specifies whether to enable or disable the AutoNumber option. Values can be: Yes, No, True, False, 1, 0.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command sets `IsAutoNumber` to enable automatic numbering of Content IDs:

```
setUCMIsAutoNumber('True')
```

3.7.11 getUCMAutoNumberPrefix

Use with WLST: Online

Description

Gets the value from the `config.cfg` file for the `AutoNumberPrefix` parameter. The prefix is used in all automatically numbered content IDs for newly checked-in files, if the `AutoNumber` parameter is enabled.

Syntax

```
getUCMAutoNumberPrefix(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the value for the `AutoNumberPrefix` parameter:

```
getUCMAutoNumberPrefix()  
dadvm10231usor
```

3.7.12 setUCMAutoNumberPrefix

Use with WLST: Online

Description

Sets the `AutoNumberPrefix` parameter value to a prefix used in all automatically numbered content IDs for newly checked-in files (if the `AutoNumber` parameter is enabled).

Syntax

```
setUCMUseAutoNumberPrefix('value', ['appName'])
```

Argument	Definition
<i>value</i>	The prefix used in all automatically numbered content IDs for newly checked-in files. Only applies if the <code>AutoNumber</code> parameter is enabled.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command sets the `AutoNumberPrefix`:

```
setUCMAutoNumberPrefix('dadvm10231usor')
```

3.7.13 getUCMMajorRevLabelSeq

Use with WLST: Online

Description

Gets the value from the `config.cfg` file for `MajorRevLabelSeq`, which defines the major sequence for revision numbers. `MajorRevLabelSeq` is the first part of the Revision Label.

Syntax

```
getUCMMajorRevLabelSeq(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the value for the `MajorRevLabelSeq` parameter:

```
getUCMMajorRevLabelSeq()
A1
```

3.7.14 setUCMMajorRevLabelSeq

Use with WLST: Online

Description

Sets the value for the `MajorRevLabelSeq` parameter, which defines the major sequence for revision numbers. `MajorRevLabelSeq` is the first part of the Revision Label.

Syntax

```
setUCMMajorRevLabelSeq('value', ['appName'])
```

Argument	Definition
<i>value</i>	Value can be any of the following: A through D, 1 through 8.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command sets `MajorRevLabelSeq` to 'A1':

```
setUCMMajorRevLabelSeq('A1')
```

3.7.15 getUCMMinorRevLabelSeq

Use with WLST: Online

Description

Gets the value from the `config.cfg` file for `MinorRevLabelSeq`, which defines the minor sequence for revision numbers. `MinorRevLabelSeq` is the second part of the Revision Label.

Syntax

```
getUCMMinorRevLabelSeq(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the value for the MinorRevLabelSeq parameter:

```
getUCMMinorRevLabelSeq()
b2
```

3.7.16 setUCMMinorRevLabelSeq

Use with WLST: Online

Description

Sets the value for the MinorRevLabelSeq parameter, which defines the minor sequence for revision numbers. MinorRevLabelSeq is the second part of the Revision Label.

Syntax

```
setUCMMinorRevLabelSeq('value', ['appName'])
```

Argument	Definition
<i>value</i>	Value can be any of the following: a through c, 1 through 7.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command sets MinorRevLabelSeq to 'b2':

```
setUCMMinorRevLabelSeq('b2')
```

3.7.17 getUCMJspServerEnabled

Use with WLST: Online

Description

Gets the value from the `config.cfg` file indicating whether the JspServerEnabled is set. If the parameter is turned on, Content Server can execute Java Server Pages. The Java Server Pages must be checked in to Content Server.

Syntax

```
getUCMJspServerEnabled(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the value for the JspServerEnabled parameter:

```
getUCMJspServerEnabled()
true
```

3.7.18 setUCMJspServerEnabled

Use with WLST: Online

Description

Sets the value for the JspServerEnabled parameter. If the parameter is turned on, Content Server can execute Java Server Pages. The Java Server Pages must be checked in to Content Server.

Syntax

```
setUCMJspServerEnabled('value', ['appName'])
```

Argument	Definition
<i>value</i>	Specifies whether to enable or disable the parameter. Values can be: Yes, No, True, False, 1, 0.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command sets JspServerEnabled to 'true':

```
setUCMJspServerEnabled('true')
```

3.7.19 getUCMJspEnabledGroups

Use with WLST: Online

Description

Gets the value from the `config.cfg` file for the JspEnabledGroups parameter and lists it. JspEnabledGroups lists security groups enabled for Java Server Page functionality. Security groups have certain permissions for contributors and administrators.

Syntax

```
getUCMJspEnabledGroups(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the value for the JspEnabledGroups parameter:

```
getUCMJspEnabledGroups()
group1
Jsp
```

3.7.20 setUCMJspEnabledGroups

Use with WLST: Online

Description

Sets the value for the `JspEnabledGroups` parameter, which specifies security groups to be enabled for Java Server Page functionality.

Syntax

```
setUCMJspEnabledGroups('value', ['appName'])
```

Argument	Definition
<i>value</i>	Specifies the security groups to be enabled for Java Server Page functionality.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command sets `JspEnabledGroups` to 'group1':

```
setUCMJspEnabledGroups('group1')
```

3.8 Content Security Configuration Commands

Use the commands in [Table 3-6](#) to configure content security options for the Oracle WebCenter Content Server instance.

Table 3-6 WLST Content Security Configuration Commands

Use this command...	To...	Use with WLST...
getUCMCopyAccess	Display the <code>CopyAccess</code> parameter value.	Online
setUCMCopyAccess	Set the <code>CopyAccess</code> parameter value.	Online
getUCMExclusiveCheckout	Display the <code>ExclusiveCheckout</code> parameter value.	Online
setUCMExclusiveCheckout	Set the <code>ExclusiveCheckout</code> parameter value.	Online
getUCMAuthorDelete	Display the <code>AuthorDelete</code> parameter value.	Online
setUCMAuthorDelete	Set the <code>AuthorDelete</code> parameter value.	Online
getUCMShowOnlyKnownAccounts	Display the <code>ShowOnlyKnownAccounts</code> parameter value.	Online
setUCMShowOnlyKnownAccounts	Set the <code>ShowOnlyKnownAccounts</code> parameter value.	Online

3.8.1 getUCMCopyAccess

Use with WLST: Online

Description

Gets the value from the `config.cfg` file and displays it. When `CopyAccess` is enabled, users with Read privilege on a content item can get a copy of the native file.

Syntax

```
getUCMCopyAccess(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the CopyAccess value:

```
getUCMCopyAccess( )
True
```

3.8.2 setUCMCopyAccess

Use with WLST: Online

Description

Sets the CopyAccess value in the `config.cfg` file. When CopyAccess is enabled, users with Read privilege on a content item can get a copy of the native file.

Syntax

```
setUCMCopyAccess( 'value' , [ 'appName' ] )
```

Argument	Definition
<i>value</i>	Specifies whether to enable or disable the parameter. Values can be: Yes, No, True, False, 1, 0.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command sets the value to enable the CopyAccess parameter:

```
setUCMCopyAccess( 'True' )
```

3.8.3 getUCMExclusiveCheckout

Use with WLST: Online

Description

Gets the value in the `config.cfg` file and displays it. When the ExclusiveCheckout parameter is enabled, Admin privilege is required to check out a content item checked in by another user.

Syntax

```
getUCMExclusiveCheckout( [ 'appName' ] )
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the value for the ExclusiveCheckout parameter:

```
getUCMExclusiveCheckout()
True
```

3.8.4 setUCMExclusiveCheckout

Use with WLST: Online

Description

Sets the value in the `config.cfg` file. When the `ExclusiveCheckout` parameter is enabled, Admin privilege is required to check out a content item checked in by another user.

Syntax

```
setUCMExclusiveCheckout('value', ['appName'])
```

Argument	Definition
<i>value</i>	Specifies whether to enable or disable the parameter. Values can be: Yes, No, True, False, 1, 0.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command sets the value to enable the `ExclusiveCheckout` parameter:

```
setUCMExclusiveCheckout('True')
```

3.8.5 getUCMAuthorDelete

Use with WLST: Online

Description

Gets the value from the `config.cfg` file and displays it. When the `AuthorDelete` parameter is enabled, authors are allowed to delete their revisions without having Delete privilege.

Syntax

```
getUCMAuthorDelete(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the `AuthorDelete` parameter value:

```
getUCMAuthorDelete()
1
```

3.8.6 setUCMAuthorDelete

Use with WLST: Online

Description

Sets the AuthorDelete parameter value in the `config.cfg` file. When the AuthorDelete parameter is enabled, authors are allowed to delete their revisions without having Delete privilege.

Syntax

```
setUCMAuthorDelete('value', ['appName'])
```

Argument	Definition
<i>value</i>	Specifies whether to enable or disable the parameter. Values can be: Yes, No, True, False, 1, 0.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command enables the AuthorDelete parameter:

```
setUCMAuthorDelete('1')
```

3.8.7 getUCMShowOnlyKnownAccounts

Use with WLST: Online

Description

Gets the value from the `config.cfg` file and displays it. When the ShowOnlyKnownAccounts parameter is enabled, the list of Content Server accounts on a check-in page will contain only globally-predefined accounts.

Syntax

```
getUCMShowOnlyKnownAccounts(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the ShowOnlyKnownAccounts parameter value as enabled:

```
getUCMShowOnlyKnownAccounts()  
Yes
```

3.8.8 setUCMShowOnlyKnownAccounts

Use with WLST: Online

Description

Sets the ShowOnlyKnownAccount parameter value in the `config.cfg` file. When the ShowOnlyKnownAccounts parameter is enabled, the list of Content Server accounts on a check-in page will contain only globally-predefined accounts.

Syntax

```
setUCMShowOnlyKnownAccounts('value', ['appName'])
```

Argument	Definition
<i>value</i>	Specifies whether to enable or disable the parameter. Values can be: Yes, No, True, False, 1, 0.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command enables the ShowOnlyKnownAccounts parameter:

```
setUCMShowOnlyKnownAccounts('yes')
```

3.9 Component Manager Configuration Commands

Use the commands in [Table 3-7](#) to configure Component Manager options for the Oracle WebCenter Content Server instance.

Table 3-7 WLST Component Manager Configuration Commands

Use this command...	To...	Use with WLST...
getUCMComponentStatus	Display the status of a component.	Online
setUCMComponentStatus	Set the status of a component.	Online
installUCMComponent	Install a component.	Online
uninstallUCMComponent	Uninstall a component.	Online
getUCMComponentConfig	Display the configuration for a component.	Online
updateUCMComponentConfig	Set configuration parameters for a component.	Online

3.9.1 getUCMComponentStatus

Use with WLST: Online

Description

Gets the status of a component. The status can be enabled or disabled.

Syntax

```
getUCMComponentStatus('componentName', ['appName'])
```

Argument	Definition
<i>componentName</i>	Specifies a valid Content Server component name. For example, 'ContentFolios'.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the status of the component "ContentFolios":

```
getUCMComponentStatus('ContentFolios')
Enabled
```

3.9.2 setUCMComponentStatus

Use with WLST: Online

Description

Sets the status of a component in the `config.cfg` file.

Syntax

```
setUCMComponentStatus('componentName','status',['appName'])
```

Argument	Definition
<i>componentName</i>	Specifies a valid Content Server component name. For example, 'ContentFolios'.
<i>status</i>	Specifies whether the status of the component is enabled or disabled. Values can be: Enable, Disable.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command sets the status of 'ContentFolios' to 'Enable':

```
setUCMComponentStatus('ContentFolios','Enable')
```

3.9.3 installUCMComponent

Use with WLST: Online

Description

Installs the component present at the specified file location.

Syntax

```
installUCMComponent('filePath',['appName'])
```

Argument	Definition
<i>filePath</i>	Specifies a valid file path to a component.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command installs the component at the location 'C:/manifest.zip':

```
installUCMComponent('C:/manifest.zip')
```

3.9.4 uninstallUCMComponent

Use with WLST: Online

Description

Uninstalls the specified component.

Syntax

```
uninstallUCMComponent('componentName', ['appName'])
```

Argument	Definition
<i>componentName</i>	Specifies a valid Content Server component name. For example, 'ContentFolios'.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command uninstalls the component named 'ContentFolios':

```
uninstallUCMComponent('ContentFolios')
```

3.9.5 downloadUCMComponent

Use with WLST: Online

Description

Downloads the specified component to the specified file location.

Syntax

```
downloadUCMComponent('componentName', 'filePath', ['appName'])
```

Argument	Definition
<i>componentName</i>	Specifies a valid Content Server component name. For example, 'ContentFolios'.
<i>filePath</i>	Specifies a valid file path to a component.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command downloads the component 'ContentFolios' to 'C:/manifest.zip':

```
downloadUCMComponent('ContentFolios', 'C:/manifest.zip')
```

3.9.6 getUCMComponentConfig

Use with WLST: Online

Description

Gets the configuration of the specified component and displays it.

Syntax

```
getUCMComponentConfig('componentName', ['appName'])
```

Argument	Definition
<i>componentName</i>	Specifies a valid component name. For example, 'ContentFolios'.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the configuration for the component 'ContentFolios':

```
getUCMComponentConfig('ContentFolios')
```

3.9.7 updateUCMComponentConfig

Use with WLST: Online

Description

Sets the specified component with the configuration options provided.

Syntax

```
updateUCMComponentConfig('componentName', 'updateParams', ['appName'])
```

Argument	Definition
<i>componentName</i>	Specifies a valid component name. For example, 'Folders_g'.
<i>updateParams</i>	Specifies valid parameters for the component.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command sets configuration parameters for the component 'Folders_g':

```
updateUCMComponentConfig('Folders_g', 'GetCopyAccess:true,CollectionHiddenMeta:xHidden')
```

3.10 Records Management Configuration Commands

Use the commands in [Table 3-8](#) to configure records management options for the Oracle WebCenter Content Server instance.

Table 3-8 WLST Records Management Configuration Commands

Use this command...	To...	Use with WLST...
getRMLevel	Display the type of records management configuration.	Online
getRMConfigurationLevel	Display the records management configuration level.	Online
getRMFeatures	Display records management features.	Online
getRMDispositionActions	Display records management dispositions actions.	Online
rmUpdate	Update the records management configuration.	Online
addOutgoingProvider	Add an outgoing provider for the Adapter server used to store records management content.	Online
registerSource	Register the source for the Adapter server repository so records management can find it.	Online

3.10.1 getRMLevel

Use with WLST: Online

Description

Gets the type of records management configuration and displays it. The type can have the following values:

- none
- standalone
- adapter

Syntax

```
getRMLevel(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the type of records management configuration:

```
getRMLevel()  
adapter
```

3.10.2 getRMConfigurationLevel

Use with WLST: Online

Description

Gets the records management configuration level and displays it. The configuration level can have the following values:

- **minimal**: Enables minimal amount of functionality and excludes some disposition actions and most of the application features. This is the default when the software is enabled.
- **typical**: Enables all disposition actions and all features except for DoD Configuration, Classified Topics, FOIA/PA tracking (Freedom of Information Act/Privacy Act), and E-mail.
- **dod2**: Enables the features from a Typical installation with the addition of DoD Configuration and E-mail.
- **dodclassified**: Enables all features except for FOIA/PA.
- **custom**: Enables the ability to choose a variety of features. Some disposition actions are dependent on other actions. If an action is selected, dependent actions are also automatically selected.

