

## **Oracle® Beehive**

Application Developer's Guide

Release 1 (1.5)

**E14839-01**

April 2009

Oracle Beehive Application Developer's Guide, Release 1 (1.5)

E14839-01

Copyright © 2008, 2009, Oracle and/or its affiliates. All rights reserved.

Primary Author: Raymond Gallardo

Contributing Authors: Tait McCarthy, Jamie Rancourt

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

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software 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 which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on 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. 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.

---

---

# Contents

<b>Preface</b> .....	xix
Audience .....	xix
Documentation Accessibility .....	xix
Related Documents .....	xx
Conventions .....	xx
 <b>1 Developing with Oracle Beehive Platform Services</b>	
Oracle Beehive Workflow Service .....	1-1
Oracle Beehive API .....	1-1
Oracle Beehive Java Content Repository API .....	1-2
Required JAR Files .....	1-2
Oracle Beehive Web Services .....	1-2
 <b>2 Oracle Beehive Custom Workflows</b>	
Constructing XPathS .....	2-1
 <b>3 Oracle Beehive Web Services</b>	
Web Service Locations .....	3-2
Web Services Security and SAML .....	3-2
Creating Server-Side Auto-Login Wallet and Configuring it for Oracle Beehive .....	3-3
Adding Signed User Certificate and Associated Trusted Certificate to Server-Side Wallet...	3-3
Adding a Self-Signed Certificate .....	3-3
Adding a CA-Signed Certificate .....	3-4
Exporting Certificates to Client-Side Oracle Wallet .....	3-5
Configuring Oracle Beehive for SAML Authentication .....	3-5
Location of SAML WSDL Documents .....	3-7
Searching for Artifacts with Web Services .....	3-7
Searching for Artifacts with Filters .....	3-8
Searching for Artifacts with FindArtifacts Method of GeneralArtifactService .....	3-8
 <b>4 Oracle Beehive Web Services</b>	
Methods .....	4-1
AddressBookService Methods .....	4-1
CalendarService Methods .....	4-1

ConferenceService Methods .....	4-3
DeviceService Methods .....	4-3
DiscussionForumService Methods .....	4-3
DocumentService Methods .....	4-4
FolderService Methods .....	4-4
GeneralArtifactService Methods .....	4-5
GroupService Methods .....	4-6
MembershipService Methods .....	4-6
MessageService Methods .....	4-7
PreferenceService Methods .....	4-7
PresenceService Methods .....	4-8
WorkspaceService Methods .....	4-8
AddressBookService .....	4-9
GetAddressBooks .....	4-9
Effects .....	4-9
Exceptions .....	4-9
UpdateAddressBook .....	4-9
Effects .....	4-10
Exceptions .....	4-10
DeleteAddressBook .....	4-10
Effects .....	4-10
Exceptions .....	4-10
GetContact .....	4-11
Effects .....	4-11
Exceptions .....	4-11
GetAllContacts .....	4-11
Effects .....	4-11
Exceptions .....	4-11
UpdateContact .....	4-11
Effects .....	4-12
Exceptions .....	4-12
DeleteContacts .....	4-12
Effects .....	4-12
Exceptions .....	4-12
CalendarService .....	4-12
GetCalendars .....	4-13
Effects .....	4-13
Exceptions .....	4-13
GetTaskAssignments .....	4-13
Effects .....	4-13
Exceptions .....	4-13
UpdateTaskAssignment .....	4-13
Effects .....	4-14
Exceptions .....	4-14
DeleteTaskAssignments .....	4-14
Effects .....	4-14
Exceptions .....	4-14

UpdateCalendar .....	4-14
Effects .....	4-14
Exceptions .....	4-14
DeleteCalendar .....	4-15
Effects .....	4-15
Exceptions .....	4-15
GetEvents .....	4-15
Effects .....	4-15
Exceptions .....	4-15
UpdateEvent .....	4-16
Effects .....	4-16
Exceptions .....	4-16
DeleteEvents .....	4-16
Effects .....	4-16
Exceptions .....	4-16
UpdateEventSeries.....	4-16
Effects .....	4-17
Exceptions .....	4-17
GetEventSeries.....	4-17
Effects .....	4-17
Exceptions .....	4-17
DeleteEventSeries.....	4-17
Effects .....	4-18
Exceptions .....	4-18
DeleteInvitations .....	4-18
Effects .....	4-18
Exceptions .....	4-18
DeleteInvitationSeries .....	4-18
Effects .....	4-18
Exceptions .....	4-18
GetInvitationSeries .....	4-19
Effects .....	4-19
Exceptions .....	4-19
GetInvitations .....	4-19
Effects .....	4-19
Exceptions .....	4-19
UpdateInvitation .....	4-20
Effects .....	4-20
Exceptions .....	4-20
UpdateInvitationSeries.....	4-20
Effects .....	4-20
Exceptions .....	4-20
GetIsBusy .....	4-20
Effects .....	4-20
Exceptions .....	4-20
GetFreeBusy.....	4-21
Effects .....	4-21

Exceptions .....	4-21
GetReminders .....	4-21
Effects .....	4-21
Exceptions .....	4-21
UpdateReminder .....	4-21
Effects .....	4-22
Exceptions .....	4-22
DeleteReminder .....	4-22
Effects .....	4-22
Exceptions .....	4-22
DeleteTasks .....	4-22
Effects .....	4-22
Exceptions .....	4-22
DeleteTaskLists .....	4-22
Effects .....	4-23
Exceptions .....	4-23
GetTaskLists .....	4-23
Effects .....	4-23
Exceptions .....	4-23
GetTasks .....	4-23
Effects .....	4-23
Exceptions .....	4-23
UpdateTask .....	4-24
Effects .....	4-24
Exceptions .....	4-24
UpdateTaskList .....	4-24
Effects .....	4-24
Exceptions .....	4-24
UpdateRecurringEventSeries .....	4-24
Effects .....	4-25
Exceptions .....	4-25
ConferenceService .....	4-25
DeleteConferences .....	4-25
Effects .....	4-25
Exceptions .....	4-25
DeleteTemplate .....	4-25
Effects .....	4-25
Exceptions .....	4-25
GetConferences .....	4-26
Effects .....	4-26
Exceptions .....	4-26
GetEndedSessions .....	4-26
Effects .....	4-26
Exceptions .....	4-26
GetLogEntries .....	4-26
Effects .....	4-27
Exceptions .....	4-27

GetRunningSession.....	4-27
Effects .....	4-27
Exceptions .....	4-27
GetTemplate.....	4-27
Effects .....	4-28
Exceptions .....	4-28
UpdateConference .....	4-28
Effects .....	4-28
Exceptions .....	4-28
UpdateConferenceSession .....	4-28
Effects .....	4-28
Exceptions .....	4-28
UpdateTemplate.....	4-29
Effects .....	4-29
Exceptions .....	4-29
DeviceService.....	4-29
DeleteDevices .....	4-29
Effects .....	4-29
Exceptions .....	4-29
GetDevicePresence.....	4-29
Effects .....	4-30
Exceptions .....	4-30
GetDevices .....	4-30
Effects .....	4-30
Exceptions .....	4-30
UpdateDevice .....	4-30
Effects .....	4-30
Exceptions .....	4-30
DiscussionForumService.....	4-31
DeleteDiscussionForums .....	4-31
Effects .....	4-31
Exceptions .....	4-31
DeleteMessages .....	4-31
Effects .....	4-31
Exceptions .....	4-31
DeleteTopic .....	4-31
Effects .....	4-32
Exceptions .....	4-32
GetDiscussionForums .....	4-32
Effects .....	4-32
Exceptions .....	4-32
GetLastPost .....	4-32
Effects .....	4-32
Exceptions .....	4-32
GetMessages .....	4-33
Effects .....	4-33
Exceptions .....	4-33

GetTopics.....	4-33
Effects .....	4-33
Exceptions .....	4-33
PostMessage.....	4-34
Effects .....	4-34
Exceptions .....	4-34
UpdateDiscussionForum .....	4-34
Effects .....	4-34
Exceptions .....	4-34
UpdateTopic .....	4-34
Effects .....	4-35
Exceptions .....	4-35
DocumentService .....	4-35
CancelCheckoutDocument .....	4-35
Effects .....	4-35
Exceptions .....	4-35
CheckinDocument .....	4-35
Effects .....	4-35
Exceptions .....	4-35
CheckoutDocument .....	4-35
Effects .....	4-36
Exceptions .....	4-36
GetDocuments .....	4-36
Effects .....	4-36
Exceptions .....	4-36
GetDocumentsInFolder .....	4-36
Effects .....	4-36
Exceptions .....	4-36
GetContent .....	4-36
Effects .....	4-37
Exceptions .....	4-37
GetContentOfDocs .....	4-37
Effects .....	4-37
Exceptions .....	4-37
UpdateDocument .....	4-37
Effects .....	4-37
Exceptions .....	4-37
DeleteDocuments .....	4-38
Effects .....	4-38
Exceptions .....	4-38
FolderService .....	4-38
GetFolders .....	4-38
Effects .....	4-38
Exceptions .....	4-38
GetSubFolders .....	4-38
Effects .....	4-39
Exceptions .....	4-39



UpdateFolder .....	4-39
Effects .....	4-39
Exceptions .....	4-39
DeleteFolders .....	4-39
Effects .....	4-39
Exceptions .....	4-39
GeneralArtifactService .....	4-40
CopyArtifact .....	4-40
Effects .....	4-40
Exceptions .....	4-40
MoveArtifact .....	4-40
Effects .....	4-40
Exceptions .....	4-40
GetTags .....	4-40
Effects .....	4-41
Exceptions .....	4-41
UpdateTag .....	4-41
Effects .....	4-41
Exceptions .....	4-41
GetBonds .....	4-41
Effects .....	4-41
Exceptions .....	4-41
UpdateBond .....	4-42
Effects .....	4-42
Exceptions .....	4-42
DeleteBonds .....	4-42
Effects .....	4-42
Exceptions .....	4-42
LoadLinks .....	4-42
Effects .....	4-42
Exceptions .....	4-42
GetLinksInFolder .....	4-43
Effects .....	4-43
Exceptions .....	4-43
GetLinksToEntity .....	4-43
Effects .....	4-43
Exceptions .....	4-43
UpdateLink .....	4-43
Effects .....	4-43
Exceptions .....	4-44
DeleteLinks .....	4-44
Effects .....	4-44
Exceptions .....	4-44
GetExternalArtifacts .....	4-44
Effects .....	4-44
Exceptions .....	4-44
UpdateExternalArtifact .....	4-44

Effects .....	4-45
Exceptions .....	4-45
DeleteExternalArtifacts .....	4-45
Effects .....	4-45
Exceptions .....	4-45
GetEffectiveLocks .....	4-45
Effects .....	4-45
Exceptions .....	4-45
UpdateLock.....	4-45
Effects .....	4-46
Exceptions .....	4-46
DeleteLocks.....	4-46
Effects .....	4-46
Exceptions .....	4-46
DeleteTags.....	4-46
Effects .....	4-46
Exceptions .....	4-46
UpdateSubscription .....	4-46
Effects .....	4-47
Exceptions .....	4-47
DeleteSubscriptions .....	4-47
Effects .....	4-47
Exceptions .....	4-47
GetSubscription.....	4-47
Effects .....	4-47
Exceptions .....	4-47
GetSubscriptionList .....	4-48
Effects .....	4-48
Exceptions .....	4-48
GetSubscriptionTemplates .....	4-48
Effects .....	4-48
Exceptions .....	4-48
FindArtifacts.....	4-48
Effects .....	4-49
Exceptions .....	4-49
FindTags.....	4-49
Effects .....	4-49
Exceptions .....	4-49
FindAllTags.....	4-49
Effects .....	4-49
Exceptions .....	4-49
FindCategories .....	4-49
Effects .....	4-50
Exceptions .....	4-50
FindAllCategories .....	4-50
Effects .....	4-50
Exceptions .....	4-50

FindRelatedArtifacts.....	4-50
Effects .....	4-50
Exceptions .....	4-50
FindAllArtifactsWithTags .....	4-50
Effects .....	4-51
Exceptions .....	4-51
FindArtifactsModifiedSince .....	4-51
Effects .....	4-51
Exceptions .....	4-51
FindArtifactsWithTags.....	4-51
Effects .....	4-51
Exceptions .....	4-52
FindArtifactsLargerThan .....	4-52
Effects .....	4-52
Exceptions .....	4-52
GroupService .....	4-52
GetGroup.....	4-52
Effects .....	4-52
Exceptions .....	4-52
GetGroups.....	4-52
Effects .....	4-53
Exceptions .....	4-53
GetGroupsForUser.....	4-53
Effects .....	4-53
Exceptions .....	4-53
GetEffectiveGroupsForUser .....	4-53
Effects .....	4-53
Exceptions .....	4-53
GetAllUsersGroup .....	4-54
Effects .....	4-54
Exceptions .....	4-54
CreateGroup .....	4-54
Effects .....	4-54
Exceptions .....	4-54
DeleteGroup.....	4-54
Effects .....	4-54
Exceptions .....	4-54
UpdateGroup.....	4-55
Effects .....	4-55
Exceptions .....	4-55
MembershipService .....	4-55
GetUsers .....	4-55
Effects .....	4-55
Exceptions .....	4-55
UpdateUser .....	4-55
Effects .....	4-56
Exceptions .....	4-56

DeleteUser .....	4-56
Effects .....	4-56
Exceptions .....	4-56
GetDelegatedPrincipals.....	4-56
Effects .....	4-56
Exceptions .....	4-56
WhoAmI .....	4-57
Effects .....	4-57
Exceptions .....	4-57
MessageService.....	4-57
GetContentData.....	4-57
Effects .....	4-57
Exceptions .....	4-57
GetInbox .....	4-57
Effects .....	4-58
Exceptions .....	4-58
GetMessageBoxes.....	4-58
Effects .....	4-58
Exceptions .....	4-58
UpdateMessageBox .....	4-58
Effects .....	4-58
Exceptions .....	4-59
DeleteMessageBoxes.....	4-59
Effects .....	4-59
Exceptions .....	4-59
GetEmailMessages .....	4-59
Effects .....	4-59
Exceptions .....	4-59
GetNewMessages.....	4-60
Effects .....	4-60
Exceptions .....	4-60
GetUnreadMessages .....	4-60
Effects .....	4-60
Exceptions .....	4-60
UpdateMessages .....	4-60
Effects .....	4-61
Exceptions .....	4-61
SendMessage .....	4-61
Effects .....	4-61
Exceptions .....	4-61
DeleteEmailMessages .....	4-61
Effects .....	4-61
Exceptions .....	4-61
SendInstantMessage .....	4-62
Effects .....	4-62
Exceptions .....	4-62
GetInstantMessage.....	4-62

Effects .....	4-62
Exceptions .....	4-62
SaveDraft .....	4-62
Effects .....	4-63
Exceptions .....	4-63
PreferenceService .....	4-63
ActivatePresenceProfile .....	4-63
Effects .....	4-63
Exceptions .....	4-63
DeletePreferenceProfiles .....	4-63
Effects .....	4-63
Exceptions .....	4-63
DeletePreferences .....	4-64
Effects .....	4-64
Exceptions .....	4-64
GetActivePreferenceProfile .....	4-64
Effects .....	4-64
Exceptions .....	4-64
GetPreferenceProfiles .....	4-64
Effects .....	4-64
Exceptions .....	4-64
UpdatePreference .....	4-65
Effects .....	4-65
Exceptions .....	4-65
UpdatePreferenceProfile .....	4-65
Effects .....	4-65
Exceptions .....	4-65
PresenceService .....	4-65
GetPresence .....	4-66
Effects .....	4-66
Exceptions .....	4-66
GetPresences .....	4-66
Effects .....	4-66
Exceptions .....	4-66
GetRosterPresences .....	4-66
Effects .....	4-66
Exceptions .....	4-66
GetSubscriptionRoster .....	4-66
Effects .....	4-66
Exceptions .....	4-67
RequestPresenceSubscription .....	4-67
Effects .....	4-67
Exceptions .....	4-67
SetPresence .....	4-67
Effects .....	4-67
Exceptions .....	4-67
SetSubscriptionStatus .....	4-67

Effects .....	4-67
Exceptions .....	4-67
DeletePresenceSubscription .....	4-68
Effects .....	4-68
Exceptions .....	4-68
UpdateActivities .....	4-68
Effects .....	4-68
Exceptions .....	4-68
WorkspaceService .....	4-68
GetWorkspaces.....	4-68
Effects .....	4-69
Exceptions .....	4-69
GetWorkspaceTemplates .....	4-69
Effects .....	4-69
Exceptions .....	4-69
GetTrashItems .....	4-69
Effects .....	4-70
Exceptions .....	4-70
UnDeleteItems .....	4-70
Effects .....	4-70
Exceptions .....	4-70
PurgeTrash.....	4-70
Effects .....	4-70
Exceptions .....	4-70
DeleteWorkspaces.....	4-71
Effects .....	4-71
Exceptions .....	4-71
UpdateWorkspace.....	4-71
Effects .....	4-71
Exceptions .....	4-71
UpdateWorkspaceMembers .....	4-71
Effects .....	4-72
Exceptions .....	4-72
<b>Types .....</b>	<b>4-72</b>
WSActivity .....	4-72
WSActivityType .....	4-72
WSActorFreeBusy .....	4-72
WSAddressBook.....	4-72
WSArtifact .....	4-73
WSAttribute .....	4-74
WSAttributeApplication .....	4-74
WSAttributeDefinition .....	4-74
WSAttributeName .....	4-75
WSAttributeTemplate.....	4-75
WSBond .....	4-76
WSBuddyList.....	4-76
WSCalendar .....	4-76

WSCalendarEvent .....	4-77
WSCalendarEventParticipant.....	4-78
WSCalendarEventParticipantStatus .....	4-78
WSCalendarEventStatus .....	4-78
WSCalendarEventTransparency .....	4-79
WSCalendarEventType .....	4-79
WSCalendarExceptionToSeries .....	4-79
WSCalendarInvitation .....	4-79
WSCalendarSeriesUpdateMode .....	4-80
WSCategory .....	4-80
WSCategoryApplication .....	4-80
WSCommunity .....	4-80
WSConference .....	4-81
WSConferenceLogEntry .....	4-81
WSConferenceSession .....	4-81
WSConferenceSetting .....	4-82
WSConferenceTemplate .....	4-82
WSConflictResolutionMode .....	4-82
WSContact.....	4-82
WSContactType.....	4-85
WSContent .....	4-85
WSDateTime .....	4-85
WSDevice .....	4-86
WSDiscussionMessage .....	4-86
WSDocument .....	4-86
WSEmailParticipant.....	4-87
WSEnterprise .....	4-87
WSEntity .....	4-87
WSEntityType .....	4-88
WSEventSeries .....	4-89
WSExternalArtifact .....	4-89
WSFilter .....	4-89
WSFolder .....	4-90
WSForum.....	4-90
WSFreeBusyInterval .....	4-91
WSGroup .....	4-91
WSInstantMessage .....	4-92
WSInvitationSeries.....	4-92
WSLink .....	4-93
WSLocale .....	4-93
WSLocation .....	4-94
WSLock.....	4-94
WSLogicalOperator.....	4-95
WSMessage .....	4-95
WSMessageAnnotation .....	4-95
WSMessageBox.....	4-96
WSMessageFlag.....	4-96