Syntax

```
getRMConfigurationLevel(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the type of records management configuration:

```
getRMConfigurationLevel(['appName'])
minimal
```

3.10.3 getRMFeatures

Use with WLST: Online

Description

Gets a list of records management features and displays it. The list can have the following values:

- feature_related_content
- feature_audit_trigger
- feature_subject_to_review
- feature_revision_dates
- feature_security_markings
- feature_email_fields
- feature_dod_config
- feature_foia_privacyact

Syntax

```
getRMFeatures(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays the records management features:

```
getRMFeatures()
feature_related_content
feature_audit_trigger
feature_subject_to_review
```

3.10.4 getRMDispositionActions

Use with WLST: Online

Description

Gets the values for records management disposition actions and displays them. The list can have the following values:

- actions_activate
- actions_obsolete
- actions_cancel
- actions_rescind
- actions_expire
- actions_cutoff
- actions_approve_deletion
- actions_destroy

Syntax

```
getRMDispositionActions(['appName'])
```

Argument	Definition
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command displays a list of records management disposition actions:

```
getRMDispositionActions()
actions_active
actions_obsolete
actions_cancel
```

3.10.5 rmUpdate

Use with WLST: Online

Description

Sets the records management configuration with feature and disposition actions and enables it.

Syntax

```
rmUpdate(urmLevel, level, featuresList, dispositionActionsList, enableRTMandURMAgent,['appName'])
```

Argument	Definition
<i>urmLevel</i>	Specifies a valid records management configuration type.
<i>level</i>	Specifies a valid records management level.
<i>featuresList</i>	Specifies valid records management features.
<i>dispositionActionsList</i>	Specifies valid disposition actions.
<i>enableRTMandURMAgent</i>	Specifies whether to enable the RTM and URM Agent.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command sets the records management configuration and enables it:

```
rmUpdate(urmLevel='standalone',level='custom',featuresList='feature_related_content:feature_revision_dates',dispositionActionsList='actions_activate:actions_obsolete',enableRTMandURMAgent='1')
```

3.10.6 addOutgoingProvider

Use with WLST: Online

Description

Defines the outgoing provider that enables the Adapter server to connect to the Content Server instance with records management enabled.

Syntax

```
addOutgoingProvider(ProviderName, ProviderDescription, ServerHostName, HTTPServerAddress, ServerPort, InstanceName, RelativeWebRoot, extraUpdateParams,['appName'])
```

Argument	Definition
<i>ProviderName</i>	Name of the outgoing provider.
<i>ProviderDescription</i>	Description of the outgoing provider.
<i>ServerHostname</i>	Name of the server host.
<i>HTTPServerAddress</i>	Address of the HTTP server.
<i>ServerPort</i>	Number of the server port.
<i>InstanceName</i>	Name of the instance for the Content Server with records management enabled.
<i>RelativeWebRoot</i>	Name of the relative web root.
<i>extraUpdateParams</i>	Extra parameters that can be used are: ClientHostIpFilter, ConnectionPassword, ConnectionPasswordName, IsRefinery, ProviderClass, ProviderConfig, ProviderConnection, RefineryMaxProJobs, RefineryReadOnly.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command defines an outgoing provider for the Adapter server to connect to the records management instance.:

```
addOutgoingProvider(ProviderName='x',ProviderDescription='y',ServerHostName='localhost',HTTPServerAddress='z',ServerPort='4444',InstanceName='w',RelativeWebRoot='cs',updateParams='RefineryMasJobs:100,IsRefinery:1')
```

3.10.7 registerSource

Use with WLST: Online

Description

Registration ensures that records management is aware of the Adapter and is ready to manage the stored content in the Adapter server's repository.

Syntax

```
registerSource(ProviderName, SourceName, SourceTableName, SourceDisplayName, ['appName'])
```

Argument	Definition
<i>ProviderName</i>	Name of the outgoing provider.
<i>SourceName</i>	Description of the source in the Adapter repository.
<i>SourceTableName</i>	Name of the source table in the Adapter repository.
<i>SourceDisplayName</i>	Display name for the source in the Adapter repository.
<i>appName</i>	Optional. Name of the deployed application.

Example

The following command registers the Adapter server's repository source for the records management stored content:

```
registerSource(ProviderName='x',SourceName='y',SourceTableName='z',SourceDisplayName='w')
```

3.11 User Interface Commands

Use the commands in [Table 3-9](#) to display and update the configuration of the Oracle WebCenter Content user interface introduced in Oracle WebCenter Content 11g Release 1 (11.1.1.8). For more information about this user interface, see *Getting Started with the WebCenter Content User Interface* in *Using Oracle WebCenter Content*.

To use these custom commands, you must invoke the WLST script from the appropriate Oracle home. Do not use the WLST script in the WebLogic Server home. For Oracle WebCenter Content user interface commands, the script is located at:

- UNIX: MW_HOME/oracle_common/common/bin/wlst.sh
- Windows: MW_HOME\oracle_common\common\bin\wlst.cmd

The WLST process must connect to the WebCenter Content UI server before you run the commands listed in the following table.

You can connect to the Content UI server instance using the `connect()` command, for example:

```
connect("weblogic","password","t3://localhost:16225").
```

Configuration changes made using these WLST commands are only effective after you restart the Managed Server on which the WebCenter Content - Web UI application is deployed.

Table 3-9 WLST WebCenter Content User Interface Commands

Use this command...	To...	Use with WLST...
displayWccAdfConfig	Display the configuration of the WebCenter Content user interface application.	Online
updateWccAdfConfig	Update the configuration of the WebCenter Content user interface application.	Online

3.11.1 displayWccAdfConfig

Use with WLST: Online

Description

Displays the configuration of the Oracle WebCenter Content user interface application.

Syntax

```
displayWccAdfConfig(appName='value',[attrName='value'])
```

Argument	Definition
<i>appName value</i>	Name of the deployed application.
<i>attrName value</i>	Optional. Name of the attribute to display If omitted all attributes are displayed.

Examples

The following command displays all configuration attributes for the Oracle WebCenter Content user interface deployed with the application name "Oracle WebCenter Content - Web UI".

```
wls:/wccadf_domain/serverConfig> displayWccAdfConfig(appName='Oracle WebCenter
Content - Web UI')
WccInstanceName = Default
WccInstanceConnectionName = WccAdfServerConnection
ProxyContent = true
DocumentPreview = DOCUMENT_VIEWER
DefaultLocale = en_US
DefaultTimeZone = America/Los_Angeles
ApplicationUrl = null
ClusterCompatible = false
TemporaryDirectory = null
MaximumUploadedFileSize = 52428800
MaximumWindowsPerSession = 7
WccLoginPageEnabled = true
SkinFamily = wcc-skin
SkinVersion = null
CustomBrandingLogo = null
CustomBrandingTitle = null
WccCustomizationLayerValues = null
OracleCustomizationLayerValues = null
CustomerCustomizationLayerValues = null
CustomRequestBinderProperties = null
UnfiledDocumentProhibited = false
viewerPagesCountForSinglePageMode = 400
disableDownloadForReadOnlyUser = false
```

The following command displays the value of the configuration attribute named "WccInstanceConnectionName".

```
wls:/wccadf_domain/serverConfig> displayWccAdfConfig(appName='Oracle WebCenter
Content - Web UI', attrName='WccInstanceConnectionName')
Attribute WccInstanceConnectionName value is: WccAdfServerConnection
```

3.11.2 updateWccAdfConfig

Use with WLST: Online

Description

Updates the configuration of the Oracle WebCenter Content user interface application. The command can be used to update a single attribute or multiple attributes.

Syntax

```
updateWccAdfConfig(appName, [wccInstanceName], [wccInstanceConnectionName],
[proxyContent], [documentPreview], [defaultLocale], [defaultTimeZone],
[applicationUrl], [clusterCompatible], [temporaryDirectory],
[maximumUploadedFileSize], [maximumWindowsPerSession],
[wccLoginPageEnabled], [skinFamily], [skinVersion], [customBrandingLogo],
[customBrandingTitle], [wccCustomizationLayerValues],
[oracleCustomizationLayerValues], [customerCustomizationLayerValues],
[customRequestBinderProperties], [unfiledDocumentProhibited],
[viewerPagesCountForSinglePageMode],
[disableDownloadForReadOnlyUser]
```

Argument	Definition
<i>appName</i>	Name of the deployed application.
<i>wccInstanceName</i>	Reserved for future use.
<i>wccInstanceConnectionName</i>	Name of the Content Server connection in Connection Architecture (<code>connections.xml</code>).
<i>proxyContent</i>	Controls how a browser retrieves document content, including the native file, the web-viewable rendition, thumbnails, and attachments. Must be "true" or "false". If "true" the browser requests this content from the Oracle WebCenter Content user interface, which in turn retrieves it from Content Server. If "false" the browser requests this content directly from Content Server.
<i>documentPreview</i>	Controls the preview displayed for a document. Must be one of the following case-sensitive values: <ul style="list-style-type: none"> DOCUMENT_VIEWER: Use the document viewer (if available) WEB_VIEWABLE: Display the web-viewable rendition (if available) NONE: No preview
<i>defaultLocale</i>	Locale to use if a user's locale cannot be otherwise determined. Must be in the format used by the Java class <code>java.util.Locale</code> (for example, "en_US").
<i>defaultTimeZone</i>	Time zone to use if the user's time zone cannot be otherwise determined. Must be in the format used by the Java method <code>java.util.TimeZone#getTimeZone</code> (for example, "America/Los_Angeles").
<i>applicationUrl</i>	Specifies the scheme, hostname, and port of URLs generated by the Oracle WebCenter Content user interface (for example, <code>https://wcc.example.com:16225</code>). Optional. If not set, the scheme, hostname, and port are determined from the HTTP request.

Argument	Definition
<i>clusterCompatible</i>	Specifies whether the Oracle WebCenter Content user interface supports session replication among nodes in a cluster. Either "true" or "false". Must be "true" if session replication is enabled on the Java EE Web Container.
<i>temporaryDirectory</i>	Base directory for temporary files created by the Oracle WebCenter Content user interface, such as in-progress uploads. This directory should be empty, on a fast file system with adequate free space, and readable and writable by only the owner of the process running the Oracle WebCenter Content user interface. The Oracle WebCenter Content user interface will create and manage a subdirectory structure. If clusterCompatible is "true" this directory must also be on a filesystem shared across all nodes in the cluster. Oracle recommends setting this attribute even if clusterCompatible is "false". If not set, the value of the Java system property <code>java.io.tmpdir</code> is used.
<i>maximumUploadedFileSize</i>	Maximum size of uploaded files (in bytes). If less than zero, there is no maximum size. If zero, file upload is disabled.
<i>maximumWindowsPerSession</i>	<p>Maximum number of active windows (or browser tabs, depending on browser configuration) per session. When this limit is reached, the least recently used window expires and subsequent interaction with that window displays an error message.</p> <p>The Oracle WebCenter Content user interface main page, <code>wccmain</code>, does not contribute to the session's window count and only expires when the session expires or is logged out. The document properties page, <code>wccdoc</code>, counts as two windows if <code>documentPreview</code> is "DOCUMENT_VIEWER", but counts as one window if <code>documentPreview</code> is "WEB_VIEWABLE" or "NONE".</p> <p>Use this attribute to control the maximum application server memory consumed by each session. Set to 0 to not limit the number of windows per session.</p>
<i>wccLoginPageEnabled</i>	Specifies whether the "wcclogin" page is enabled or disabled. Either "true" or "false". When Single Sign-On (SSO) authentication is enabled, the attribute value should be set to "false" to prevent users from using the "wcclogin" page instead of SSO.
<i>skinFamily</i>	Name of the skin family. The <code><skin-family></code> element in <code>trinidad-config.xml</code> is set to the value of this attribute.
<i>skinVersion</i>	Skin version. The <code><skin-version></code> element in <code>trinidad-config.xml</code> is set to the value of this attribute. Optional. If not set, no skin version is specified.
<i>customBrandingLogo</i>	HTTP URL of an image to display in the branding bar. Optional. If not set, a default image is displayed.
<i>customBrandingTitle</i>	The title to display in the branding bar. Optional. If not set, a default title is displayed.
<i>wccCustomizationLayerValue</i>	The values for the 'wcc' customization layer. Use comma-separated format to define multiple values. Optional. If not set, no customization is available for this layer. Example: <code>demo, test, or production</code> .

Argument	Definition
<i>oracleCustomizationLayerValues</i>	The values for the 'oracle' customization layer. Use comma-separated format to define multiple values. Optional. If not set, no customization is available for this layer. Example: demo, test, or production.
<i>customerCustomizationLayerValues</i>	The values for the 'customer' customization layer. Use comma-separated format to define multiple values. Optional. If not set, no customization is available for this layer. Example: demo, test, or production.
<i>customRequestBinderProperties</i>	The list of supported custom properties that can be injected to Content Server service calls via ContentProxyServlet requests. Use comma-separated format to define multiple values. Optional. If not set, no injection would happen in Content Server service calls. Example: XFND_SCHEME_ID, XFND_RANDOM, and XFND_EXPIRES.
<i>unfiledDocumentProhibited</i>	Controls the creation of unfiled documents. When its value is true, uploading unfiled documents and unfiled documents are both disabled. Optional. The default value is false.
<i>viewerPagesCountForSinglePageMode</i>	Controls the page navigation mode in the Viewer. If any document contains more than this page count, then the viewer switches to single page navigation mode. The default value and also the recommended value is 400.
<i>disableDownloadForReadOnlyUser</i>	When this parameter is set to true, it disables/removes the Download button (which is used to download a document) for users who have read-only privileges in the ADF interface.

 **Note:**

The value for this parameter should not be greater than 400 due to ADF table limitations.