WSMessageHeader .....	4-96
WSMessageType .....	4-97
WSNotificationChannel .....	4-97
WSOrganization .....	4-97
WSParticipant .....	4-97
WSParticipantRole .....	4-98
WSPermission .....	4-98
WSPhysicalLocationType .....	4-98
WSPredicate .....	4-98
WSPreference .....	4-98
WSPreferenceProfile .....	4-99
WSPreferenceProperty .....	4-99
WSPresence .....	4-100
WSPresenceContactMethod .....	4-100
WSPresenceStatusType .....	4-100
WSPriority .....	4-100
WSProjection .....	4-101
WSProjectionType .....	4-101
WSProperty .....	4-101
WSPropertyValue .....	4-101
WSProvisioningStatus .....	4-101
WSQuota .....	4-101
WSReminder .....	4-102
WSReminderAction .....	4-102
WSReminderActionType .....	4-103
WSReminderCondition .....	4-103
WSReminderRule .....	4-103
WSResource .....	4-103
WSResultStatus .....	4-103
WSRole .....	4-104
WSRoleDefinition .....	4-104
WSRule .....	4-104
WSScope .....	4-104
WSSearchResult .....	4-105
WSSearchResultItem .....	4-105
WSSensitivity .....	4-105
WSSortCriteria .....	4-106
WSSubscription .....	4-106
WSSubscriptionRoster .....	4-107
WSSubscriptionRosterElement .....	4-107
WSSubscriptionTemplate .....	4-107
WSTag .....	4-107
WSTask .....	4-108
WSTaskAssignment .....	4-108
WSTaskList .....	4-109
WSTaskParticipantStatus .....	4-110
WSTimedTrigger .....	4-110



WSTimedTriggerType.....	4-110
WSTopic.....	4-110
WSUser .....	4-110
WSUserType .....	4-111
WSVersion.....	4-111
WSVersioningType .....	4-111
WSWorkspace.....	4-112
WSWorkspaceParticipant .....	4-113
WSWorkspaceTemplate .....	4-113

## 5 Oracle Beehive Business Reporting

<b>Overview of Business Views.....</b>	<b>5-1</b>
Accessing a View.....	5-1
Saving a View .....	5-2
<b>Types of Views .....</b>	<b>5-2</b>
Regular Reporting Views.....	5-3
Sensitive Reporting Views.....	5-3
<b>Workspace Service Views .....</b>	<b>5-4</b>
bee_enterprises.....	5-4
bee_enterprise_bonds.....	5-5
bee_enterprise_users.....	5-7
bee_organizations .....	5-7
bee_team_wspc_members .....	5-9
bee_user_markers .....	5-10
bee_user_marker_totals.....	5-11
bee_workspaces.....	5-12
bee_workspace_documents.....	5-14
bee_workspace_trash.....	5-17
bee_wspc_member_roles .....	5-18
bee_wspc_role_groups .....	5-19
bee_wspc_role_users .....	5-20
bee_wspc_trash_totals.....	5-21
<b>Device Management Service Views.....</b>	<b>5-22</b>
dm_applications .....	5-22
dm_devices.....	5-22
dm_device_apps.....	5-23
dm_device_cmds.....	5-23
dm_device_logs.....	5-23
dm_dev_profs_to_dev_types .....	5-24
<b>Collaboration Service Views.....</b>	<b>5-24</b>
es_message_headers .....	5-24
es_message .....	5-24
<b>Message Delivery Service Views .....</b>	<b>5-25</b>
mds_delivery_status.....	5-25
<b>Mobility Service Views .....</b>	<b>5-25</b>
oma_ds_data_operations .....	5-25
oma_ds_data_stores.....	5-25

oma_ds_sessions .....	5-26
oma_ds_user_devices .....	5-26
oma_syncml_logs .....	5-26
<b>User Directory Service Views .....</b>	<b>5-27</b>
uds_address_books.....	5-27
uds_contacts.....	5-27
uds_external_persons .....	5-28
uds_groups.....	5-28
uds_group_contact_members .....	5-28
uds_group_members.....	5-29
uds_sync_profile .....	5-29
uds_users.....	5-29
<b>Creating Schema with Access Only to Oracle Beehive Views .....</b>	<b>5-30</b>

---

---

# Preface

The Oracle Beehive Application Developer's Guide describes how to integrate Oracle Beehive into your applications.

## Audience

The Oracle Beehive Application Developer's Guide is directed at any application developer or administrator who wants to integrate Oracle Beehive with BPEL processes or custom and third party applications.

## Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

### Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

### Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

### Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at <http://www.fcc.gov/cgb/consumerfacts/trs.html>, and a list of phone numbers is available at <http://www.fcc.gov/cgb/dro/trsphonebk.html>.

## Related Documents

For more information, see the following documents in the Oracle Beehive Release 1 documentation library:

- *Oracle Beehive Administrator's Guide*
- *Oracle Beehive Administrator's Reference Guide*
- *Oracle Beehive Concepts*
- *Oracle Beehive Deployment Guide*
- *Oracle Beehive Installation Guide for Linux*
- *Oracle Beehive Installation Guide for Microsoft Windows*
- *Oracle Beehive Installation Guide for Solaris Operating System (SPARC 64-Bit)*
- *Oracle Beehive Java Content Repository Java API Reference*

## Conventions

The following text conventions are used in this document:

Convention	Meaning
<b>boldface</b>	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.

---

# Developing with Oracle Beehive Platform Services

Oracle Beehive Platform Services provides you with the following tools that allow you to integrate Oracle Beehive into your applications:

- **Oracle Beehive Workflow Service** enables you to integrate Oracle Beehive with BPEL processes.
- **Oracle Beehive API** allow you to retrieve and manipulate data from Oracle Beehive and integrate it into custom and third party applications, which users may execute on remote systems (detached from Oracle Beehive).

## Oracle Beehive Workflow Service

Oracle Beehive Workflow Service enables you to create BPEL processes that can be invoked with Oracle Beehive custom policies.

A typical usage of Oracle Beehive Workflow Service is at the completion of an Oracle Beehive operation, start a custom workflow that obtains the approval of one or more users, and perform another operation if the users are approved.

Oracle Beehive supports user-defined BPEL workflows that can either be invoked automatically from Oracle Beehive through policies or from some external source. Oracle Beehive can be integrated with BPEL Human Tasks so that these tasks show up in Beehive as task assignments. Refer to the Oracle Beehive Web site on Oracle Technology Network for a tutorial that demonstrates how to define a BPEL process that is invoked from Beehive and leverages task integration.

## Oracle Beehive API

Oracle Beehive API allow you to retrieve and manipulate collaborative data from Oracle Beehive and integrate it into custom and third party applications, which users may execute on remote systems (detached from Oracle Beehive).

Oracle Beehive API consist of the following interfaces:

- Oracle Beehive
- **Oracle Beehive Java Content Repository API** enables you to manipulate an Oracle Beehive instance's workspaces and its data like a content repository.
- **Oracle Beehive Web services** Oracle Beehive is a unified representation of Oracle Beehive artifacts. It provides you with WSDL files so you may invoke these Web services from your custom applications through standard protocols such as SOAP.

## Oracle Beehive Java Content Repository API

Oracle Beehive Java Content Repository (JCR) API implements the Content Repository API for Java Technology specification (Java Specification Request 170, version 1.0). You may find this specification at <http://jcp.org/en/jsr/detail?id=170>. The Content Repository API is a common, standardized Java interface for content repositories. With the Oracle Beehive JCR API, you may access and manipulate an Oracle Beehive instance's workspaces and its data like a content repository.

For more information about how to use the Oracle Beehive JCR API, including code samples, refer to the Oracle Beehive Web site on Oracle Technology Network.

For Oracle Beehive JCR API Javadoc, refer to *Oracle Beehive Java Content Repository Java API Reference*.

### Required JAR Files

Clients that use the Oracle Beehive JCR API require the following JAR files:

- `<Oracle home>/beehive/jlib/beehive_jcr-1.0.jar`
- `<Oracle home>/j2ee/home/lib/ejb.jar`
- `<Oracle home>/j2ee/home/lib/ejb30.jar`
- `<Oracle home>/j2ee/home/lib/oc4j-internal.jar`
- `<Oracle home>/j2ee/home/oc4jclient.jar`
- `<Oracle home>/lib/xmlparserv2.jar`
- `<Oracle home>/opmn/lib/optic.jar`

## Oracle Beehive Web Services

Oracle Beehive Web services enables you to develop Web applications in any environment, including non-OC4J ones such as .NET, which can generate a proxy implementation from WSDL files. Oracle Beehive Oracle Beehive Oracle Beehive

All Oracle Beehive Web services methods require an argument of type `WSEntity`, which can be a user, group, or workspace. This allows methods to be overloaded; their behavior varies depending on `WSEntity`. The `WSEntity` type itself holds other information such as name, description, and ID. In addition, all commands that update an entity are overloaded to either update or create based on the input provided.

---

## Oracle Beehive Custom Workflows

Oracle Beehive's integration with Oracle BPEL Process Manager allows you to create your own BPEL workflow processes and deploy them in your environment. A BPEL process can leverage Oracle Beehive task integration, and it can be invoked either from Oracle Beehive or from external sources.

Refer to "Oracle Beehive Custom Workflow" found in the Oracle Beehive Web site on Oracle Technology Network for a tutorial that demonstrates how to define a BPEL process that is invoked from Oracle Beehive and leverages BPEL task integration.