Examples

The following command sets the proxyContent attribute to "true" for the Oracle WebCenter Content user interface deployed with the application name "Oracle WebCenter Content - Web UI". With this setting, browsers will retrieve document content from the Oracle WebCenter Content user interface, rather than directly from Content Server.

```
updateWccAdfConfig(appName='Oracle WebCenter Content - Web UI', proxyContent=true)
```

The following command sets the defaultLocale attribute to "en_US" and the defaultTimeZone attribute to "America/Los_Angeles". If the Oracle WebCenter Content user interface cannot determine a user's preferred locale and time zone (for example, based on user preferences), it will default to the English language and display times using the America/Los_Angeles time zone.

```
updateWccAdfConfig(appName='Oracle WebCenter Content - Web UI', defaultLocale='en_US', defaultTimeZone='America/Los_Angeles')
```

The following command sets the `temporaryDirectory` attribute to `"/prod/wcctmp"`. The Oracle WebCenter Content user interface will store temporary files in this directory on the application server.

```
updateWccAdfConfig(appName='Oracle WebCenter Content - Web UI',
temporaryDirectory='/prod/wcctmp')
```

The following command sets the `wccLoginPageEnabled` property to `"false"`. This prevents users from authenticating to the Oracle WebCenter Content user interface using its `"wcclogin"` page, requiring them to use Single Sign-On (SSO).

```
updateWccAdfConfig(appName='Oracle WebCenter Content - Web UI',
wccLoginPageEnabled=false)
```

We strongly recommend to set a value for the `applicationUrl` attribute.

```
updateWccAdfConfig(appName='Oracle WebCenter Content - Web UI',
applicationUrl='https://wcc.example.com:16225')
```

The default value for the `maximumWindowsPerSession` attribute is set to 7. If the WebCenter Content instance is experiencing higher than desired memory consumption, the setting can be changed to 4 to reduce the required heap size.

```
updateWccAdfConfig(appName='Oracle WebCenter Content - Web UI',
maximumWindowsPerSession=4)
```

3.12 User Interface Connection Commands

Use the commands in [Table 3-10](#) to manage connections for the Oracle WebCenter Content user interface introduced in WebCenter Content 11g Release 1 (11.1.1.8). For more information about this user interface, see *Getting Started with the WebCenter Content User Interface* in *Using Oracle WebCenter Content*.

To use these custom commands, you must invoke the WLST script from the appropriate Oracle home. Do not use the WLST script in the WebLogic Server home. For the Oracle WebCenter Content user interface commands, the script is located at:

- UNIX: `MW_HOME/oracle_common/common/bin/wlst.sh`
- Windows: `MW_HOME\oracle_common\common\bin\wlst.cmd`

The WLST process must connect to the WebCenter Content UI server before you run the connection commands listed in the following table.

You can connect to the Content UI server instance using the `connect()` command, for example:

```
connect("weblogic", "password", "t3://localhost:16225").
```

Configuration changes made using these WLST commands are only effective after you restart the Managed Server on which the WebCenter Content - Web UI application is deployed.

Table 3-10 WLST WebCenter Content User Interface Connection Commands

Use this command...	To...	Use with WLST...
createRIDCConnection	Create a new RIDC connection.	Online

Table 3-10 (Cont.) WLST WebCenter Content User Interface Connection Commands

Use this command...	To...	Use with WLST...
updateRIDCConnection	Update existing RIDC connection properties.	Online
listRIDCConnections	List all the RIDC connection Mbeans present in an application.	Online
deleteRIDCConnection	Remove an existing RIDC connection.	Online
displayRIDCConnection	Get all the attributes and their values associated with an RIDC connection Mbean with an option to get the value for a given attribute name.	Online

3.12.1 createRIDCConnection

Use with WLST: Online

Description

Creates a new RIDC connection reference and adds the connection name to the mds layer.

Syntax

```
createRIDCConnection('appName', 'conName')
```

Argument	Type	Description
<i>appName</i>	java.lang.String	Name of the application for which the connection will be created.
<i>connName</i>	java.lang.String	Name of the connection which is the placeholder for RIDC connection reference.

Example

The following command creates an RIDC connection 'WccAdfServerConnection' in the application 'Oracle WebCenter Content - Web UI':

```
createRIDCConnection('Oracle WebCenter Content - Web UI', 'WccAdfServerConnection')
```

3.12.2 updateRIDCConnection

Use with WLST: Online

Description

Updates the connection properties for an existing RIDC connection reference. These changes and additions are read as runtime data and saved in the mds layer.

Syntax

```
updateRIDCConnection(appName, connName, ['connUrl'], ['connSockettimeout'],
['connPoolMethod'], ['connPoolSize'], ['connWaitTime'], ['credUsername'],
['credPassword'], ['credAppidkey'], ['credImpersonationAllowed'], ['jaxwsStack'],
['jaxwsPolicy'], ['jaxwsUpsconfigfile'], ['jaxwsSkipstackoptimizations'],
['jaxwsServerInstancename'], ['jaxwsRegisteridentityswitchfilter'],
```

```
['httpLibrary'], ['idcsAlgorithm'], ['idcsKeystoreFile'],
['idcsKeystorePassword'], ['idcsKeystoreAlias'], ['idcsKeystoreAliasPassword'],
['idcsTrustmanagerFile'], ['idcsTrustmanagerPassword'], [sessionPoolSupport],
[sessionPoolAnonymousFallback], [sessionPoolCredentialProviderClass])
```

Argument	Type	Definition
<i>appName</i>	java.lang.String	Name of the application for which the connection has to be updated.
<i>connName</i>	java.lang.String	Name of the connection which will be placeholder for RIDC connection properties.
<i>connURL</i>	java.lang.String	RIDC connection URL property.
<i>connSockettimeout</i>	java.lang.Integer	RIDC connection socket timeout property in seconds.
<i>connPoolMethod</i>	java.lang.String	RIDC connection pool method property.
<i>connPoolSize</i>	java.lang.Integer	RIDC connection pool size property.
<i>connWaitTime</i>	java.lang.Integer	RIDC connection wait time property.
<i>credUsername</i>	java.lang.String	RIDC authorization credential username property.
<i>credPassword</i>	java.lang.String	RIDC authorization credential password property.
<i>credAppidkey</i>	java.lang.String	RIDC authorization credential appid key property.
<i>credImpersonationAllowed</i>	java.lang.Boolean	Credential impersonation allowed property.
<i>jaxwsStack</i>	java.lang.String	JaxWs protocol stack property.
<i>jaxwsPolicy</i>	java.lang.String	JaxWs protocol policy property.
<i>jaxwsJpsconfigfile</i>	java.lang.String	JaxWs protocol Jps config file property.
<i>jaxwsSkipstackoptimizations</i>	java.lang.Boolean	JaxWs protocol skip stack optimizations property.
<i>jaxwsServerInstancename</i>	java.lang.String	JaxWs protocol server instance name property.
<i>jaxwsRegisteridentityswitchfilter</i>	java.lang.Boolean	JaxWs protocol register identity switch filter property.
<i>httpLibrary</i>	java.lang.String	Http protocol library property.
<i>idcsAlgorith</i>	java.lang.String	Secure Intradoc Protocol Algorithm property.
<i>idcsKeystoreFile</i>	java.lang.String	Secure Intradoc Protocol Keystore File property.
<i>idcsKeystorePassword</i>	java.lang.String	Secure Intradoc Protocol Keystore Password property.
<i>idcsKeystorealias</i>	java.lang.String	Secure Intradoc Protocol Keystore Alias property.
<i>idcsKeystoreAliasPassword</i>	java.lang.String	Secure Intradoc Protocol Keystore Alias password property.
<i>idcsTrustmanagerFile</i>	java.lang.String	Secure Intradoc Protocol TrustManager File property.
<i>idcsTrustmanagePassword</i>	java.lang.String	Secure Intradoc Protocol TrustManager Password property.

Argument	Type	Definition
<i>sessionPoolSupport</i>	java.lang.Boolean	RIDC Session Pool Supported property.
<i>sessionPoolAnonymousFallback</i>	java.lang.Boolean	RIDC Session Pool Anonymous Fallback property.
<i>sessionPoolCredentialProviderClass</i>	java.lang.String	RIDC Session Pool Credential Provider Class property.

Example

The following command updates the Intradoc Protocol Connection with a 90 second socket timeout.

```
updateRIDCConnection('Oracle WebCenter Content - Web
UI', 'WccAdfServerConnection',
connUrl='idc://contentserver:4444', credUsername='weblogic', connSocketTimeout=90)
```

3.12.3 listRIDCConnections

Use with WLST: Online

Description

Lists all the RIDC connection Mbeans present in an ADF connection architecture based application.

Syntax

```
listRIDCConnections('appName')
```

Argument	Type	Description
<i>appName</i>	java.lang.String	Name of the application for which the connection mbeans has to be returned.

Example

The following command returns all the connection Mbeans with connection type 'RIDC' present in application 'Oracle WebCenter Content - Web UI'.

```
listRIDCConnections('Oracle WebCenter Content - Web UI')
```

3.12.4 deleteRIDCConnection

Use with WLST: Online

Description

Lists all the RIDC connection Mbeans present in an ADF connection architecture based application.

Syntax

```
deleteRIDCConnection('appName', 'connName')
```

Argument	Type	Description
<i>appName</i>	java.lang.String	Name of the application for which the connection will be removed.
<i>connName</i>	java.lang.String	Name of the connection to be removed for the given application.

Example

The following command removes the connection name 'WccAdfServerConnection' present in application 'Oracle WebCenter Content - Web UI'.

```
deleteRIDCConnection('Oracle WebCenter Content - Web UI','WccAdfServerConnection')
```

3.12.5 displayRIDCConnection

Use with WLST: Online

Description

Gets all the attribute name-value pairs present in a given connection Mbean of an application, with an option to get the value of an attribute present in a given connection Mbean of an application.

Syntax

```
displayRIDCConnection(appName,connName,['attrName'])
```

Argument	Type	Description
<i>appName</i>	java.lang.String	Name of the application for which the connection Mbean attribute value is desired.
<i>connName</i>	java.lang.String	Name of the connection that contains the mbean property.
<i>attrName</i>	java.lang.String	Optional. Attribute name in connection Mbean for which value is desired.

Examples

The following command retrieves the connection property value of 'PropConnectionUrl'.

```
displayRIDCConnection('Oracle WebCenter Content - Web UI',  
'WccAdfServerConnection','PropConnectionUrl')
```

The following command retrieves all the connection properties present in connection 'WccAdfServerConnection'.

```
ddisplayRIDCConnection('Oracle WebCenter Content - Web UI', 'WccAdfServerConnection')
```

4

Oracle WebCenter Content: Imaging Custom WLST Commands

This chapter provides detailed descriptions of custom WLST commands for Oracle WebCenter Content: Imaging, including command syntax, arguments and command examples.

The following sections describe the WLST commands that are specific to Oracle WebCenter Content: Imaging. Topics include:

- [Overview of Imaging WLST Command Categories](#)
- [Diagnostic Commands](#)
- [Imaging Configuration Commands](#)

4.1 Overview of Imaging WLST Command Categories

WLST commands specific to Imaging are divided into the following categories.

Table 4-1 Imaging WLST Command Categories

Command category	Description
Diagnostic Commands	Return workflow agent and other processing information.
Imaging Configuration Commands	Configure settings specific to Imaging and Process Management.

4.2 Diagnostic Commands

Use the Imaging WLST diagnostic commands, listed in table [Table 4-2](#), to list and organize processing failures during workflow processes.

Table 4-2 Diagnostic Commands for Imaging

Use this command...	To...	Use with WLST...
clearIPMWorkflowFaults	Clear processing failures that occurred during workflow agent processing.	Online
clearIPMWorkflowFaultsByDocId	Clear processing failures that have occurred during workflow agent processing for documents entered.	Online
listIPMWorkflowFaults	Provide details of processing failures that occurred during workflow agent processing.	Online
repairIPMWorkflowFaults	Repair processing failures that occurred during workflow agent processing.	Online

Table 4-2 (Cont.) Diagnostic Commands for Imaging

Use this command...	To...	Use with WLST...
repairIPMWorkflowFaultsByDocId	Repair processing failures that have occurred during workflow agent processing for documents entered.	Online
sumIPMWorkflowFaults	Count processing failures during workflow agent processing, grouped by choice of date, application ID, or batch ID.	Online
resetIPMDMSMetrics	Reset DMS metrics to zero.	Online

4.2.1 clearIPMWorkflowFaults

Command Category: Diagnostic Commands

Use with WLST: Online

Description

Clear processing failures that have occurred during workflow agent processing.

Syntax

```
clearIPMWorkflowFaults([startDate], [endDate], [appId], [batchId])
```

Argument	Definition
<i>startDate</i>	Optional. The start of the date range for which error details should be repaired, in yyyy-MM-dd format.
<i>endDate</i>	Optional. The end of the date range for which error details should be repaired, in yyyy-MM-dd format.
<i>appId</i>	Optional. The application ID for which error details should be repaired, in yyyy-MM-dd format.
<i>batchId</i>	Optional. The batch ID for which error details should be repaired.

Example

The following example clears the faults within the specified parameters.

```
clearIPMWorkflowFaults(startDate="2009-06-01", endDate="2009-06-02")
clearIPMWorkflowFaults(appId=3)
clearIPMWorkflowFaults(batchId=15)
clearIPMWorkflowFaults(startDate="2009-06-01", endDate="2009-06-02", appId=3)
```

4.2.2 clearIPMWorkflowFaultsByDocId

Command Category: Diagnostic Commands

Use with WLST: Online

Description

Clear processing failures that have occurred during workflow agent processing for documents entered.

Syntax

```
clearIPMWorkflowFaultsByDocId(docIds=<list of docIds>)
```

Argument	Definition
<i>docIds</i>	Required. The list of document ids for which the processing failures should be cleared.

Example

The following example clears the faults within the specified parameters.

```
clearIPMWorkflowFaultsByDocId(docIds=<list of docIds>)
```

4.2.3 listIPMWorkflowFaults

Command Category: Diagnostic Commands

Use with WLST: Online

Description

List details on processing failures that have occurred during workflow agent processing.

Syntax

```
listIPMWorkflowFaults([startDate], [endDate], [appId], [batchId])
```

Argument	Definition
<i>startDate</i>	Optional. The start of the date range for which error details should be repaired, in yyyy-MM-dd format.
<i>endDate</i>	Optional. The end of the date range for which error details should be repaired, in yyyy-MM-dd format.
<i>appId</i>	Optional. The application ID for which error details should be repaired.
<i>batchId</i>	Optional. The batch ID for which error details should be repaired.

Example

The following example clears the faults within the specified parameters.

```
listIPMWorkflowFaults(startDate="2009-06-01", endDate="2009-06-02")
listIPMWorkflowFaults(appId=3)
listIPMWorkflowFaults(batchId=15)
listIPMWorkflowFaults(startDate="2009-06-01", endDate="2009-06-02", appId=3)
```

4.2.4 repairIPMWorkflowFaults

Command Category: Diagnostic Commands

Use with WLST: Online

Description

Repair processing failures that have occurred during workflow agent processing.

Syntax

```
repairIPMWorkflowFaults([startDate], [endDate], [appId], [batchId])
```

Argument	Definition
<i>startDate</i>	Optional. The start of the date range for which error details should be repaired, in yyyy-MM-dd format.
<i>endDate</i>	Optional. The end of the date range for which error details should be repaired, in yyyy-MM-dd format.
<i>appId</i>	Optional. The application ID for which error details should be repaired.
<i>batchId</i>	Optional. The batch ID for which error details should be repaired.

Example

The following example clears the faults within the specified parameters.

```
repairIPMWorkflowFaults(startDate="2009-06-01", endDate="2009-06-02")
repairIPMWorkflowFaults(appId=3)
repairIPMWorkflowFaults(batchId=15)
repairIPMWorkflowFaults(startDate="2009-06-01", endDate="2009-06-02", appId=3)
```

4.2.5 repairIPMWorkflowFaultsByDocId

Command Category: Diagnostic Commands

Use with WLST: Online

Description

Repair processing failures that have occurred during workflow agent processing for documents entered. This operation may take a long time depending on the number of documents being repaired.

Syntax

```
repairIPMWorkflowFaultsByDocId(docIds=<list of docIds>)
```

Argument	Definition
<i>docIds</i>	Required. The list of document ids for which processing failures should be repaired.

Example

The following example repairs the faults within the specified parameters.

```
repairIPMWorkflowFaultsByDocId(docIds=<list document ids>)
```

4.2.6 sumIPMWorkflowFaults

Command Category: Diagnostic Commands

Use with WLST: Online

Description

Provides a count of processing failures that have occurred during workflow agent processing. The results are grouped by date, application ID, or batch ID.

Syntax

```
sumIPMWorkflowFaults(group)
```

Argument	Definition
<i>groupOption</i>	Required. One of the following: <ul style="list-style-type: none"> DATE: Returns fault counts grouped by date. APPID: Returns fault counts grouped by application ID. BATCHID: Returns fault counts grouped by batch ID.

Example

The following example returns all workflow faults grouped first by date, then by applications ID, then again grouped by batch ID.

```
sumIPMWorkflowFaults(group="DATE")
sumIPMWorkflowFaults(group="APPID")
sumIPMWorkflowFaults(group="BATCHID")
```

4.2.7 resetIPMDMSMetrics

Command Category: Diagnostic Commands

Use with WLST: Online

Description

Resets all Dynamic Monitoring Server (DMS) metrics associated with I/PM to zero. This is generally done if the administrator finds that historical performance data is skewing the current results.

Syntax

```
resetIPMDMSMetrics()
```

Example

The following example resets all DMS metrics to zero.

```
resetIPMDMSMetrics()
```

4.3 Imaging Configuration Commands

Use the Imaging configuration commands, listed in [Table 4-3](#), to list and set configuration values specific to Imaging.

Table 4-3 Configuration Commands for Imaging

Use this command...	To...	Use with WLST...
createIPMConnection	Creates a new Imaging connection from a connection definition file.	Online
modifyIPMConnection	Modifies the Oracle WebCenter Content: Imaging connection from a connection definition file.	Online
getIPMConfig	Get an Imaging configuration setting value, similar to navigating to the custom Imaging config mbean and using the standard WLST <i>set</i> command.	Online
getIPMParamLimits	Lists parameter limits for Oracle WebCenter Content: Imaging MBean parameters.	Online
grantIPMCredAccess	Grants CredentialAccessPermissions to Imaging when Imaging Managed Servers are in a separate domain home from the Administration Server.	Online
importIPMApplication	Imports an application definition from a previously exported definition file.	Online
importIPMInput	Imports an input definition from a previously exported definition file.	Online
importIPMSearch	Imports a search definition from a previously exported definition file.	Online
listIPMConfig	Lists Imaging configuration mbeans.	Online
listIPMDefinitions	List the definitions that are on the system for one of the following types; APPLICATIONS, SEARCHES, INPUTS.	Online
listIPMExportFile	Lists the contents of an exported Imaging definitions file.	Online
refreshIPMSecurity	Refresh security items currently stored in the Imaging database.	Online
setIPMConfig	Sets an Imaging configuration value.	Online
submitIPMToWorkflow	Submits a document to the workflow agent.	Online

4.3.1 createIPMConnection

Command Category: Imaging Configuration Commands

Use with WLST: Online

Description

Creates a new Imaging connection from a connection definition file. The connection definition file is an XML file that describes a single Imaging connection definition using the Connection element type from the Imaging ConnectionService web services API schema definition. This schema is available from a running Imaging server using at the following URL:

`http://ipm_server_machine:ipm_server_port/imaging/ws/ConnectionService?xsd=1`

For more information about the connection definition file format, see Managing Connections in *Administering Oracle WebCenter Content: Imaging*.

Syntax

```
createIPMConnection(connectionFile)
```

Argument	Definition
<i>connectionFile</i>	Required. A full path to the connection definition file's location on the Imaging server Node. Must be enclosed in single or double quotes.

Example

The following example creates a connection based on the specified attribute.

```
createIPMConnection(connectionFile="/home/ipmuser/localCSConnection.xml")
```

4.3.2 modifyIPMConnection

Command Category: Imaging Configuration Commands

Use with WLST: Online

Description

Modifies the Oracle WebCenter Content: Imaging connection from a connection definition file.

The connection definition file is an XML file that describes a single Oracle WebCenter Content: Imaging connection definition using the Connection element type from the Oracle WebCenter Content: Imaging ConnectionService web services API schema definition. This schema is available from a running Oracle WebCenter Content: Imaging server using the following url:

```
http://ipm_server_machine:ipm_server_port/imaging/ws/ConnectionService?xsd=1
```

For more information about the connection definition file format, see Managing Connections in *Administering Oracle WebCenter Content: Imaging*.

Syntax

```
modifyIPMConnection(connectionFile)
```

Argument	Definition
<i>connectionFile</i>	Required. A full path to the connection definition file's location on the Imaging server node. Must be enclosed in single or double quotes

Example

The following example modifies a connection based on the specified attribute.

```
modifyIPMConnection(connectionFile="/home/ipmuser/localCSConnection.xml")
```

4.3.3 getIPMConfig

Command Category: Imaging Configuration Commands

Use with WLST: Online

Description

Gets an Imaging configuration setting value. The command is equivalent to browsing the custom mbean hierarchy to the Imaging config mbean and using the standard WLST *set* command to set an mbean attribute.

Syntax

```
getIPMConfig(attrName)
```

Argument	Definition
<i>attrName</i>	Required. Name of the attribute to be read. Must be enclosed in single or double quotes.

Example

The following example returns the value for the specified attribute names.

```
getIPMConfig('AgentUser')getIPMConfig('CheckInterval')
```

4.3.4 getIPMParamLimits

Command Category: Imaging Configuration Commands

Use with WLST: Online

Description

Lists parameter limits for Oracle WebCenter Content: Imaging MBean parameters. Returns an array with the limits applicable for an Imaging Mbean parameter.

Syntax

```
getIPMParamLimits(paramName="<MBean parameter name>")
```

Argument	Definition
<i>paramName</i>	Optional. The MBean parameter name to return the limits for. If not set, will return all valid parameter names.

Example

The following examples return the limits for the specified attribute names.

```
getIPMParamLimits("UIMaxSearchResults")
getIPMParamLimits("InputAgentCheckInterval")
getIPMParamLimits()
```

4.3.5 grantIPMCredAccess

Command Category: Imaging Configuration Commands

Use with WLST: Online

Description

Grants `CredentialAccessPermissions` to Imaging so that it can read credentials from the credential store. This command is required in configurations where Imaging managed servers are in a separate domain home from the Administration Server. When at least one Imaging managed server is in the same domain home as the Administration Server, this command is not required, as `CredentialAccessPermissions` are granted during Imaging startup.

When the Imaging Managed Server is not in the same domain home as the Administration Server, however, the Imaging startup grant only affects the local settings. Local settings get overwritten when the Administration Server synchronizes its copy as the domain wide configuration, so this command updates the Administration Server configuration such that permissions are distributed correctly to all domain nodes.

Syntax

```
grantIPMCredAccess()
```

Example

The following example returns a list of all Imaging configuration mbeans.

```
grantIPMCredAccess()
```

4.3.6 importIPMApplication

Command Category: Imaging Configuration Commands

Use with WLST: Online

Description

Imports an application definition from a previously exported definition file.

Syntax

```
importIPMApplication(exportFile, action, name, repository, securityOption,
securityMember, docSecurityOption, docSecurityGroup, storageOption, storageVolume)
```

Argument	Definition
<i>exportFile</i>	Required. A full path to the export definition file's location on the Imaging server node. Must be enclosed in single or double quotes.
<i>action</i>	Required. The action to be performed. Available actions are: <ul style="list-style-type: none"> • Add: Creates a new input. Fails if an application with the same name already exists. • Update: Modifies and existing input. Fails if an application with the same name does not exist. • AddOrUpdate: Creates a new application if it does not already exist or updates one that does.
<i>name</i>	Required. The name of the application being imported from the exported definitions file.
<i>repository</i>	The name of the repository in which to create the application. Required when adding an application, ignored when updating or modifying an application.

Argument	Definition
<i>securityOption</i>	<p>Optional. Specifies how to define security for the imported application as follows:</p> <ul style="list-style-type: none"> • Existing: Uses application security as defined in the existing definition. Valid only for an update action. • Imported: Attempts to use application security as defined in the import file. Fails if any members defined in the import file are invalid. • ValidOnly: Uses application security as defined in the import file and filters out any invalid members. • CurrentUser: Sets full permissions to the user used to connect to the server. • User: Sets full permissions to the user name provided in the securityMember parameter. • Group: Sets full permissions to the group name provided in the securityMember parameter.
<i>securityMember</i>	<p>Name of the user or group given full permissions to the application. Valid only when securityOption is set to either <i>User</i> or <i>Group</i>, otherwise it is ignored.</p>
<i>docSecurityOption</i>	<p>Optional. Specifies how to define document security for the imported application.</p> <ul style="list-style-type: none"> • Existing: Uses document security as defined in the existing application. Valid only for an update action. • Imported: Attempts to use document security as defined in the import file. Fails if any members defined in the import file are invalid. • ValidOnly: Uses document security as defined in the import file and filters out any invalid members. • Group: Sets full permissions to the group name provided in the docSecurityGroup parameter.
<i>docSecurityGroup</i>	<p>Name of group given full permissions to document security. Valid only when docSecurityOption is set to <i>Group</i>, otherwise it is ignored.</p>
<i>storageOption</i>	<p>Optional. Specifies how to define the storage policy for the imported application.</p> <ul style="list-style-type: none"> • Existing: Uses the document storage policy as defined in the existing application. Valid only for an update action. • Imported: Attempts to use storage policy as defined in the import file. • Volume: Uses the specific volume named in the storageVolume parameter. • Default: Sets up the storage policy to use the system default volume.
<i>storageVolume</i>	<p>Required. Volume for setting storage policy. Valid only when a storageOption of <i>Volume</i> is used. Ignored otherwise.</p>

Example 1

The following example updates an existing application named *Invoices*. Note that the repository is listed as **None** because the update action uses the repository specified in the original application.

```
importIPMApplication(exportFile="/home/ipmuser/exportdefinitions.xml",
action="Update", name="Invoices", repository=None, securityOption="Existing")
```

Example 2

The following example creates a new application named Receipts. Note that the repository is explicitly named because the add action requires a valid repository be named.

```
importIPMApplication(exportFile="/home/ipmuser/exportdefinitions.xml", action="Add",
name="Receipts", repository="LocalCS", securityOption="ValidOnly")
```

4.3.7 importIPMInput

Command Category: Imaging Configuration Commands

Use with WLST: Online

Description

Imports an input definition from a previously exported definition file.

Syntax

```
importIPMInput(exportFile, action, name, securityOption, securityMember)
```

Argument	Definition
<i>exportFile</i>	Required. A full path to the export definition file's location on the Imaging server node. Must be enclosed in single or double quotes.
<i>action</i>	Required. The action to be performed. Available actions are: <ul style="list-style-type: none"> • Add: Creates a new input. Fails if an input with the same name already exists. • Update: Modifies an existing input. Fails if an input with the same name does not exist. • AddOrUpdate: Creates a new application if it does not already exist or updates one that does.
<i>name</i>	Required. The name of the input being imported from the exported definitions file.
<i>repository</i>	The name of the repository in which to create the application. Required when adding an application, ignored when updating or modifying an application.
<i>securityOption</i>	Optional. Specifies how to define security for the imported application as follows: <ul style="list-style-type: none"> • Existing: Uses input security as defined in the existing definition. Valid only for an update action. • Imported: Attempts to use input security as defined in the import file. Fails if any members defined in the import file are invalid. • ValidOnly: Uses input security as defined in the import file and filters out any invalid members. • CurrentUser: Sets full permissions to the user used to connect to the server. • User: Sets full permissions to the user name provided in the securityMember parameter. • Group: Sets full permissions to the group name provided in the securityMember parameter.

Argument	Definition
<i>securityMember</i>	Name of the user or group given full permissions to the input. Valid only when <i>securityOption</i> is set to either <i>User</i> or <i>Group</i> , otherwise it is ignored.

Example 1

The following example updates an existing input named *Invoices*. Note that the repository is listed as **None** because the update action uses the repository specified in the original application.

```
importIPMInput (exportFile="/home/ipmuser/exportdefinitions.xml",
action="Update", name="Invoices", securityOption="Existing")
```

Example 2

The following example creates a new input named *Receipts*. Note that the repository is explicitly named because the add action requires a valid repository be named.

```
importIPMInput (exportFile="/home/ipmuser/exportdefinitions.xml", action="Add",
name="Receipts", securityOption="ValidOnly")
```

4.3.8 importIPMSearch

Command Category: Imaging Configuration Commands

Description

Import a search definition from a previously exported definition file.

Syntax

```
importIPMSearch(exportFile, action, name, securityOption, securityMember)
```

Argument	Definition
<i>exportFile</i>	Required. A full path to the export definition file's location on the Imaging server node. Must be enclosed in single or double quotes.
<i>action</i>	Required. The action to be performed. Available actions are: <ul style="list-style-type: none"> • Add: Creates a new search. Fails if a search with the same name already exists. • Update: Modifies an existing search. Fails if a search with the same name does not exist. • AddOrUpdate: Creates a new search if it does not already exist or updates one that does.
<i>name</i>	Required. The name of the search being imported from the exported definitions file.
<i>repository</i>	The name of the repository in which to create the application. Required when adding an application, ignored when updating or modifying an application.