In this tutorial, you define an Oracle Beehive policy that will determine when the BPEL process will be invoked. This policy definition also determines which XML data is sent to the workflow process, which is defined with JXPaths. The following section describes how to construct JXPaths.

### Constructing JXPaths

Each JXPath has the following structure:

```
<method>#<method>#...@<XML element name>
```

For example, the following is the first JXPath in this policy, which retrieves the identifier of the newly added document, then maps this value to the <entityId> element in the generated XML:

```
getCommonAttributes#getEntityId#toCollabId@entityId
```

To get the list of available attributes for a particular event, use the command `beectl list_events --name <event name>`. For <event name>, use the value in the <eventTypeName> element in the policy.

For example, this policy invokes the `HelloWorldWFAction` when a document is created. The event name for this is `DOCUMENT_CREATED`. The following is a description of this event and its attributes:

```
beectl list_events --name DOCUMENT_CREATED
```

```
Event Name: DOCUMENT_CREATED
```

```
Event Description: Raised when a new document is created in a heterogeneous folder. This event is also raised when an existing document is copied to a new heterogeneous folder.
```

```
Is Synchronous: N
```

```
-----
Event Subscriptions:
```

```
-----
Name: HelloWorldDocCreated
```

```
Event Subscription Id: 26C9:2946:evts:37275AA4FF80934DE040578C201A154B0000000493F5
```

```
Event Action Function: oracle.ocs.management.model.WorkflowService:HelloWorld
```

Is PLSQL Action: N

-----  
Event Attributes:  
-----

Name: COMMON_ATTRIBUTES	Type: OCS_CODE.ECA_COMMON_EVENT_ATTRIBS_T
Name: ENTITY_ID	Type: OCS_CODE.OCS_COLLAB_ID_T
Name: CONTAINER	Type: OCS_CODE.OCS_COLLAB_ID_T
Name: ACTOR_ID	Type: OCS_CODE.OCS_COLLAB_ID_T
Name: OPERATION	Type: STRING
Name: STATUS	Type: STRING
Name: MESSAGE	Type: STRING
Name: EVENT_NAME	Type: STRING
Name: LOGON_RECORD_ID	Type: INTEGER
Name: EVENT_ID	Type: INTEGER
Name: CUSTOM_ATTRIBUTES	Type: OCS_CODE.WS_DOCUMENT_EVENT_ATTRIBS_T
Name: ARTIFACT_ATTRIBUTES	Type: OCS_CODE.AM_COMMON_EVENT_ATTRIBS_T
Name: SIZE_CHANGE	Type: INTEGER
Name: NEW_CONTAINER	Type: OCS_CODE.OCS_COLLAB_ID_T

To retrieve the value of an attribute, perform the following steps to the attribute's name:

1. Convert the attribute name to lower case
2. Capitalize the first letter of the attribute name. and the letter after each underscore
3. Remove all underscores
4. Append `get` to the beginning of the name

Attributes are stored in a tree structure. For example, to retrieve the value of `ENTITY_ID`, you will have to retrieve `COMMON_ATTRIBUTES` first.

You may use the following methods in your JXPath:

- `toCollabID`: Retrieves the identifier of the specified entity
- `ASEntity`: Converts the identifier back into an entity so that you can retrieve other attributes and entities from it, such as its name with `getName`.
- `iterator`: Use this with `next` to retrieve all entities in a list.



---

## Oracle Beehive Web Services

This module describes the Web services available in Oracle Beehive and how to access them.

The following are the names of all Oracle Beehive Web services:

- **AddressBookService Methods:** Manage personal or team workspace address books and their contents, such as contacts and groups.
- **CalendarService Methods:** Manage personal or team workspace calendars, task lists, meetings, invitations, tasks, task assignments, reminders and Free/Busy information.
- **ConferenceService Methods:** Manage Web conferences, including retrieving information about completed and running Web conference sessions.
- **DeviceService Methods:** Manage devices, which are supported client software installed on computers, such as Oracle Beehive Integration for Outlook, and mobile devices.
- **DiscussionForumService Methods:** Manage threaded, online discussion forums, in which users post messages about a topic within a forum. These methods manage and organize forums, topics, and messages.
- **DocumentService Methods:** Manage documents in personal or team workspaces.
- **FolderService Methods:** Manage messaging and document folders in personal or team workspaces.
- **GeneralArtifactService Methods:** Manage object metadata and data relationships, as well as provide cross artifact capabilities, searching, and notification capabilities.
- **GroupService Methods:** Directly manage personal, workspace, or system groups without having to navigate through scope details such as enterprise, organization and workspace.
- **MembershipService Methods:** Directly manage the memberships of groups and workspaces without having to load the contents or other attributes of the object.
- **MessageService Methods:** Manage personal or team workspace email and instant messages.
- **PreferenceService Methods:** Manage user preferences stored on the server and leveraged by Oracle Beehive clients.
- **PresenceService Methods:** Manage a user's presence information, subscribe to other user's presence, and view the XMPP roster.

- [WorkspaceService Methods](#): Manage personal or team workspaces and their top-level folders.

See [Appendix 4, "Oracle Beehive Web Services"](#) for descriptions of Oracle Beehive Web services methods and the data types they use.

## Web Service Locations

Each Web service URL has the following syntax:

```
http://<Oracle Beehive server host>:<Port>/ws/<Web service name>
```

When you open a Web service URL in a browser, you will find a form that enables you to input parameters and invoke different methods in the Web service.

This form also contains a link named "Service Description" that brings you to the WSDL for the Web service. The following is the syntax for the WSDL URL:

```
<Web service URL>?WSDL
```

For example, if your Oracle Beehive server host is `www.example.com` and its port is `7777`, then the following is the location for `WorkspaceService`:

```
http://www.example.com:7777/ws/WorkspaceService
```

The following is the location for the WSDL of `WorkspaceService`:

```
http://www.example.com:7777/ws/WorkspaceService?WSDL
```

To use Web services in a Java programming environment, you must generate proxy classes for each Web service you use. In JDeveloper, use the Web service WSDL URL to generate proxy classes.

## Web Services Security and SAML

Security Assertion Markup Language (SAML) is an XML-based framework for exchanging security information. Oracle Beehive Web services support SAML tokens for message authentication. SAML authentication enables you to secure Web Services clients with a different authentication mechanism that does not require the user name and password of any Oracle Beehive users. Enabling SAML token authentication for Oracle Beehive Web Services involves configuring the Oracle Beehive keystore, which is an Oracle Wallet.

Oracle Beehive Web Services supports SAML sender-vouches and SAML holder-of-key authentication methods.

In a sender-vouches scenario, an attesting entity has an existing trust relationship with the receiver (Oracle Beehive). The attesting entity vouches for the verification of the subject (such as a user invoking a Web Services client). In this scenario, you first establish the attesting entity (who is simply an arbitrary user) by adding a signed user certificate (issued by the attesting entity) and the associated trusted certificate to the Oracle Beehive keystore. Then, you establish the trust relationship between the attesting entity and Oracle Beehive with the command `beectl add_trusted_identity`. In your Web Services client, you specify the distinguished name of the attesting entity in the SAML token.

In a holder-of-key scenario, there is a third party involved, the identity provider, who has the existing trust relationship with the receiver. In this scenario, you add a signed user certificate (signed by the identity provider) and the associated trusted certificate to the Oracle Beehive keystore. In your Web Services client, you include the location of

a keystore in the SAML token. This keystore contains a user certificate (signed by the identity provider) and the associated trusted certificate.

For more information about SAML, refer to the following chapters and sections in *Oracle Application Server Web Services Security Guide*:

- Chapter 1, "Introduction"
- Chapter 3, "Administering Web Services Security" for information about SAML authentication methods
- "WS-Security" in Chapter 5, "Secure Web Service Usage Scenarios" for information about sender-vouches and holder-of-key use cases.

Configuring Oracle Beehive Web services to use SAML authentication involves the following steps:

1. [Creating Server-Side Auto-Login Wallet and Configuring it for Oracle Beehive](#)
2. [Adding Signed User Certificate and Associated Trusted Certificate to Server-Side Wallet](#)
3. [Exporting Certificates to Client-Side Oracle Wallet](#)
4. [Configuring Oracle Beehive for SAML Authentication](#)

To create a Web service client that uses Oracle Beehive Web services secured for SAML authentication, refer to "[Location of SAML WSDL Documents](#)", then refer to the Oracle Beehive Web site on Oracle Technology Network for sample code.

## Creating Server-Side Auto-Login Wallet and Configuring it for Oracle Beehive

If you have not already done so, create an auto-login Oracle Beehive and configure it for Oracle Beehive as described in "Configuring TLS with Oracle Wallet" in the Oracle Beehive installation guide of your operating system.

## Adding Signed User Certificate and Associated Trusted Certificate to Server-Side Wallet

You may either add a self-signed certificate or a CA-signed (certificate authority) certificate.

Use a self-signed certificate only for test purposes because it does not provide adequate security for a production environment. With a self-signed certificate, a user creates a certificate, then signs it with his or her own private key. In this case, the user himself or herself is verifying his or her own identity, and therefore, you should not trust the certificate.

With a CA-signed certificate, a user creates a certificate, then has a trusted CA, such as VeriSign, sign it. This means that the CA (who you trust) has verified the identity of the user (meaning that the CA certifies that the public key is of the claimed entity). Therefore you may trust the certificate.

Consequently, use a self-signed certificate in situations where security is not an issue, such as test environments. In a production environment, you should use a CA-signed certificate.

### Adding a Self-Signed Certificate

Add a self-signed certificate to the wallet with the following command:

```
orapki wallet add
-wallet <Oracle home>/Apache/Apache/conf/ssl.wlt/default/
```

```
-dn CN=user
-keysize 2048
-self_signed
-validity 3650
```

CN=user is the distinguished name of an arbitrary user who will issue the private key and also be the SAML authentication assertion issuer. Later, you will register this SAML authentication assertion issuer with Oracle Beehive.

### Adding a CA-Signed Certificate

1. Add a certificate request to the Oracle Beehive wallet:

```
orapki wallet add
-wallet <Oracle home>/Apache/Apache/conf/ssl.wlt/default/
-dn CN=user
-keysize 2048
-validity 3650
```

The directory `<Oracle home>/Apache/Apache/conf/ssl.wlt/default/` is the Oracle Beehive default wallet directory. CN=user is the distinguished name of an arbitrary user who will issue the private key and also be the SAML authentication assertion issuer. Later, you will register this SAML authentication assertion issuer with Oracle Beehive.

2. Export the certificate request to a file:

```
orapki wallet export
-wallet <Oracle home>/Apache/Apache/conf/ssl.wlt/default/
-dn CN=user
-request certificate_request.txt
```

The file `certificate_request.txt` is the exported certificate request.

3. With your certificate authority (CA) and your certificate request (`certificate_request.txt`), create a signed user certificate. In addition, export the trusted certificate from your CA. These steps use the file `user_certificate.txt` as the signed user certificate and the file `trusted_certificate.txt` as the trusted certificate exported from your CA.

You may use Oracle Wallet as a CA for testing purposes by following these steps.

- a. Create an auto-login wallet to act as a certificate authority. These steps assume that this wallet is stored in `/private/ca_wallet`. Create a signed certificate from the request for test purposes:

```
orapki cert create
-wallet /private/ca_wallet
-request certificate_request.txt
-cert user_certificate.txt
-validity 3650
```

The file `user_certificate.txt` is the signed user certificate.

- b. Export the trusted certificate from the CA wallet:

```
orapki wallet export
-wallet /private/ca_wallet
-dn CN=ca_user
-cert trusted_certificate.txt
```

The file `trusted_certificate.txt` is the exported (test) trusted certificate from the CA wallet.

4. Add the trusted certificate from the CA to the Oracle Beehive wallet:

```
orapki wallet add
-wallet <Oracle home>/Apache/Apache/conf/ssl.wlt/default/
-trusted_cert
-cert trusted_certificate.txt
```

5. Add the user certificate to the Oracle Beehive wallet:

```
orapki wallet add
-wallet <Oracle home>/Apache/Apache/conf/ssl.wlt/default/
-user_cert user_certificate.txt
```

## Exporting Certificates to Client-Side Oracle Wallet

1. Create a wallet with the `orapki wallet create` command on the machine from which you will run your Web services client. These steps assume that you have created this wallet in the directory `/private/client_side_wallet`.

2. If you are using a self-signed certificate, follow these steps:

- a. Export the self-signed user certificate to a file:

```
orapki wallet export
-wallet <Oracle home>/Apache/Apache/conf/ssl.wlt/default/
-dn CN=user
-cert exported_certificate.txt
```

- b. Add the self-signed certificate to the client-side wallet:

```
orapki wallet add
-wallet /private/client_side_wallet
-user_cert exported_certificate.txt
```

3. If you are using a CA-signed certificate, follow these steps:

- a. Add the trusted certificate from the CA to the client-side wallet:

```
orapki wallet add
-wallet /private/client_side_wallet
-trusted_cert
-cert trusted_certificate.txt
```

- b. Add the CA-signed user certificate to the wallet:

```
orapki wallet add
-wallet /private/client_side_wallet
-user_cert user_certificate.txt
```

---

**Note:** For testing purposes, you may simply copy the server-side keystore to the machine from which you will run your Web services client.

---

## Configuring Oracle Beehive for SAML Authentication

1. In the component `_authenticationService`, set the following properties with the `beectl` command:
  - `WsSecuritySamlEnabled: true`

- **WsSecuritySigKeyAlias**: This is the distinguished name you specified when you created a self-signed certificate or certificate request.

```
beectl list_properties --component _authenticationService
```

```
-----+-----
Property name      | Property value
-----+-----
...
WsSecuritySamlEnabled | false
-----+-----
WsSecuritySigKeyAlias |
-----+-----
WsSecuritySigKeyPwd   |
-----+-----
...
27 Record(s) displayed.
```

```
beectl modify_property
--component _authenticationService
--name WsSecuritySamlEnabled
--value true
```

Changes to configuration repository are not activated.  
Successfully stored the property for component id  
cfaaf634-df35-46da-b5e7-456672d9b495.

```
beectl modify_property
--component _authenticationService
--name WsSecuritySigKeyAlias
--value "CN=user"
```

Changes to configuration repository are not activated.  
Successfully stored the property for component id  
cfaaf634-df35-46da-b5e7-456672d9b495.

```
beectl list_properties --component _authenticationService
```

```
-----+-----
Property name      | Property value
-----+-----
...
*WsSecuritySamlEnabled | true
-----+-----
*WsSecuritySigKeyAlias | CN=user
-----+-----
...
27 Record(s) displayed.
NOTE:- * indicates that property value is changed and change is not yet
activated.
```

```
beectl activate_configuration
```

Local configuration files are not in sync with system model. Please run  
"modify\_local\_configuration\_files" manually.  
Proposed configuration is saved successfully and activated now.

```
beectl modify_local_configuration_files
```

The following local configuration files were modified for authentication  
 ...  
 Successfully ran the command in Oracle home /example/product/beehive.  
 Please run this command on all midtier instances.

2. Register the SAML authentication assertion issuer. Depending on the type of SAML authentication you are using, follow one of these steps:

- If you are using SAML sender-vouches authentication, register the SAML issuer as a trusted *service* identity with the following beectl command:

```
beectl add_trusted_identity
--type WSSEC
--is_service true
--key_alias CN=user
--service_name example.com
```

The DN CN=user is the alias of the trusted service certificate and example.com is an arbitrary name to identify this trusted service identity.

- If you are using SAML holder-of-key authentication, register the SAML issuer as a trusted *issuer* identity with the following beectl command:

```
beectl add_trusted_identity
-type WSSEC
--is_service false
--key_alias CN=user
--service_name example.com
```

---

**Note:** The --service\_name option specifies the name of the service that is being authenticated. It is used only for sender-vouches authentication.

When you use sender-vouches subject confirmation, Oracle Beehive first authenticates the service name, then authenticates the credential in the message (which is the user credential). This, you may interpret running the command `beectl add_trusted_identity` as "Add a service with name <service name> to the list of trusted services kept by Oracle Beehive. This service can perform actions for Oracle Beehive end users".

---

## Location of SAML WSDL Documents

After configuring Oracle Beehive for SAML authentication, the location of this WSDL document will be `http://<Oracle Beehive host>:<Web services port>/ws/saml/<Web service name>?WSDL`.

For example, if your Oracle Beehive instance is hosted on example.com, the Web services port is 7777, and you want to generate the proxy classes for WorkspaceService, you would use the SAML WSDL document located at `http://example.com:7777/ws/saml/WorkspaceService?WSDL`.

## Searching for Artifacts with Web Services

You may use the following methods to search for artifacts with Oracle Beehive Web Services:

- [Searching for Artifacts with Filters](#)

- [Searching for Artifacts with FindArtifacts Method of GeneralArtifactService](#)

## Searching for Artifacts with Filters

Each Web service contains one or more methods whose name starts with "get" that retrieves artifacts of the specified type. For example, the [GetWorkspaces](#) method of [WorkspaceService](#) retrieves workspaces.

Each of these get methods may take a [WSFilter](#) object as an input element. A filter determines the contents of the returned list of artifacts for the get method. It consists of the following:

- A [WSLogicalOperator](#) object named anyAllListRelation.
  - If the value of this object is AND, then an artifact will be selected by the filter if both matchAllList and matchAnyList evaluate to true.
  - If the value of this object is OR, then an artifact will be selected by the filter if either matchAllList or matchAnyList evaluate to true.
- A list of [WSPredicate](#) objects named matchAnyList. It evaluates to true if any of its predicates are satisfied
- A [WSProjection](#) object. A projection defines the amount of data retrieved for each item in the retrieved list of artifacts. It may have a value of FULL, NONE, BASIC, or META. Use NONE or BASIC if you expect the retrieved list of artifacts to be large and do not require detailed information for each artifact to improve the performance of the get method.
- A list of [WSPredicate](#) objects named matchAllList. It evaluates to true only if all of its predicates are satisfied
- A list of [WSSortCriteria](#) objects named sortCriteriaList. A sort criteria contains a parameter (for example, NAME) by which you want to sort the retrieved list of artifacts and the sort order you want the artifacts ordered (either ascending or descending)

A predicate contains the value you want to find, the kind of value you want to find, and the operation you want to use to evaluate the search. For example, you may define a predicate that determines whether a particular artifact contains (which is the operation) the string "Summary" (which is the value) in the name parameter (which is the kind of value you want to find).

## Searching for Artifacts with FindArtifacts Method of GeneralArtifactService

The searchString input element of the [FindArtifacts](#) method of [GeneralArtifactService](#) method accepts any string value, which is then used to perform a basic keyword search across all searchable artifacts in Oracle Beehive. Currently, Oracle Beehive supports searching of e-mail messages, documents, and calendar meetings. The following attributes are included in basic keyword searches in Oracle Beehive. Use wildcards (asterisk (\*)) and question mark (?) by default) to search all the following attributes except ID attributes, which only support exact matches:

- Documents
  - Name
  - Creator name
  - Creator ID (internal enterprise ID)
  - Content



- E-mail messages
  - Subject
  - Sender name
  - Sender ID (internal enterprise ID)
  - To, Cc, or Bcc recipient
  - Body
- Meetings
  - Title
  - Creator ID (internal enterprise ID)
  - Details
  - Invitees



## Oracle Beehive Web Services

This module lists all available Oracle Beehive Web services, their methods, and the data types they use.