Argument	Definition
<i>securityOption</i>	Optional. Specifies how to define security for the imported application as follows: <ul style="list-style-type: none"> • Existing: Uses search security as defined in the existing definition. Valid only for an update action. • Imported: Attempts to use search security as defined in the import file. Fails if any members defined in the import file are invalid. • ValidOnly: Uses search security as defined in the import file and filters out any invalid members. • CurrentUser: Sets full permissions to the user used to connect to the server. • User: Sets full permissions to the user name provided in the securityMember parameter. • Group: Sets full permissions to the group name provided in the securityMember parameter.
<i>securityMember</i>	Name of the user or group given full permissions to the search. Valid only when securityOption is set to either <i>User</i> or <i>Group</i> , otherwise it is ignored.

Example 1

The following example updates an existing search named *Invoices*. Note that the repository is listed as **None** because the update action uses the repository specified in the original application.

```
importIPMSearch(exportFile="/home/ipmuser/exportdefinitions.xml", action="Update",
name="Invoices", securityOption="Existing")
```

Example 2

The following example creates a new search named *Receipts*. Note that the repository is explicitly named because the add action requires a valid repository be named.

```
importIPMSearch(exportFile="/home/ipmuser/exportdefinitions.xml", action="Add",
name="Receipts", securityOption="ValidOnly")
```

4.3.9 listIPMConfig

Command Category: Imaging Configuration Commands

Use with WLST: Online

Description

Provides a listing of Imaging configuration mbeans. The command is equivalent to browsing the custom mbean hierarchy and listing the Imaging mbean attributes.

Syntax

```
listIPMConfig()
```

Example

The following example returns a list of all Imaging configuration mbeans.

```
listIPMConfig()
```

4.3.10 listIPMDefinitions

Command Category: Imaging Configuration Commands

Use with WLST: Online

Description

Lists the definitions that are on the system for one of the following types: APPLICATIONS, SEARCHES, INPUTS.

Syntax

```
listIPMDefinitions(defType="<definition type>")
```

Argument	Definition
<i>defType</i>	Required. One of APPLICATIONS, SEARCHES, INPUTS. This argument is case insensitive. <ul style="list-style-type: none"> APPLICATIONS will return the list of applications residing on the system. SEARCHES will return the list of searches residing on the system. INPUTS will return the list of inputs residing on the system

Example

The following examples return the list of inputs and applications residing on the system.

```
listIPMDefinitions(defType="INPUTS")
listIPMDefinitions(defType="APPLICATIONS")
```

4.3.11 listIPMExportFile

Command Category: Imaging Configuration Commands

Use with WLST: Online

Description

Lists the contents of an exported Imaging definitions file.

Syntax

```
listIPMExportFile(exportFile="<path to file>")
```

Argument	Definition
<i>exportFile</i>	Required. A full path to the export definition file's location on the Imaging server node. Must be enclosed in single or double quotes.

Example

The following example returns the contents of an Imaging definitions file.

```
listIPMExportFile(exportFile="/home/ipmuser/exportdefinitions.xml")
```

4.3.12 refreshIPMSecurity

Command Category: Imaging Configuration Commands

Use with WLST: Online

Description

Refreshes security items currently stored in the Imaging database. This is typically done when migrating security to a different policy store and only updates security items found in the new policy store.

Syntax

```
refreshIPMSecurity()
```

Example

The following example refreshes the security items stored in the Imaging database.

```
refreshIPMSecurity()
```

4.3.13 setIPMConfig

Command Category: Imaging Configuration Commands

Use with WLST: Online

Description

Sets an Imaging configuration setting value. The command is equivalent to browsing the custom mbean hierarchy to the Imaging config mbean and using the standard WLST 'set' command to set an mbean attribute.

Syntax

```
setIPMConfig(attrName, value)
```

Argument	Definition
<i>attrName</i>	Required. Name of the attribute to be set. Must be enclosed in single or double quotes.
<i>value</i>	Required. Value of the attribute to be set. Only enclosed in single or double quotes if value is a string literal.

Example

The following example sets the specified values for the specified attribute names.

```
setIPMConfig('AgentUser', 'agentadmin')setIPMConfig('CheckInterval', 30)
```

4.3.14 submitIPMToWorkflow

Command Category: Imaging Configuration Commands

Use with WLST: Online

Description

Submits a document to the workflow agent. Note that a confirmation message is displayed stating that the document has been submitted, however if the document is stored in an application that is not configured with a workflow, no action is taken.

Syntax

```
submitIPMToWorkflow(documentId)
```

Argument	Definition
<i>documentId</i>	Required. The unique document ID of the submitted document.

Example

The following example submits a document to a workflow.

```
submitIPMToWorkflow(documentId="2.IPM_12345" )
```

5

Oracle WebCenter Enterprise Capture Custom WLST Commands

This chapter provides detailed descriptions of custom WLST commands for Oracle WebCenter Enterprise Capture, including command syntax, arguments and command examples.

The following sections describe the custom WLST commands for Oracle WebCenter Enterprise Capture. These commands enable you to access and modify various configuration parameters for Oracle WebCenter Enterprise Capture from the command line. Topics include:

- [Overview of WLST Oracle WebCenter Enterprise Capture Command Categories](#)
- [Configuration Commands](#)

To display all the supported Capture commands that can be invoked from WLST, enter the following command at the WLST prompt:

```
help('capture')
```

For information about Enterprise Capture workspace console use, see *About the Capture Workspace Console* in *Managing Oracle WebCenter Enterprise Capture*.

Note:

To use the Oracle WebCenter Enterprise Capture custom commands, you must invoke the WLST script from the Oracle Common home in which the component has been installed. Either connect to the Oracle WebLogic Server managed server (default port 16400) on which the Oracle WebCenter Enterprise Capture application is deployed, or connect to the Oracle WebLogic Server administration server and change your location to domainRuntime by executing the command:

```
domainRuntime()
```

5.1 Overview of WLST Oracle WebCenter Enterprise Capture Command Categories

WLST Oracle WebCenter Enterprise Capture commands are covered in the following category:

Table 5-1 WLST Oracle WebCenter Enterprise Capture Command Categories

Command Category	Description
Configuration Commands	View and manage configuration for Oracle WebCenter Enterprise Capture.

5.2 Configuration Commands

Use the commands in [Table 5-2](#) to configure Oracle WebCenter Enterprise Capture.

Table 5-2 WLST Server Oracle WebCenter Enterprise Capture Configuration Commands

Use this command...	To...	Use with WLST...
listWorkspaces	List all of the Capture workspaces to which the administrator or workspace manager has access.	Online
listBatches	List all of the batches within a specified workspace.	Online
exportBatch	Export a batch including images and metadata to a ZIP file.	Online
exportWorkspace	Export an entire Capture workspace including documents, metadata, scripts, profiles and jobs to an XML file.	Online
exportEWSEmailMessage	This email export method is specific to Microsoft Email Exchange Service. It exports the EWS email message with specified <code>messageId</code> to the specified file. Export File name should be specified along with its complete path and must have <code>.eml</code> extension.	Online
exportIMAPEmailMessage	Exports the IMAP email message with specified <code>messageId</code> to the specified file. Export File name should be specified along with its complete path and must have <code>.eml</code> extension. <code>EmailConnectionSecurity</code> must be an integer, where 0 is for No Security, 1 is for security type <code>SSL_TLS</code> and 2 is for security type <code>STARTTLS</code> . All parameters should be in String format other than <code>EmailConnectionSecurity</code> and <code>Port</code> .	Online
importWorkspace	Import a workspace from an XML file generated with the <code>exportWorkspace</code> command	Online
unlockBatch	Delete the batch lock record for the specified batch.	Online
listLockedBatches	Display a list of locked batches for a specified workspace.	Online
listCaptureConfig	Display the list of all Enterprise Capture configuration attributes with their values.	Online
getCaptureConfig	Fetch the value of the attribute specified as an argument.	Online
setCaptureConfig	Set the value of the Enterprise Capture attribute to the provided value.	Online

Table 5-2 (Cont.) WLST Server Oracle WebCenter Enterprise Capture Configuration Commands

Use this command...	To...	Use with WLST...
scanForClientBundles	Scan the client bundle directory for updated bundles.	Online
deleteBundle	Delete an existing client bundle.	Online
deleteBatches	Deletes the batches with specified IDs.	Online
setObjectProperty	Set the value of a property defined in a Capture object which could be a client profile, processor job, or a commit profile.	Online
getObjectProperty	Print the specific value of a property of a Capture object which could be a client profile, processor job, or a commit profile.	Online
setObjectCredentials	Set the user name and password associated with a Capture object (processor job or commit profile).	Online
updateScript	Update a script within a workspace.	Online
cloneWorkspaceFromID	Copy an existing workspace and create a new workspace.	Online
cloneWorkspaceFromFile	Read an output file created by the <code>exportWorkspace</code> command and create a unique copy of the workspace.	Online
listImportHATokens	Display all the Import Processor tokens for a specific workspace.	Online
deleteImportHAToken	Delete a specific Import Processor HA Token from the <code>ECIMPORTHATOKEN</code> and <code>ECIMPORTHATOKENITEMS</code> tables.	Online

5.2.1 listWorkspaces

Use with WLST: Online.

Description

Lists all Capture workspaces to which the administrator or workspace manager has access.

Syntax

```
listWorkspaces()
```

Example

The following example lists all of the workspaces, preceded by their IDs.

```
listWorkspaces()
4 bills_workspace
2 certificates_workspace
5 receipts_workspace
```

5.2.2 listBatches

Use with WLST: Online.

Description

Lists all batches for the workspace specified by its ID. Each batch listed has five columns (displayed in this order): Batch ID, Batch Name, Created By (user), Last Updated By (user), and Last Modified (date).

Syntax

```
listBatches (WORKSPACE_ID)
```

Argument	Definition
<i>WORKSPACE_ID</i>	ID number for a workspace. Can be optionally specified within single quotes. For example, both 3 and '3' are valid arguments for a workspace with ID 3.



Note:

To obtain a list of valid workspace IDs, use the `listWorkspaces` command.

Example

The following example lists the batches for a specific workspace with ID = 1.

```
wls:/base_domain/domainRuntime> listBatches(1)
14 ABI1 <anonymous> <anonymous> 4/26/13
15 ABI2 <anonymous> <anonymous> 4/26/13
1 BCP11 captureuser <anonymous> 4/17/13
```

5.2.3 exportBatch

Use with WLST: Online.

Description

Exports a batch including images and metadata to a ZIP file containing an XML file and the batch item files.

Syntax

```
exportBatch(BATCH_ID, 'FILE_NAME')
```

Argument	Definition
<i>BATCH_ID</i>	ID number for a batch. Can be optionally specified within single quotes. For example, both 3 and '3' are valid arguments for a batch with ID 3.
<i>FILE_NAME</i>	Name of the ZIP file to which the batch will be exported. The file name must be specified along with its complete path.

 **Note:**

To obtain a list of valid batch IDs, use the `listBatches` command.

 **Note:**

The path should be separated by forward slashes "/" regardless of the operating system.

Example

The following example exports a batch.

```
exportBatch(8, '/home/abc/batch8.zip')  
Batch successfully exported
```

5.2.4 exportWorkspace

Use with WLST: Online.

Description

Exports an entire workspace including documents, metadata, scripts, profiles and jobs to an XML file.

 **Note:**

Workspace security settings will not be exported into the resulting XML file.

Syntax

```
exportWorkspace(WORKSPACE_ID, 'FILE_NAME')
```

Argument	Definition
<i>WORKSPACE_ID</i>	ID number for a workspace. Can be optionally specified within single quotes. For example, both 3 and '3' are valid arguments for a workspace with ID 3.

 **Note:**

To obtain a list of valid workspace IDs, use the `listWorkspaces` command.

<i>FILE_NAME</i>	Name of the file to which the workspace will be exported. The file name must be specified along with its complete path.
------------------	---

 **Note:**

The path should be separated by forward slashes "/" regardless of the operating system.

Example

The following example exports the workspace with the ID of 3.

```
exportWorkspace(3, '/home/abc/workspace3.xml')
Workspace successfully exported
```

5.2.5 exportEWSEmailMessage

Use with WLST: Online.

Description

Exports the EWS email message with specified `messageId` to the specified file.

Syntax

```
exportEWSEmailMessage('<MICROSOFT_EMAIL_EXCHANGE_SERVICE_URL>', '<EMAIL_ADDRESS>',
'<PASSWORD>', '<MESSAGE_ID>', '<SEARCH_FOLDER>',
'<EXPORT_FILE_NAME>')
```

Argument	Definition
<i>MICROSOFT_EMAIL_EXCHANGE_SERVICE_URL</i>	Microsoft Email Exchange Service URL
<i>Email_ADDRESS</i>	Email accounts to be checked for message
<i>PASSWORD</i>	Password for email account
<i>MESSAGE_ID</i>	Unique Id of the message
<i>SEARCH_FOLDER</i>	Folder to be checked for message
<i>EXPORT_FILE_NAME</i>	Export File name along with its complete path, must have .eml extension

Example

The following example exports message with id '12abc-3456-7ddd-e891-b1234567' to location '/home/abc/' with file name 'Exported_Email_Message_.eml'.

```
exportEWSEmailMessage('https://<hostname>/ews/
exchange.asmx', 'abc.xyz@oracle.com', 'myEmailPassword', '12abc-3456-7ddd-e891-b1234567',
'inbox', '/home/abc/Exported_Email_Message_.eml')
```

Email Message successfully exported.

5.2.6 exportIMAPEmailMessage

Use with WLST: Online.

Description

Exports the IMAP email message with specified `messageId` to the specified file. Export File name should be specified along with its complete path and must have `.eml` extension. `EmailConnectionSecurity` must be an integer, where 0 is for No Security, 1 is for security type `SSL_TLS` and 2 is for security type `STARTTLS`. All parameters should be in String format except `EmailConnectionSecurity` and `Port`.

Syntax

```
exportIMAPEmailMessage('<EMAIL_SERVER_NAME>', <PORT>, <IMAP_CONNECTION_SECURITY>,
'<Email_ADDRESS>', '<PASSWORD>', '<MESSAGE_ID>', '<SEARCH_FOLDER>',
'<EXPORT_FILE_NAME>')
```

Argument	Definition
<i>EMAIL_SERVER_NAME</i>	Email Server Name (DNS name or IP address)
<i>Port</i>	Port number
<i>IMAP_CONNECTION_SECURITY</i>	0 for Security - None, 1 for Security - SSL/TLS, 2 for Security - STARTTLS
<i>Email_ADDRESS</i>	Email accounts to be checked for message
<i>PASSWORD</i>	Password for email account
<i>MESSAGE_ID</i>	Unique Id of the message
<i>SEARCH_FOLDER</i>	Folder to be checked for message
<i>EXPORT_FILE_NAME</i>	Export File name along with its complete path, must have <code>.eml</code> extension.

Example

The following example exports message with id '12abc-3456-7ddd-e891-b1234567' to location '/home/abc/' with file name 'Exported_Email_Message_.eml'.

```
exportIMAPEmailMessage('stbeehive.oracle.com', 993, 1, 'abc.xyz@oracle.com', 'myEmailPassw
rd', '12abc-3456-7ddd-e891-b1234567', 'inbox', '/hom/abc/Exported_Email_Message_.eml')
```

Email Message successfully exported.