### Methods

This section lists Oracle Beehive Web services, categorized by service name.

#### AddressBookService Methods

Provides methods to manage personal or team workspace address books and their contents, such as contacts and groups.

**Table 0–1 AddressBookService**

Method	Return Type
<a href="#">GetAddressBooks</a> ( tns:WSEntity uID, tns:WSEntity[] abookList, tns:WSFilter abFilter )	tns:WSAddressBook[]
<a href="#">UpdateAddressBook</a> ( tns:WSEntity uID, tns:WSEntity parentAB, tns:WSAddressBook abook )	tns:WSAddressBook
<a href="#">DeleteAddressBook</a> ( tns:WSEntity abID )	tns:WSResultStatus
<a href="#">GetContact</a> ( tns:WSEntity[] cID, tns:WSFilter contactFilter )	tns:WSContact[]
<a href="#">GetAllContacts</a> ( tns:WSEntity uID, tns:WSEntity abook, xsd:string contactType, tns:WSFilter contactFilter )	tns:WSContact[]
<a href="#">UpdateContact</a> ( tns:WSEntity uID, tns:WSEntity abook, tns:WSContact contact )	tns:WSContact
<a href="#">DeleteContacts</a> ( tns:WSEntity[] cIDList )	tns:WSResultStatus[]

#### CalendarService Methods

Provides methods for managing personal or team workspace calendars, task lists, meetings , invitations, tasks, task assignments, reminders and Free/Busy information.

**Table 0–2 CalendarService**

Method	Return Type
<a href="#">GetCalendars</a> ( tns:WSEntity uID, tns:WSEntity[] calIDList, tns:WSFilter calFilter )	tns:WSCalendar[]
<a href="#">GetTaskAssignments</a> ( tns:WSEntity uID, tns:WSEntity taskListID, tns:WSEntity[] taskAssignmentIDs )	tns:WSTaskAssignment[]

**Table 0–2 (Cont.) CalendarService**

Method	Return Type
<a href="#">UpdateTaskAssignment</a> ( tns:WSTaskAssignment task )	tns:WSTaskAssignment
<a href="#">DeleteTaskAssignments</a> ( tns:WSEntity taskAssignmentIDList )	tns:WSResultStatus[]
<a href="#">UpdateCalendar</a> ( tns:WSEntity uID, tns:WSCalendar calendar )	tns:WSCalendar
<a href="#">DeleteCalendar</a> ( tns:WSEntity calID )	tns:WSResultStatus
<a href="#">GetEvents</a> ( tns:WSEntity uID, tns:WSEntity calID, tns:WSEntity[] eventIDList, xsd:dateTime startTime, xsd:dateTime endTime, tns:WSFilter eventFilter )	tns:WSCalendarEvent[]
<a href="#">UpdateEvent</a> ( tns:WSEntity uID, tns:WSEntity calID, tns:WSCalendarEvent calEvent )	tns:WSCalendarEvent
<a href="#">DeleteEvents</a> ( tns:WSEntity[] eventIDList )	tns:WSResultStatus[]
<a href="#">UpdateEventSeries</a> ( tns:WSEntity uID, tns:WSEntity calID, tns:WSEventSeries eventSeries )	tns:WSEventSeries[]
<a href="#">GetEventSeries</a> ( tns:WSEntity uID, tns:WSEntity calID, tns:WSEntity[] eventSeriesIDList, tns:WSFilter eventSeriesFilter )	tns:WSEventSeries[]
<a href="#">DeleteEventSeries</a> ( tns:WSEntity eventSeriesID )	tns:WSResultStatus
<a href="#">DeleteInvitations</a> ( tns:WSEntity[] invitationIDList )	tns:WSResultStatus[]
<a href="#">DeleteInvitationSeries</a> ( tns:WSEntity invSeriesID )	tns:WSResultStatus
<a href="#">GetInvitationSeries</a> ( tns:WSEntity uID, tns:WSEntity calID, tns:WSEntity[] invitationSeriesIDList, tns:WSFilter invitationSeriesFilter )	tns:WSCalendarInvitation[]
<a href="#">GetInvitations</a> ( tns:WSEntity uID, tns:WSEntity calID, tns:WSEntity[] invitationIDList, xsd:dateTime startTime, xsd:dateTime endTime, tns:WSFilter inviteFilter )	tns:WSCalendarInvitation[]
<a href="#">UpdateInvitation</a> ( tns:WSCalendarInvitation ci )	tns:WSCalendarInvitation
<a href="#">UpdateInvitationSeries</a> ( tns:WSInvitationSeries is )	tns:WSCalendarInvitation
<a href="#">GetIsBusy</a> ( tns:WSEntity[] uIDList, xsd:dateTime time )	xsd:boolean[]
<a href="#">GetFreeBusy</a> ( tns:WSEntity[] uIDList, xsd:dateTime startTime, xsd:dateTime endTime )	tns:WSActorFreeBusy[]
<a href="#">GetReminders</a> ( tns:WSEntity uID, tns:WSEntity[] reminderIDList, tns:WSFilter remFilter )	tns:WSReminder[]
<a href="#">UpdateReminder</a> ( tns:WSEntity uID, tns:WSReminder rem )	tns:WSReminder
<a href="#">DeleteReminder</a> ( tns:WSEntity remID )	tns:WSResultStatus
<a href="#">DeleteTasks</a> ( tns:WSEntity[] taskIdList )	tns:WSResultStatus
<a href="#">DeleteTaskLists</a> ( tns:WSEntity[] taskListIDs )	tns:WSResultStatus
<a href="#">GetTaskLists</a> ( tns:WSEntity uID, tns:WSEntity[] taskListIDs )	tns:WSTaskList[]
<a href="#">GetTasks</a> ( tns:WSEntity uID, tns:WSEntity taskListID, tns:WSEntity[] taskIDs )	tns:WSTask
<a href="#">UpdateTask</a> ( tns:WSTask task )	tns:WSTask
<a href="#">UpdateTaskList</a> ( tns:WSEntity uID, tns:WSTaskList taskList )	tns:WSTaskList
<a href="#">UpdateRecurringEventSeries</a> ( tns:WSEntity uID, tns:WSEntity calID, tns:WSEventSeries eventSeries )	tns:WSEventSeries

## ConferenceService Methods

Provides methods that manage Web conferences, including retrieving information about completed and running Web conference sessions.

**Table 0–3** *ConferenceService*

Method	Return Type
<a href="#">DeleteConferences</a> ( tns:WSEntity[] confIDList )	tns:WSResultStatus[]
<a href="#">DeleteTemplate</a> ( tns:WSEntity cTemplateID )	tns:WSResultStatus[]
<a href="#">GetConferences</a> ( tns:WSEntity uID, xsd:dateTime startTime, xsd:dateTime endTime, tns:WSEntity[] confIDList, tns:WSFilter confFilter )	tns:WSConference[]
<a href="#">GetEndedSessions</a> ( tns:WSEntity[] uID, tns:WSEntity[] confID )	tns:WSConferenceSession[]
<a href="#">GetLogEntries</a> ( tns:WSEntity uID, tns:WSEntity confID, tns:WSParticipant p )	tns:WSConferenceLogEntry[]
<a href="#">GetRunningSession</a> ( tns:WSEntity uID, tns:WSEntity confID, tns:WSFilter confFilter )	tns:WSConferenceSession
<a href="#">GetTemplate</a> ( tns:WSEntity uID, tns:WSEntity confID, tns:WSFilter templateFilter )	tns:WSConferenceTemplate
<a href="#">UpdateConference</a> ( tns:WSEntity uID, tns:WSConference conf )	tns:WSConference
<a href="#">UpdateConferenceSession</a> ( tns:WSEntity uID, tns:WSConferenceSession cs )	tns:WSConferenceSession
<a href="#">UpdateTemplate</a> ( tns:WSEntity uID, tns:WSConferenceTemplate cTemplate )	tns:WSConferenceTemplate

## DeviceService Methods

Provides methods that manage devices, which are supported client software installed on computers, such as Oracle Beehive Integration for Outlook, and mobile devices.

**Table 0–4** *DeviceService*

Method	Return Type
<a href="#">DeleteDevices</a> ( tns:WSEntity[] deviceIDList )	tns:WSResultStatus[]
<a href="#">GetDevicePresence</a> ( tns:WSEntity uID, tns:WSEntity devID, tns:WSFilter devFilter )	tns:WSPresence
<a href="#">GetDevices</a> ( tns:WSEntity uID, tns:WSEntity[] deviceIDList, tns:WSFilter devFilter )	tns:WSDevice[]
<a href="#">UpdateDevice</a> ( tns:WSDevice dev )	tns:WSDevice

## DiscussionForumService Methods

**Table 0–5** *DiscussionForumService*

Method	Return Type
<a href="#">DeleteDiscussionForums</a> ( tns:WSEntity[] forumIDList )	tns:WSResultStatus[]
<a href="#">DeleteMessages</a> ( tns:WSEntity[] dmIDList )	tns:WSResultStatus[]

**Table 0–5 (Cont.) DiscussionForumService**

Method	Return Type
DeleteTopic( tns:WSEntity topicID )	tns:WSResultStatus
GetDiscussionForums( tns:WSEntity wID, tns:WSEntity[] forumIds, tns:WSFilter dfFilter )	tns:WSForum[]
GetLastPost( tns:WSEntity wID, tns:WSEntity fID, tns:WSEntity topicID, tns:WSEntity dmFilter )	tns:WSDiscussionMessage
GetMessages( tns:WSEntity wID, tns:WSEntity fID, tns:WSTopic topic, tns:WSEntity[] dmIDList, tns:WSEntity dmFilter )	tns:WSDiscussionMessage[]
GetTopics( tns:WSEntity forumID, tns:WSEntity[] topicIDs, tns:WSFilter forumFilter )	tns:WSTopic[]
PostMessage( tns:WSEntity parentMessage, tns:WSDiscussionMessage dm )	tns:WSDiscussionMessage
UpdateDiscussionForum( tns:WSEntity parentForum, tns:WSForum forum )	tns:WSForum
UpdateTopic( tns:WSEntity fID, tns:WSTopic topic )	tns:WSTopic

## DocumentService Methods

Provides methods for managing documents in personal or team workspaces.

**Table 0–6 DocumentService**

Method	Return Type
CancelCheckoutDocument( tns:WSEntity docID )	
CheckinDocument( tns:WSEntity docID, xsd:string versionName )	tns:WSVersion
CheckoutDocument( tns:WSEntity docID, xsd:string checkoutComment )	tns:WSVersion
GetDocuments( tns:WSEntity[] docIDList, tns:WSFilter docFilter )	tns:WSDocument[]
GetDocumentsInFolder( tns:WSEntity folderID, tns:WSFilter docFilter )	tns:WSDocument[]
GetContent( tns:WSEntity[] contentIDList )	tns:WSContent[]
GetContentOfDocs( tns:WSEntity[] contentIDList )	tns:WSContent[]
UpdateDocument( tns:WSEntity docID, tns:WSDocument doc, tns:WSContent content, tns:WSConflictResolutionMode conflictResolutionMode )	tns:WSDocument
DeleteDocuments( tns:WSEntity[] docIDList )	tns:WSResultStatus[]

## FolderService Methods

Provides methods for managing messaging and document folders in personal or team workspaces.

**Table 0–7 FolderService**

Method	Return Type
GetFolders( tns:WSEntity uID, tns:WSEntity[] fIDList, tns:WSFilter folderFilter )	tns:WSFolder[]

**Table 0–7 (Cont.) FolderService**

Method	Return Type
<a href="#">GetSubFolders</a> ( tns:WSEntity containerID, tns:WSFilter folderFilter )	tns:WSFolder[]
<a href="#">UpdateFolder</a> ( tns:WSEntity uID, tns:WSFolder folder )	tns:WSFolder
<a href="#">DeleteFolders</a> ( tns:WSEntity[] folderIDList )	tns:WSResultStatus[]

## GeneralArtifactService Methods

Provides methods for managing object metadata and data relationships, as well as provides cross artifact capabilities, searching, and notification capabilities.

**Table 0–8 GeneralArtifactService**

Method	Return Type
<a href="#">CopyArtifact</a> ( tns:WSEntity newParent, tns:WSArtifact artifact )	tns:WSArtifact
<a href="#">MoveArtifact</a> ( tns:WSEntity newParent, tns:WSArtifact artifact )	tns:WSArtifact
<a href="#">GetTags</a> ( tns:WSEntity uID, tns:WSFilter tagFilter )	tns:WSTag[]
<a href="#">UpdateTag</a> ( tns:WSEntity uID, tns:WSTag tag )	tns:WSTag
<a href="#">GetBonds</a> ( tns:WSEntity entityID, tns:WSFilter bondFilter )	tns:WSBond[]
<a href="#">UpdateBond</a> ( tns:WSBond bond )	tns:WSBond
<a href="#">DeleteBonds</a> ( tns:WSEntity[] entityList )	tns:WSResultStatus[]
<a href="#">LoadLinks</a> ( tns:WSEntity entityID, tns:WSProjection linkProjection )	tns:WSLink[]
<a href="#">GetLinksInFolder</a> ( tns:WSEntity folderID, tns:WSFilter linkFilter )	tns:WSLink[]
<a href="#">GetLinksToEntity</a> ( tns:WSEntity entityID, tns:WSFilter linkFilter )	tns:WSLink[]
<a href="#">UpdateLink</a> ( tns:WSLink link )	tns:WSLink
<a href="#">DeleteLinks</a> ( tns:WSEntity[] linkList )	tns:WSResultStatus[]
<a href="#">GetExternalArtifacts</a> ( tns:WSEntity[] externalArtifactIDList )	tns:WSEExternalArtifact[]
<a href="#">UpdateExternalArtifact</a> ( tns:WSEntity parent, tns:WSEExternalArtifact externalArtifact )	tns:WSEExternalArtifact
<a href="#">DeleteExternalArtifacts</a> ( tns:WSEntity[] externalArtifactIDList )	tns:WSResultStatus[]
<a href="#">GetEffectiveLocks</a> ( tns:WSEntity entity )	tns:WSLock[]
<a href="#">UpdateLock</a> ( tns:WSLock lock )	tns:WSLock
<a href="#">DeleteLocks</a> ( tns:WSEntity[] lockList )	tns:WSResultStatus[]
<a href="#">DeleteTags</a> ( tns:WSEntity[] tagIDList )	tns:WSResultStatus[]
<a href="#">UpdateSubscription</a> ( tns:WSEntity eID, tns:WSSubscription sub )	tns:WSSubscription
<a href="#">DeleteSubscriptions</a> ( tns:WSEntity[] subIDList )	tns:WSResultStatus[]
<a href="#">GetSubscription</a> ( tns:WSEntity uID, tns:WSEntity notID, tns:WSFilter subFilter )	tns:WSSubscription

**Table 0–8 (Cont.) GeneralArtifactService**

Method	Return Type
<a href="#">GetSubscriptionList</a> ( tns:WSEntity eID, tns:WSFilter subFilter )	tns:WSSubscription[]
<a href="#">GetSubscriptionTemplates</a> ( xsd:string subscriptionType )	tns:WSSubscriptionTemplate[]
<a href="#">FindArtifacts</a> ( xsd:string searchString, tns:WSFilter filter, xsd:int maxCount, xsd:int start, xsd:int end )	tns:WSSearchResult[]
<a href="#">FindTags</a> ( xsd:string[] tagNameList )	tns:WSTag[]
<a href="#">FindAllTags</a> ( tns:WSEntity uID, tns:WSEntity artifactType )	tns:WSTag[]
<a href="#">FindCategories</a> ( xsd:string[] categoryNameList )	tns:WSCategory[]
<a href="#">FindAllCategories</a> ( tns:WSEntity uID, tns:WSEntity artifactType )	tns:WSCategory[]
<a href="#">FindRelatedArtifacts</a> ( tns:WSEntity artifactID, xsd:string artifactType )	tns:WSArtifact[]
<a href="#">FindAllArtifactsWithTags</a> ( tns:WSEntity uID, xsd:string artifactType, xsd:string tagName )	tns:WSArtifact[]
<a href="#">FindArtifactsModifiedSince</a> ( tns:WSEntity uID, xsd:dateTime date, xsd:string artifactType )	tns:WSArtifact[]
<a href="#">FindArtifactsWithTags</a> ( tns:WSEntity uID, tns:WSEntityType artifactType, xsd:string[] tagNames )	tns:WSArtifact[]
<a href="#">FindArtifactsLargerThan</a> ( xsd:int size )	tns:WSArtifact[]

## GroupService Methods

Provides methods for directly managing personal, workspace, or system groups without having to navigate through scope details such as enterprise, organization and workspace.

**Table 0–9 GroupService**

Method	Return Type
<a href="#">GetGroup</a> ( tns:WSEntity uID, tns:WSProjection projection )	tns:WSGroup
<a href="#">GetGroups</a> ( tns:WSFilter groupFilter )	tns:WSGroup[]
<a href="#">GetGroupsForUser</a> ( tns:WSEntity uID, tns:WSProjection projection )	tns:WSGroup[]
<a href="#">GetEffectiveGroupsForUser</a> ( tns:WSEntity uID, tns:WSProjection projection )	tns:WSGroup[]
<a href="#">GetAllUsersGroup</a> ( tns:WSProjection projection )	tns:WSGroup[]
<a href="#">CreateGroup</a> ( tns:WSEntity uID, tns:WSGroup gID )	tns:WSGroup
<a href="#">DeleteGroup</a> ( tns:WSEntity gID )	tns:WSResultStatus
<a href="#">UpdateGroup</a> ( tns:WSGroup grp )	tns:WSGroup

## MembershipService Methods

Provides methods to retrieve, modify, and delete users and groups in the system. To retrieve the members of a group, specify a FULL projection type with the [GetGroup](#) method. To retrieve the members of a workspace, specify a FULL projection in a filter with the [GetWorkspaces](#) method.



**Table 0–10    MembershipService**

Method	Return Type
<a href="#">GetUsers</a> ( tns:WSEntity[] uIDList, tns:WSFilter userFilter )	tns:WSUser[]
<a href="#">UpdateUser</a> ( tns:WSUser wsUser )	tns:WSUser
<a href="#">DeleteUser</a> ( tns:WSEntity uID )	tns:WSResultStatus
<a href="#">GetDelegatedPrincipals</a> ( tns:WSEntity uID, tns:WSEntity userFilter )	tns:WSUser[]
<a href="#">WhoAmI</a> ()	tns:WSUser

## MessageService Methods

Provides methods for managing personal or team workspace email and instant messages.