5.2.7 importWorkspace

Use with WLST: Online.

Description

Imports a workspace from a XML file generated with the `exportWorkspace` command. The workspace being imported must not exist in the Oracle WebCenter Enterprise Capture database.



Note:

The security settings are not imported when you import a workspace.

Syntax

```
importWorkspace('FILE_NAME')
```

Argument	Definition
<i>FILE_NAME</i>	Name of the XML file from which a workspace is imported. The file name must be specified along with its complete path.

Example

The following example imports a workspace with the file name of `workspace3.xml`.

```
importWorkspace('/home/abc/workspace3.xml')
Workspace successfully imported
```

5.2.8 unlockBatch

Use with WLST: Online.

Description

Deletes the batch lock record for the specified batch. The batch will be put into a READY state so that it can be opened by the client.

Syntax

```
unlockBatch('BATCH_ID')
```

Argument	Definition
<i>BATCH_ID</i>	ID number for a batch. Can be optionally specified within single quotes. For example, both 3 and '3' are valid arguments for a batch with ID 3.



Note:

To obtain a list of locked batches, use the `listLockedBatches` command.

Example

The following example unlocks a batch with ID = 8.

```
unlockBatch('8')
Batch successfully unlocked
```

5.2.9 listLockedBatches

Use with WLST: Online.

Description

Displays the list of locked batches. Each batch listed has six columns (displayed in this order): Workspace ID, Batch ID, Batch Name, Created By (user), Workstation (name/IP address), Last Modified (date).

Syntax

```
listLockedBatches('WORKSPACE_ID')
```

Argument	Definition
<i>WORKSPACE_ID</i>	ID number for a workspace. Can be optionally specified within single quotes. For example, both 3 and '3' are valid arguments for a workspace with ID 3.

Note:

To obtain a list of valid workspace IDs, use the `listWorkspaces` command.

Example

The following example lists locked batches for the workspace with ID = 1.

```
wls:/base_domain/domainRuntime> listLockedBatches(1)
1 14 ABI1 <anonymous> sic01lzz.example.com 4/26/13
1 15 ABI2 <anonymous> slc01lzz.example.xom 4/26/13
1 17 BCP16 <captureuser> 10.159.104.189 4/30/13
```

5.2.10 listCaptureConfig

Use with WLST: Online.

Description

Displays the list of all Enterprise Capture configuration attributes with their values. If a value cannot be fetched, it will be shown as "VALUE CANNOT BE DISPLAYED". The most likely cause of an unfetched value is insufficient privilege.

Syntax

```
listCaptureConfig()
```

Example

The following example lists Enterprise Capture configuration attributes and values.

```
listCaptureConfig()
CaptureSystemID CAPTURE_01
BatchRefreshLimit 5
```

5.2.11 getCaptureConfig

Use with WLST: Online.

Description

Fetches the value of the attribute specified as an argument. Use the `listCaptureConfig` command to list possible attribute names.

Syntax

```
getCaptureConfig(['NAME_OF_ATTRIBUTE'])
```

Argument	Definition
<i>NAME_OF_ATTRIBUTE</i>	Name of attribute (as a string) for Enterprise Capture configuration.

Example

The following example fetches the value of the attribute for the Enterprise Capture system ID.

```
getCaptureConfig('CaptureSystemID')  
CAPTURE_01
```

5.2.12 setCaptureConfig

Use with WLST: Online.

Description

Sets the value of the Enterprise Capture attribute to the provided value.

Syntax

```
setCaptureConfig(['NAME_OF_ATTRIBUTE'], ['VALUE_OF_ATTRIBUTE'])
```

Argument	Definition
<i>NAME_OF_ATTRIBUTE</i>	Name of Enterprise Capture configuration attribute to configure. <i>E</i>
<i>VALUE_OF_ATTRIBUTE</i>	Value to set for the Enterprise Capture configuration attribute. <i>TE</i>

Example

The following example sets the `CaptureSystemID` attribute to the value `CAPTURE_02`.

```
setCaptureConfig('CaptureSystemID', 'CAPTURE_02')  
Attribute 'CaptureSystemID' changed to "CAPTURE_02"
```

5.2.13 scanForClientBundles

Use with WLST: Online.

Description

Scans the client bundle directory for bundles. Data for existing bundles will be updated and information for new bundles will be loaded and cached in the database.

Syntax

```
scanForClientBundles()
```

Example

The following example scans for bundles and lists processed bundles.

```
scanForclientBundles()
Processed capture bundle.jar
```

5.2.14 deleteBatches

Use with WLST: Online.

Description

Deletes the batches with specified IDs.

Syntax

```
deleteBatches(<BATCH_ID_1>,<BATCH_ID_2>,...<BATCH_ID_N>)
```

Argument	Definition
<i>BATCH_ID</i>	Id number for a batch. Should be only in integer format. For example, 1

Example

The following example deletes batch with ID 3 and 7.

```
deleteBatches(3,7)
Batch(es) successfully deleted.
```

5.2.15 deleteBundle

Use with WLST: Online.

Description

Deletes an existing bundle.

Syntax

```
deleteBundle('bundlename')
```

Argument	Definition
<i>bundlename</i>	Name of an existing bundle to delete.

Example

The following example deletes the specified bundle.

```
deleteBundle('Example')
```

5.2.16 setObjectProperty

Use with WLST: Online.

Description

Sets the value of a property defined in a Capture Object, which could be a client profile, processor job, or a commit profile.

For information on the properties that can be set for each Capture object, see [Capture Object Properties](#).

Syntax

```
setObjectProperty('<WORKSPACE_NAME>', '<CLASS_NAME>', '<OBJECT_IDENTIFIER>', '<PROPERTY_ASSIGNMENT_STRING>')
```

This command requires the following parameters:

Argument	Definition
<i>WORKSPACE_NAME</i>	The name of the Capture workspace that contains the objects that will be modified. If the calling user has rights to more than one workspace with the specified name, the first workspace found will be used.
<i>CLASS_NAME</i>	The name of the class that defines the client profile, processor job, or commit profile.
<i>OBJECT_IDENTIFIER</i>	A key/value pair consisting of the name of a property that can be used to locate the object and the expected value of that property.
<i>PROPERTY_ASSIGNMENT_STRING</i>	The name and value of the property to update, separated by an equal sign.

Note:

To obtain a list of workspaces, use the `listWorkspaces` command.

Points to Note

The following list provides some additional points to note when you use the `setObjectProperty` command:

- Modifiable objects expose their properties using the standard JavaBean getter/setter pattern. In the example below it is assumed the methods `getProfileDesc` and `setProfileDesc` exist in the Profile class.
- The value being assigned to the property should be compatible with the property's data type. If a value is not compatible with the property's data type, an exception will occur.
- The following are valid class names:
 - `oracle.oddc.data.Profile` (Capture object being modified: Client Profile)
 - `oracle.oddc.recognition.RecognitionJob` (Capture object being modified: Recognition Processor Job)

- `oracle.odc.importprocessor.ImportJob` (Capture object being modified: Import Processor Job)
- `oracle.odc.entity.CommitProfileEntity` (Capture object being modified: Commit Profile)
- `oracle.odc.docconverter.DocConverterJob` (Capture object being modified: Document Conversion Job)
- In cases where a value needs to be set on an object nested within a profile or job, “dot” notation can be used to access the nested property: `propertyName.propertyName=value`.
- Multiple levels of nesting are supported. The following could be used to modify the `barcodeName` property of a Recognition Processor job's `docTypeBarcode` property: `docTypeBarcode.barcodeName=value`.
- To support accessing a property of an object contained in a collection, the following “angle bracket” notation can be used: `propertyName<identifierPropertyName=value>.propertyName=value`.
 - The value between the angle brackets will be used to locate the object to modify within the collection. The `identifierPropertyName` is the name of a property that can be used to identify the object in the collection and `value` is the value that property should contain to be considered a match. For example: `barcodes<barcodeName=barcode 1>.barcodeName=barcode 2`.

Example

- The following is an example of `<Object Identifier>` that would apply to commit profiles:

```
profileName=Commit to CS
```

- The following is an example of `<Property Assignment String>` for updating the property `profileDesc` for a client profile:

```
profileDesc=Test Profile Desc
```

- The following is an example of setting the property `<serverURL>` associated with a WebCenter Content commit profile named `Commit to CS`:

```
setObjectProperty('TEST_WORKSPACE', 'oracle.odc.entity.CommitProfileEntity', 'profileName=Commit to CS', 'serverURL=http://myhost.example.com:16200/cs/idcplg/')
```

5.2.17 getObjectProperty

Use with WLST: Online.

Description

Prints the value of a specific property of a Capture Object (client profile, processor job, or commit profile).

For information on the properties that can be printed for each Capture object, see [Capture Object Properties](#).

Syntax

```
getObjectProperty('<WORKSPACE_NAME>', '<CLASS_NAME>', '<OBJECT_IDENTIFIER>', '<PROPERTY_NAME>')
```

This method requires the following parameters:

Argument	Definition
<i>WORKSPACE_NAME</i>	The name of the Capture workspace that contains the object that will be modified. If the calling user has rights to more than one workspace with the specified name, the first workspace found will be used.
<i>CLASS_NAME</i>	The name of the class that defines the client profile, processor job or commit profile.
<i>OBJECT_IDENTIFIER</i>	A key/value pair consisting of the name of a property that can be used to locate the object and the expected value of that property.
<i>PROPERTY_NAME</i>	The name of the property to access. The "dot" notation and "angle bracket" notation described in <code>setObjectProperty</code> can be used to isolate the property to access.

 **Note:**

To obtain a list of workspaces, use the `listWorkspaces` command.

 **Note:**

The backing MBean method for this command can be used in automated tests to validate if the changes made with the `setObjectProperty` command have taken effect.

Example

- The following are valid class names:

```
oracle.oddc.data.Profile
oracle.odc.recognition.RecognitionJob
oracle.odc.importprocessor.ImportJob
oracle.odc.entity.CommitProfileEntity
oracle.odc.docconverter.DocConverterJob
```

- The following is an example of `<Object Identifier>` that would apply to import jobs:

```
jobName=Import Email
```

- The following is an example for retrieving the value of the property `<serverName>` associated with an email import job.

```
getObjectProperty('TEST-
WORKSPACE','oracle.odc.importprocessor.ImportJob','jobName=Import
Email','serverName')
```

5.2.18 setObjectCredentials

Use with WLST: Online.

Description

It is used to set the user name and password associated with a profile or job.

Syntax

```
setObjectCredentials('<WORKSPACE_NAME>', '<CLASS_NAME>', '<OBJECT_IDENTIFIER>', '<USER_NAME>', '<PASSWORD>')
```

This method requires the following parameters:

Argument	Definition
<i>WORKSPACE_NAME</i>	The name of the Capture workspace that contains the credential that will be modified. If the calling user has rights to more than one workspace with the specified name, the first workspace found will be used.
<i>CLASS_NAME</i>	The name of the class required to modify the credentials.
<i>OBJECT_IDENTIFIER</i>	A key/value pair consisting of the property name of the object to modify and the expected value.
<i>USER_NAME</i>	The user name to assign to the object's credential.
<i>PASSWORD</i>	The password to assign to the object's credential.

Note:

To obtain a list of workspaces, use the `listWorkspaces` command.

Example

- Valid class names include the following:

```
oracle.odc.importprocessor.ImportJob
oracle.odc.entity.CommitProfileEntity
```

- The following is an example for setting credentials associated with an Email Import Job:

```
setObjectCredentials('TEST-
WORKSPACE', 'oracle.odc.importprocessor.ImportJob', 'jobName=import
Email', 'xyz@example.com', 'welcome123')
```

- The following is an example for setting credentials associated with a WebCenter Imaging commit profile:

```
setObjectCredentials('TEST-
WORKSPACE', 'oracle.odc.entity.CommitProfileEntity', 'profileName=commit to
Imaging', 'admin', 'welcome123')
```

5.2.19 updateScript

Use with WLST: Online.

Description

Updates a script within a workspace.

Syntax

```
updateScript('<WORKSPACE_NAME>', '<SCRIPT_NAME>', '<SCRIPT_FILE>')
```

This command requires the following parameters:

Argument	Definition
<i>WORKSPACE_NAME</i>	The name of the workspace that contains the script. The name should be used to lookup the unique identifier of the workspace. If the name is not found, an error message is displayed.
<i>SCRIPT_NAME</i>	The name of the script to update. If the script cannot be found, an error message is displayed.
<i>SCRIPT_FILE</i>	The absolute path to the script file that should be updated. The script should be updated with the contents of this file.

 **Note:**

To obtain a list of workspaces, use the `listWorkspaces` command.

Example

The following example updates a script within a workspace:

```
updateScript('TEST-WORKSPACE', 'Client_Script', '/scratch/aime/scripts/client.txt')
```

5.2.20 cloneWorkspaceFromID

Use with WLST: Online.

Description

Creates a unique copy of a workspace specified by its ID.

Syntax

```
cloneWorkspaceFromID(WORKSPACE_ID, 'WORKSPACE_NAME', 'WORKSPACE_DESCRIPTION')
```

This command requires the following parameters:

Argument	Definition
<i>WORKSPACE_ID</i>	The ID number of the workspace to be copied. Can be optionally specified within single quotes. For example, both 3 and '3' are valid arguments for a workspace with ID 3.
<i>WORKSPACE_NAME</i>	Name for the new workspace being created.
<i>WORKSPACE_DESCRIPTION</i>	Description for the new workspace being created.

 **Note:**

To obtain a list of valid workspace IDs, use the `listWorkspaces` command.

Example

The following example clones an existing workspace:

```
cloneWorkspaceFromID(3, 'Expenses', 'Expenses Workspace')
```

5.2.21 cloneWorkspaceFromFile

Use with WLST: Online.

Description

Reads an output file created by the `exportWorkspace` command and creates a unique copy of the workspace.

Syntax

```
cloneWorkspaceFromFile('FILE_NAME', 'WORKSPACE_NAME', 'WORKSPACE_DESCRIPTION')
```

This command requires the following parameters:

Argument	Definition
<i>FILE_NAME</i>	The name of the output file from which the content has to be copied.
<i>WORKSPACE_NAME</i>	Name for the new workspace being created.
<i>WORKSPACE_DESCRIPTION</i>	Description for the new workspace being created.

Example

The following example reads an output file to create a workspace:

```
cloneWorkspaceFromFile('/home/bills/expense1.xml', 'Expenses', 'Expenses Workspace')
```

5.2.22 listImportHATokens

Use with WLST: Online.

Description

Displays all the Import Processor tokens for a specific workspace.

Syntax

```
listImportHATokens(WORKSPACE_ID)
```

This command requires the following parameters:

Argument	Definition
<i>WORKSPACE_ID</i>	The ID number of the workspace for which the tokens have to be displayed. Can be optionally specified within single quotes. For example, both 3 and '3' are valid arguments for a workspace with ID 3.

 **Note:**

To obtain a list of valid workspace IDs, use the `listWorkspaces` command.

Example

The following example displays Import Processor tokens for the workspace with ID = 3:

```
listImportHATokens(3)
```

5.2.23 deleteImportHAToken

Use with WLST: Online.

Description

Deletes a specific Import Processor HA token from the `ECIMPORTHATOKEN` and `ECIMPORTHATOKENITEMS` tables.

Syntax

```
deleteImportHAToken('TOKEN_ID')
```

This command requires the following parameters:

Argument	Definition
<i>TOKEN_ID</i>	ID of the token to be deleted.

Example

The following example deletes a specific Import Processor HA token:

```
deleteImportHAToken('d:\bills\ha_token.lst')
```

A

Capture Object Properties

This appendix lists all the properties that can be set for each Capture object using the `setObjectProperty` command.

This appendix contains the following sections:

- [Client Profile Object Properties](#)
- [Import Processor Job Object Properties](#)
- [Recognition Processor Job Object Properties](#)
- [Commit Profile Object Properties](#)
- [Document Conversion Job Object Properties](#)

A.1 Client Profile Object Properties

The following table lists all the properties that can be set for a Client Profile object (class name: `oracle.oddc.data.Profile`).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Profile Name	<code>profileName</code>	String	Any value
Description	<code>profileDesc</code>	String	Any value
Online	<code>profileStatus</code>	boolean	Any value
Apply Default Brightness and Contrast	<code>applyBrightness</code>	boolean	Any value
Batch Prefix	<code>batchPrefix</code>	String	Any value
Profile Type	<code>scanningType</code>	int	1 - Capture only 2 - Capture and Index 3 - Index only
Non-Image File Preview Page Limit	<code>maxPages</code>	int	Any value
Default Status	<code>defaultStatus</code>	String	Any value
Batch Priority	<code>defaultPriority</code>	int	Any value
Document Creation Option	<code>docOption</code>	int	1 - One page (for example, Simplex) 2 - Two pages (for example, Duplex) 3 - Variable number of pages 4 - Prompt the user
Separator Sheet Byte Threshold	<code>sepByteThreshold</code>	int	Any value
Blank Page Byte Threshold:	<code>blankByteThreshold</code>	long	Any value
Database Lookup Profile	<code>dbLookupProfile</code>	String	Any value
Always Display Records	<code>dbLookupHitList</code>	boolean	Any value

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Dependent Choice List	pickListRelationshipProfile	String	Any value
Records Returned Limit	dbLookupMaxRecords	int	Any value
Brightness	brightness	int	Any value
Contrast	contrast	int	Any value
Batch Visibility	batchVisibility	int	0 - User and workstation 1 - User 2 - All users
Non-Image File Import Action	nonImageImportAction	int	0 - Do not import 1 - Import in native format 2 - Convert to image format
Additional Batch Prefixes (separate by ;)	filterPrefix	String	Any value
Days Old: From	filterDaysFrom	int	Any value
Days Old: To	filterDaysTo	int	Any value
Batch Statuses	filterStatus	List<String>	Any value
Priorities	filterPriority	List<Integer>	[0,1,2,3,4,5,6,7,8,9,10]
Primary Sort	filterPrimarySort	String	BatchName/ BatchPageCount/ BatchDate/ BatchPriority/ BatchStatus
Primary Sort Order	filterPrimaryOrder	int	0 - Ascending 1 - Descending
Secondary Sort	filterSecondarySort	String	BatchName/ BatchPageCount/ BatchDate/ BatchPriority/ BatchStatus
Secondary Sort Order	filterSecondaryOrder	int	Any value
Document Profiles	supportedDocumentTypes	List<String>	Any value
Batch Processor	batchProcessorID	String	oracle.odc.docconverter.processor.DocumentConverterProcessor/ oracle.odc.commitprocessor.CommitProcessor/ oracle.odc.recognitionprocessor.RecognitionProcessor
Batch Processor Job	batchProcessorJobID	String	Any value

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Default Color	defaultColor	int	0 - Not specified 1 - Black and white 2 - Gray scale 3 - Color
Default DPI:	defaultDPI	int	0(Not specified)/100 /150/200/ 240/300/400/600
Prevent Default Override	preventDefaultColorOve rride	boolean	Any value
Prevent Default Override	preventDefaultDPIOverr ide	boolean	Any value
Non-Image conversion : Color	convertColor	int	0 - TIFF: black and white 1 - JPEG: color
Non-Image conversion :JPEG Image Quality	convertQuality	int	Any value
Non-Image conversion:DPI	convertDPI	int	0(Not specified)/100 /150/200/ 240/300/400/600

A.2 Import Processor Job Object Properties

The following section lists all the properties that can be set for an Import Processor Job object (class name: oracle.odc.importprocessor.ImportJob).

This section contains the following topics:

- [General Import Job Properties](#)
- [Folder Import Job Specific Properties](#)
- [List File Import Job Specific Properties](#)
- [Email Job Specific Properties](#)

A.2.1 General Import Job Properties

The following table lists the General Import Job properties that can be set (use class name: oracle.odc.importprocessor.ImportJob).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Online	isJobOnline	Boolean	Any value
Import Job Name	jobName	String	Any value

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Image Down-Sample	imageDownsample	Integer	0 - None (retain image format) 1 - Down-sample color to 8 bit (gray scale) 2 - Down-sample color or gray scale to black and white
JPEG Image Quality	jpegQuality	Integer	Any value
Batch Prefix	batchPrefix	String	Any value
Default Batch Status	defaultBatchStatusID	String	Any value
Default Batch Priority	defaultPriority	Integer	Any value
Database Lookup Profile	dbSearchID	String	Any value
Database Search Field	dbSearchFieldID	String	Any value
Default Document Profile	defaultDocumentTypeID	String	Any value
When more than one record is found	searchResultOption	Integer	0 - Use first record 1 - Do not select a record
Script	scriptID	String	Any value
Import Frequency	importFrequency	Integer	15: 15 seconds 30: 30 seconds 60: 1 minute 300: 5 minutes 900: 15 minutes 1800: 30 minutes 3600: 1 hour -1: Daily
Import Source	importSourceClassName	String	oracle.odc.importprocessor.folder.FolderSource / oracle.odc.importprocessor.email.EmailSource / oracle.odc.importprocessor.listfile.ListFileSource
Batch Processor	batchProcessorClassName	String	oracle.odc.docconverterprocessor.DocumentConverterProcessor/ oracle.odc.commitprocessor.CommitProcessor/ oracle.odc.recognitionprocessor.RecognitionProcessor
Batch Processor Job	batchProcessorJobID	String	Any value
Import Freq: Every Day: Time Hr:	hour	Integer	Any value
Import Freq: Every Day: Time Min:	minute	Integer	Any value
Default Locale	locale	Locale	Any value

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
If Image Validation Fails:	imageFailureAction	Integer	0 - Delete the batch 1 - Skip the file
Default Date Format	defaultDateFormat	String	Any value
Description	description	String	Any value
Encoding	encoding	String	Any value

A.2.2 Folder Import Job Specific Properties

The following table lists the Folder Import Job properties that can be set (use class name: oracle.odc.importprocessor.ImportJob).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Import Folder Path	folder	String	Any value
Process subfolders	processSubfolders	boolean	Any value
Create a New Batch	batchCreationOption	int	0 - Per file 1 - Per folder
File Mask(s)	fileMasks	String	Any value
Primary Sort Type	primarySortType	FolderSortType	0 - None 1 - File Name 2 - File Extension 3 - File Modified Date
Primary Sort Order	primarySortOrder	FolderSortOrder	0 - Ascending 1 - Descending
Secondary Sort Type	secondarySortType	FolderSortType	0 - None 1 - File name 2 - File extension 3 - File modified date
Secondary Sort Order	secondarySortOrder	FolderSortOrder	0(Ascending) / 1(Descending)
Ready File	readyFile	String	Any value
After Import	postProcessingOption	int	0 - Delete file 1 - Change file extension 2 - Add file prefix
New Extension	fileExtension	String	Any value
Prefix	filePrefix	String	Any value
Delete processed subfolder if empty	deleteSubfolders	boolean	Any value

A.2.3 List File Import Job Specific Properties

The following table lists the List File Import Job properties that can be set (use class name: oracle.odc.importprocessor.ImportJob).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Import Folder Path	folder	String	Any value
Create a New Batch:	batchCreationOption	int	0 - Per list file 1 - Per folder
File Mask(s)	fileMasks	String	Any value
Process subfolders	processSubfolders	boolean	Any value
Field Delimiter	fieldDelimiter	String	Any value
Maximum Fields Per Document	maxFieldsPerRecord	int	Any value
Document File Field Position	documentFieldPosition	int	Any value
After Import	postProcessingOption	int	0 - Delete file 1 - Change file extension 2 - Add file prefix
New Extension	fileExtension	String	Any value
Prefix	filePrefix	String	Any value
Delete document files after successful import	deleteDocumentFiles	boolean	Any value

A.2.4 Email Job Specific Properties

The following table lists the Email Job properties that can be set (use class name: oracle.odc.importprocessor.ImportJob).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
IMAP Connection Security	imapConnectSecurity	EmailConnection Security	0 - NONE 1 - STARTTLS 2 - SSL_TLS
Email Server Name (DNS name or IP address)	serverName	String	Any value
Port	port	int	Any value
Folders to Process	folderList	List<String>	Any value
Message Filters: From Address	filterFromAddressEnabled	boolean	Any value
Message Filters: From Address Field Contains	filterFromAddress	String	Any value

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Message Filters: Subject	filterSubjectEnabled	boolean	Any value
Message Filters: Subject Field Contains	filterSubject	String	Any value
Message Filters: Message Body	filterMessageBodyEnabled	boolean	Any value
Message Filters: Message Body Field Contains	filterMessageBody	String	Any value
Include attachments matching these mask(s)	attachmentsMask	String	Any value
Exclude attachments matching these mask(s)	attachmentsExcludeMask	String	Any value
Email Message Options	includeMessageBodyOption	int	0 - Import as text 1 - Import in EML format
Import message body file	includeMessageBodyFile	boolean	Any value
Import entire message in EML format	includeEntireEMLFile	boolean	Any value
Document Ordering	emailDocumentOrder	List<Integer>	0 - Message body (text file) 1 - Message body (EML file) 2 - Email message (EML file) 3 - Attachment(s)
Include when no attachments exist	includeMessageBodyWithout Attachments	boolean	Any value
Upon Failed Import	postProcessingFailureOption	int	0 - Delete the message 1 - Move the message to another folder 2 - Do not delete the message
Upon Successful Import:	postProcessingSuccessOption	int	0 - Delete message 1 - Move message to another folder
Folder Name:	postProcessingFailureMoveFolder	String	Any value
Folder Name	postProcessingSuccessMoveFolder	String	Any value

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Search Operator	searchOperator	int	0 - AND 1 - OR
Determined based on email importance	determineBatchPriority	boolean	Any value
Email Importance Low: Batch Priority	batchPriorityLow	int	Any value
Email Importance Normal: Batch Priority	batchPriorityNormal	int	Any value
Email Importance High: Batch Priority	batchPriorityHigh	int	Any value
Email Accounts to Process	emailAccounts	List<String>	Any value

A.3 Recognition Processor Job Object Properties

The following table lists all the properties that can be set for a Recognition Processor Job object (class name: oracle.odc.recognition.RecognitionJob).

This section contains the following topics:

- [General Recognition Job Properties](#)
- [BarcodeDefinition Class Properties](#)
- [DocumentDefinition Class Properties](#)
- [RecognitionJobField Class Properties](#)
- [SeparatorDefinition Class Properties](#)

A.3.1 General Recognition Job Properties

The following table lists the General Recognition Job properties that can be set (use class name: oracle.odc.recognition.RecognitionJob).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Recognition Job Name	name	String	Any value
Description	description	String	Any value
Script	scriptID	String	Any value
Online	online	Boolean	Any value
Bar Codes	barcodes	List<Barcode Definition>	Any value
Enable Auto-Detect	autoDetectBarcodes	Boolean	Any value
Validate optional checksum	validateChecksum	Boolean	Any value

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Bar Code Recognition	sybologies	List<Integer>	[0,1,2,3,4,5,6,7,8,9] (Auto-Detectable 1-D Sybologies) / [10,11,12,13,14,15,16,17,18](Other 1-D Sybologies) / [19,20,21](2-D Sybologies)
Organize documents based on	batchOrganization	Integer	0 - Fixed number of pages per document 1 - None: Do not perform document organization 2 - Same bar code value on each page 3 - Separator pages 4 - Hierarchical separator pages
Number of Pages Per Document:	documentPageCount	Integer	Any value
Number of pages per document to read bar codes:	pagesPerDoc2ReadBarcodes	Integer	Any value
Maximum Number of Pages Per Document:	maxPageCountPerDoc	Integer	Any value
Optimistic bar code detection	optimizeBarcodeDetection	Boolean	Any value
Separator Page:	coverPages	List<SeparatorDefinition>	Any value
If more than one value is found for a bar code within a document	multiBarcodeValuesOption	Integer	0 - Use the first value found 1 - Always overwrite the bar code value 2 - Clear the bar code value
Source Document Attachments:	sourceDocAttachments	Integer	0 - Include all attachments to created documents 1 - Include attachments with matching Document Profile attachment types 2 - Do not include attachments

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Dynamic Document Profile Option:	dynamicDocType	Integer	0 - Do not determine dynamically 1 - Determine dynamically using bar code 2 - Determine dynamically using separator page
Default Document Profile:	defaultDocTypeID	String	Any value
Document Profile and Bar Code Value Mappings	docTypeMappings	List<DocumentDefinition>	Any value
Fields	jobFields	List<RecognitionJobField>	Any value
DB Lookup Using:	dblookupUsing	Integer	0 - None 1 - Bar code 2 - Field value
Field Name	dblookupIndexDefID	String	Any value
Database Lookup Profile	dblookupProfile	String	Any value
Database Search Field:	dblookupSearchField	String	Any value
When more than one record is found:	dblookupMultipleRecordAction	Integer	0 - Use first record 1 - Do not link
When no record is found:	dblookupNoMatchAction	Integer	0 - Allow document commit 1 - Prevent document commit
To rename the batch, enter a prefix:	renamePrefix	String	Any value
To send email notification, enter email address:	renameEmail	String	Any value
To change batch status, select a status:	renameStatus	String	Any value
To change batch priority, enter a priority between 1 to 10:	renamePriority	Integer	Number between 1 to 10
Batch Processor	processorID	String	Any value
Batch Processor Job:	processorJobID	String	Any value
To rename the batch, enter a prefix:	failureRenamePrefix	String	Any value

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
To send email notification, enter email address:	failureRenameEmail	String	Any value
To change batch status, select a status:	failureRenameStatus	String	Any value
To change batch priority, enter a priority between 1 to 10:	failureRenamePriority	Integer	Number between 1 to 10
Batch Processor	failureProcessorID	String	Any value
Batch Processor Job:	failureProcessorJobID	String	Any value
Bar Code Engine	decoder	Integer	0 - Universal Decoder 1 - Windows Decoder
Maximum Bar Codes per Image	maxBarcodes	Integer	Number between 1 to 10
Minimum Bar Code Height	minHeight	Double	Any valid value between 0.2 to 3.0
Unit of Measure	unitMeasure	Integer	0 - Inch 1 - Centimeter
Maximum Bar Code Width	maxWidth	Double	Any valid value between minHeight to 10
Minimum Bar Code Characters	minChars	Integer	Number between 1 to 1000

A.3.2 BarcodeDefinition Class Properties

The following table lists the BarcodeDefinition properties that can be set (use class name: oracle.odc.recognition.RecognitionJob).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Name	barcodeName	String	Any value
Validation Rules	validationRule	Integer	0 - None 1 - Length 2 - Mask 3 - Regular expression 4 - Choice list
Length	validationLength	Integer	Any value
Mask	validationMask	String	Any value
Regular Expression	validationRegularExpression	String	Any value
Choice List Source	pickListSourceID	String	Any value
Choice List Source	pickListID	String	Any value

A.3.3 DocumentDefinition Class Properties

The following table lists the DocumentDefinition properties that can be set (use class name: oracle.odc.recognition.RecognitionJob).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Bar Code Value Matches:	mappingType	Integer	0 - Single value 1 - Choice list
Single Value:	value	String	Any value

A.3.4 RecognitionJobField Class Properties

The following table lists the RecognitionJobField properties that can be set (use class name: oracle.odc.recognition.RecognitionJob).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Auto Populate	autoPopulate	Integer	0 - None 1 - Bar code 2 - Batch name 3 - Default 4 - Index date 5 - Scan date
Barcode	populateValue	String	Any value

A.3.5 SeparatorDefinition Class Properties

The following table lists the SeparatorDefinition properties that can be set (use class name: oracle.odc.recognition.RecognitionJob).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Rule Name	name	String	Any value
Delete this separator page upon commit	deleteUponCommit	Boolean	Any value
Operator	operator	Integer	0 - OR 1 - AND
Document Profile	docTypeID	String	Any value
Attachment Type	attachmentTypeID	String	Any value

A.4 Commit Profile Object Properties

The following table lists all the properties that can be set for a Commit Profile object (class name: oracle.odc.entity.CommitProfileEntity).

This section contains the following topics:

- [General Commit Profile Properties](#)
- [Text File Commit Profile Specific Properties](#)
- [Webcenter Content Commit Profile Specific Properties](#)
- [ContentAttributeMappingInfo Class Properties](#)
- [CaptureToContentFieldMapping Class Properties](#)
- [ContentFieldNameInfo Class Properties](#)
- [Webcenter Imaging Commit Profile Specific Properties](#)
- [PDF Searchable Document Output Specific Properties](#)
- [DOCS Commit Driver Properties](#)

A.4.1 General Commit Profile Properties

The following table lists the General Commit Profile properties that can be set (use class name: oracle.odc.entity.CommitProfileEntity).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Commit Profile Name	profileName	String	Any value
Document Output Format	exportDriverID	String	oracle.odc.export.PDF Searchable/ oracle.odc.export.PDFImageOnly/ oracle.odc.export.TIFF Multimage
Attachment Document Output Format	attachExportDriverID	String	oracle.odc.export.PDF Searchable/ oracle.odc.export.PDFImageOnly/ oracle.odc.export.TIFF Multimage
Commit Driver	commitDriverID	String	oracle.odc.commit.TextFileCommitDriver(Text File)/ oracle.odc.commit.cs.ContentCommitDriver(WebCenter Content) / oracle.odc.commit.ipm.IPCommitDriver(WebCenter Content Imaging)
Online	active	Boolean	Any value
Edit Execution Order	executionOrder	Integer	Any value
Error Handling Policy	errorHandlingPolicy	Integer	0 - Continue to the next document 1 - Cancel to next commit profile 2 - Cancel commit

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Restrict Commit to Document Profiles	documentTypes	List<DocumentTypeEntity>	Any value
Default Locale	defaultLocaleString	String	Any value
Default Date Format	defaultDateFormat	String	Any value
Encoding	encoding	String	Any value



Note:

To update the Default Locale field associated with a commit profile, pass a value that includes all the three parts of a locale, that is, language, country, and variant. Use a vertical bar (|) to separate the parts of the locale. The following example sets the locale to Chinese (China): defaultLocaleString = zh|CN|.

A.4.2 Text File Commit Profile Specific Properties

The following table lists the Text File Commit Profile properties that can be set (use class name: oracle.odc.entity.CommitProfileEntity).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Do not create Commit Text File	doNotCreateTextFile	boolean	Any value
Store in subfolders	useTextSubFolder	boolean	Any value
Exclude Attachments	excludeAttachment	boolean	Any value
Text File Folder : Subfolder Options	textSubFolderOption	int	0 - Year 1 - Year and month 2 - Year, month, and date
Commit Text File Folder	textFileFolder	String	Any value
File Prefix	textFilePrefix	String	Any value
File Extension	textFileExtension	String	Any value
Create a copy for each page	duplicateLinkedItems	boolean	Any value
Create a folder per committed batch	oneFolderPerBatch	boolean	Any value
Store in subfolders	useDocSubFolder	boolean	Any value

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Document Folder :Subfolder Options	docSubFolderOption	int	0 - Year 1 - Year and month 2 - Year, month, and date 3 - Metadata field(s)
Document Folder	docFolder	String	Any value
Selected Metadata Fields	subFolderIndexes	List<String>	Any value
SubFolder Path	subFolderPath	String	Any value
Field Delimiter	delimiterOption	int	0 - Comma 1 - Semicolon 2 - Other
Text Qualifier	textQualifier	int	0 -None 1 - Double quote 2 - Single quote
Other Character	formatDelimiter	String	Any value
Text Qualifier	formatFields	List<String>	Any value
Name document file based on metadata field values	nameDoc	boolean	Any value
If File Name Consists of Invalid Characters	docNameInvalidCharOption	int	0 - Remove invalid characters 1 - Cancel document commit
Document File Naming: Field Delimiter	docNameDelimiter	String	Any value
Fields to Include in Document File Name	docNameFields	List<String>	Any value

A.4.3 Webcenter Content Commit Profile Specific Properties

The following table lists the Webcenter Content Commit Profile properties that can be set (use class name: oracle.odc.entity.CommitProfileEntity).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Server URL	serverURL	String	Any value
Name document file based on metadata values	documentFileNaming	boolean	Any value
Document File Name	documentFilename	String	Any value
If file name consists of invalid characters:	removeDocFilenameInvalidChars	boolean	Any value

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Document Title	documentTitle	String	Any value
Default Account	documentAccount	String	Any value
Default Security Group	documentSecurityGroup	String	Any value
Default Type	documentType	String	Any value
Assign Values Dynamically	dynamicAssignment	boolean	Any value
Assign Value	dynamicMappingOption	int	0 - By field mappings 1 - By metadata field
Metadata Field	dynamicMappingIndexField	String	Any value
Choice List Source	dynamicMappingPicklistSourceId	String	Any value
Choice List	dynamicMappingPicklistId	String	Any value
Account	accountMappingField	String	Any value
Type	contentTypeMappingField	String	Any value
Security Group	securityGroupMappingField	String	Any value
Content Attribute Mapping	attributeMappingList	List<ContentAttributeMappingInfo>	Any value
Bypass Workflows	bypassWorkflow	boolean	Any value
Alternative Check-In Service	alternateCheckInService	String	Any value
Select Capture metadata fields to assign to Content Server fields	fieldMappingList	List<CaptureToContentFieldMapping>	Any value
Custom Fields	customMetadataList	List<ContentFieldMetadataInfo>	Any value
Exclude Attachments	excludeAttachments	boolean	Any value

A.4.4 ContentAttributeMappingInfo Class Properties

The following table lists the ContentAttributeMappingInfo Class properties that can be set (use class name: oracle.odc.entity.CommitProfileEntity).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Field Value	fieldValue	String	Any value
Type	docType	String	Any value
Security Group	docSecurityGroup	String	Any value
Account	docAccount	String	Any value

A.4.5 CaptureToContentFieldMapping Class Properties

The following table lists the CaptureToContentFieldMapping Class properties that can be set (use class name: oracle.odc.entity.CommitProfileEntity).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Content Server Field	captureField	String	Any value
Capture Field	contentField	String	Any value

A.4.6 ContentFieldNameInfo Class Properties

The following table lists the ContentFieldNameInfo Class properties that can be set (use class name: oracle.odc.entity.CommitProfileEntity).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Custom Field Name	name	String	Any value
Custom Field Caption	caption	String	Any value

A.4.7 Webcenter Imaging Commit Profile Specific Properties

The following table lists the Webcenter Imaging Commit Profile properties that can be set (use class name: oracle.odc.entity.CommitProfileEntity).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Commit Methods	commitMethod	int	0 - Direct commit 1 - Input agent commit
Search and append to matching Imaging documents	append	boolean	Any value
Imaging WebService URL	url	String	Any value
Security Policy	policy	int	0 - Basic security 1 - Token security
Capture Output Directory	inputDir	String	Any value
Input Agent Directory	inputAgentDir	String	Any value
Search Definition	searchName	String	Any value

A.4.8 PDF Searchable Document Output Specific Properties

The following table lists the PDF Searchable Document Output properties that can be set (use class name: oracle.odc.entity.CommitProfileEntity).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Create Text File	textFormat	Integer	0 - Do not create text file 1 - Create text file format
Languages:	language	String	comma separated language IDs [0-122] For example, "0, 2, 3, 18"
Text File Code Page	codePage	String	Any value
Professional Dictionaries	profDictionary	String	Comma separated professional dictionary name. For example, "English Medical Dictionary, German Legal Dictionary"
Color Image Quality	colorImageQuality	int	0 - Minimum: minimum size 1 - Good: medium size 2 - Best: large size
Compatibility	compatibility	String	PDF14/PDF15/ PDF16/PDF17/ PDFA1B/PDFA2B/ PDFA2U
Create Linear PDF for Efficient Web Viewing	linearized	boolean	Any value
Preserve Original Image Orientation	keepImgOrientation	boolean	Any value
Single Language Detection per Page	singleLangPerPage	boolean	Any value

A.4.9 DOCS Commit Driver Properties

The following table lists the DOCS Commit Driver properties that can be set (use class name: oracle.odc.entity.CommitProfileEntity).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Server URL	serverURL	String	Any value
Name	parentFolderName	String	Any value
ID	parentFolderID	String	Any value
Sub-Folder Fields:	subFolderFields	List<String>	Any value
Create Sub-Folders using Field Values	createSubFolders	boolean	Any value

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
If folder name consists of invalid characters:	folderInvalidCharOption	int	0 - Remove invalid characters 1 - Cancel document commit
Use original file name for non-image files	useOriginalFileName	boolean	Any value
Name document file based on metadata field values	nameDoc	boolean	Any value
Fields to Include in Document File Name:	docFileFields	List<String>	Any value
Field Delimiter	docFileDelimiter	String	Any value
If file name consists of invalid characters:	docFileInvalidCharOption	int	0 - Remove invalid characters 1 - Cancel document commit

A.5 Document Conversion Job Object Properties

The following table lists all the properties that can be set for a Document Conversion Job object (class name: oracle.odc.docconverter.DocConverterJob).

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Name:	jobName	String	Any value
Online:	online	boolean	Any value
JPEG Image Quality	jpegQuality	Integer	1<jpegQuality<=99
Documents to Convert:	docFilter	Boolean	true: Selected non-image documents false: All non-image documents
Attachments to Convert:	docAttachmentFilter	Boolean	Any value
Output Image Format:	outputImageFormat	Integer	0 - TIFF: black and white 1 - JPEG: color
File Name Filter:	filterpattern	String	Any value
Batch Merge Option:	documentMergeType	Integer	0 - Do not merge documents 1 - Merge the first document with all other documents 2 - Merge the last document with all other documents 3 - Merge all documents

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Source Document Page Placement:	mergeToBeginning	boolean	true: Start of the target document false: End of the target document
Apply source document's metadata values	applySourceMetadata	boolean	Any value
Allow target document's metadata values to be overwritten	overwriteDestinationMetadata	boolean	Any value
Description:	description	String	Any value
Batch Processor:	processorID	String	Any value
Batch Processor Job:	processorJobID	String	Any value
To rename the batch, enter a prefix:	renamePrefix	String	Any value
To send email notification, enter email address:	renameEmail	String	Any value
To change batch status, select a status:	renameStatus	String	Any value
To change batch priority, enter a priority between 1 to 10:	renamePriority	Integer	[1-10]
Batch Processor:	errProcessorID	String	Any value
Batch Processor Job:	errProcessorJobID	String	Any value
To rename the batch, enter a prefix:	errRenamePrefix	String	Any value
To send email notification, enter email address:	errRenameEmail	String	Any value
To change batch status, select a status:	errRenameStatus	String	Any value
To change batch priority, enter a priority between 1 to 10:	errRenamePriority	Integer	[1-10]
DPI:	dpi	Integer	100 / 150 /200 / 240/300/400/600
External Conversion Use:	extConvEnabled	Boolean	true: On false: Off
File Name Filter(s):	extConvFilterPattern	String	Any value
External Conversion Program:	extConvProgram	String	Any value
Command Line Parameters:	extConvCommandLine	String	Any value

Property Name (on the UI)	Property Name (in the Class)	Data Type	Valid Options
Process Monitoring Method:	extConvMonitoringMethod	Integer	0 - Duration time out 1 - Output file inactivity time out
Timeout (minutes):	extConvTimeout	Integer	[1-1000]
Success Return Code	extConvSuccessReturnCode	Integer	Any value
Do not convert:	doNotConvertDocs	Boolean	Any value
Do not convert:	doNotConvertDocAttachments	Boolean	Any value
File Name Filter:	attachmentFilterPattern	String	Any value
Source Attachments:	documentAttachmentMergeType	Integer	0 - Do not include attachments 1 - Include all attachments to the merged documents 2 - Include attachments with the matching Document Profile Attachment Types
Restrict to Document Profiles:	documentTypes	List<DocumentTypeEntity>	Any value
Restrict to Attachment Types:	attachmentTypes	List<DocumentAttachmentTypeEntity>	Any value