**Table 0–11    MessageService**

Method	Return Type
<a href="#">GetContentData</a> ( tns:WSEntity emailID, xsd:base64Binary partIdentifier, xsd:int size )	xsd:base64Binary
<a href="#">GetInbox</a> ( tns:WSFilter msgFilter )	tns:WSMessageBox
<a href="#">GetMessageBoxes</a> ( tns:WSEntity uID, tns:WSEntity[] mdIDList, tns:WSFilter msgFilter )	tns:WSMessageBox[]
<a href="#">UpdateMessageBox</a> ( tns:WSEntity uID, tns:WSEntity parentMBid, tns:WSMessageBox mbox )	tns:WSMessageBox
<a href="#">DeleteMessageBoxes</a> ( tns:WSEntity[] mbIDList )	tns:WSResultStatus[]
<a href="#">GetEmailMessages</a> ( tns:WSEntity uID, tns:WSEntity msgBoxID, tns:WSEntity[] msgIDList, tns:WSFilter msgFilter )	tns:WSMessage[]
<a href="#">GetNewMessages</a> ( tns:WSEntity uID, tns:WSEntity msgBoxID, tns:WSFilter msgFilter )	tns:WSMessage[]
<a href="#">GetUnreadMessages</a> ( tns:WSEntity uID, tns:WSEntity msgBoxID, tns:WSFilter msgFilter )	tns:WSMessageHeader[]
<a href="#">UpdateMessages</a> ( tns:WSMessage[] msgList )	tns:WSMessage[]
<a href="#">SendMessage</a> ( tns:WSEntity uID, tns:WSMessage msg )	tns:WSResultStatus
<a href="#">DeleteEmailMessages</a> ( tns:WSEntity[] msgIDList )	tns:WSResultStatus[]
<a href="#">SendInstantMessage</a> ( tns:WSInstantMessage instantMsg )	tns:WSResultStatus
<a href="#">GetInstantMessage</a> ( xsd:string clientSideID, xsd:string conversationID )	tns:WSInstantMessage[]
<a href="#">SaveDraft</a> ( tns:WSEntity uID, tns:WSMessage msg, tns:WSEntity draftFolderId )	tns:WSResultStatus

## PreferenceService Methods

Provides methods to manage user preferences stored on the server and leveraged by Oracle Beehive clients.

**Table 0–12    PreferenceService**

Method	Return Type
<a href="#">ActivatePresenceProfile</a> ( tns:WSEntity prefProfileId )	tns:WSPreferenceProfile
<a href="#">DeletePreferenceProfiles</a> ( tns:WSEntity[] prefProfIDList )	tns:WSResultStatus[]
<a href="#">DeletePreferences</a> ( tns:WSEntity[] prefProfIDList )	tns:WSResultStatus[]
<a href="#">GetActivePreferenceProfile</a> ( tns:WSEntity uID )	tns:WSPreferenceProfile
<a href="#">GetPreferenceProfiles</a> ( tns:WSEntity uID, tns:WSFilter prefFilter )	tns:WSPreferenceProfile[]
<a href="#">UpdatePreference</a> ( tns:WSEntity prefHolder, tns:WSEntity prefProfile )	tns:WSPreference
<a href="#">UpdatePreferenceProfile</a> ( tns:WSEntity uID, tns:WSPreferenceProfile prefProf )	tns:WSPreferenceProfile

## PresenceService Methods

Provides methods to manage a user's presence information, subscribe to other user's presence, and view the XMPP roster

**Table 0–13    PresenceService**

Method	Return Type
<a href="#">GetPresence</a> ( tns:WSEntity watchable )	tns:WSPresence[]
<a href="#">GetPresences</a> ( tns:WSEntity[] watchables )	tns:WSPresence[]
<a href="#">GetRosterPresences</a> ( )	tns:WSPresence[]
<a href="#">GetSubscriptionRoster</a> ( )	tns:WSSubscriptionRoster[]
<a href="#">RequestPresenceSubscription</a> ( tns:WSEntity watchavke )	:
<a href="#">SetPresence</a> ( tns:WSPresenceContactMethod contactMethod )	:
<a href="#">SetSubscriptionStatus</a> ( tns:WSEntity watchable, tns:WSEntity status )	:
<a href="#">DeletePresenceSubscription</a> ( tns:WSEntity watchable )	:
<a href="#">UpdateActivities</a> ( tns:WSActivity[] activitiesToAdd, tns:WSActivity[] activitiesToRemove )	:

## WorkspaceService Methods

Provides methods for managing personal or team workspaces and their top-level folders.

**Table 0–14    WorkspaceService**

Method	Return Type
<a href="#">GetWorkspaces</a> ( tns:WSEntity uID, xsd:string wspType, tns:WSFilter wspFilter )	tns:WSWorkspace[]
<a href="#">GetWorkspaceTemplates</a> ( tns:WSEntity uID, tns:WSFilter wspTemplateFilter )	tns:WSWorkspaceTemplate[]
<a href="#">GetTrashItems</a> ( tns:WSEntity uID )	tns:WSArtifact[]
<a href="#">UnDeleteItems</a> ( tns:WSEntity[] artifactList )	tns:WSResultStatus[]

**Table 0–14 (Cont.) WorkspaceService**

Method	Return Type
<a href="#">PurgeTrash</a> ( tns:WSEntity[] wIDList )	tns:WSResultStatus[]
<a href="#">DeleteWorkspaces</a> ( tns:WSEntity[] wIDList )	tns:WSResultStatus[]
<a href="#">UpdateWorkspace</a> ( tns:WSEntity uID, tns:WSWorkspace wksp )	tns:WSWorkspace
<a href="#">UpdateWorkspaceMembers</a> ( tns:WSEntity workspace, tns:WSWorkspaceParticipant[] workspaceParticipants )	tns:WSWorkspace

## AddressBookService

Provides methods to manage personal or team workspace address books and their contents, such as contacts and groups.

### GetAddressBooks

**Table 0–15 GetAddressBooks Inputs**

Input Name	Type	Description
uID	tns:WSEntity	Null, a user ID or workspace ID; if null the logged in user is assumed.
abookList	tns:WSEntity[]	An array of address book identifiers. If this parameter is null, then default address book for the uID will be assumed. If this is specified, then specifying uID is not required.
abFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–16 GetAddressBooks Outputs**

Type	Description
tns:WSAddressBook[]	<p>If uID is null, it returns an array of address books for the user currently logged in.</p> <p>If uID denotes a user, then an array of address books for the specified user is returned.</p> <p>If uID denotes a group or workspace, then the address books of that group or workspace are returned. If the abookList parameter is specified, then uID is ignored. The first element is always the default address book.</p>

**Effects** None

**Exceptions** None

### UpdateAddressBook

**Table 0–17 UpdateAddressBook Inputs**

Input Name	Type	Description
uID	tns:WSEntity	User or workspace. The logged in user is assumed if it is null.

**Table 0–17 (Cont.) UpdateAddressBook Inputs**

Input Name	Type	Description
parentAB	tns:WSEntity	ID of an existing address book. If it is null and uID is null as well, then logged in user's default address book is assumed as parentAB. If this is specified then uID is ignored.  If an existing address book is to be moved from one parent to another, then parentAB should be the old (existing) parent with a valid entity ID and the new parent should be specified in the abook parameter.
abook	tns:WSAddressBook	Information about a new or existing address book. If the address book is new, then there is no entity ID provided in abook; otherwise, the entity ID in abook must be a valid identifier of an already existing address book. This parameter cannot be null.

**Table 0–18 UpdateAddressBook Outputs**

Type	Description
tns:WSAddressBook	The new or updated address book is returned. If the address book is new then an entity ID is created and set in the output parameter.

**Effects** If abook is new, for instance, it does not contain an entity ID, then a new address book is created for uID. This new address book is created in the workspace of uID if parentAB is null; otherwise it is created as a sub-address book of parentAB.

If abook denotes an existing address book, for instance, contains a valid entity ID, then the existing address book is updated according to the information provided in abook. If parentAB specifies the old parent, the service will re-parent the existing abook.

**Exceptions** An exception is thrown if the update fails for some reason.

## DeleteAddressBook

**Table 0–19 DeleteAddressBook Inputs**

Input Name	Type	Description
abID	tns:WSEntity	Identifier of an existing address book.

**Table 0–20 DeleteAddressBook Outputs**

Type	Description
tns:WSResultStatus	If the delete operation succeeded, then the result status has an error code of zero, and a non-zero error code and error message otherwise.

**Effects** The address book denoted by abID is deleted from the system.

**Exceptions** None

## GetContact

**Table 0–21** *GetContact Inputs*

Input Name	Type	Description
cID	tns:WSEntity[]	List of valid identifiers for contacts defined in the system.
contactFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–22** *GetContact Outputs*

Type	Description
tns:WSContact[]	Detailed information for the contacts in cID.

**Effects** None.

**Exceptions** If cIDList is null or a contact does not exist in the system an exception is thrown.

## GetAllContacts

**Table 0–23** *GetAllContacts Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user or workspace identifier. If it is null, the logged in user is assumed.
abook	tns:WSEntity	Identifier of a valid address book in the system. If it is null, the default address book is assumed for uID.
contactType	xsd:string	May be null, but if defined, then all contacts of the specified resource type are requested.
contactFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–24** *GetAllContacts Outputs*

Type	Description
tns:WSContact[]	List of contacts for the given user and the address book. If contactType was specified, then all contacts of the given type, for instance, person, group, or resource, are returned.

**Effects** None.

**Exceptions** None

## UpdateContact

**Table 0–25** *UpdateContact Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user or workspace identifier. If it is null the logged in user is assumed.

**Table 0–25 (Cont.) UpdateContact Inputs**

Input Name	Type	Description
abook	tns:WSEntity	Valid address book identifier; If null, default address book for uID is assumed.
contact	tns:WSContact	New or existing contact. If the contact is new, then the entityID attribute of contact must be null; otherwise the entityID attribute must be a valid identifier of an already existing contact.

**Table 0–26 UpdateContact Outputs**

Type	Description
tns:WSContact	The new or updated contact is returned. If the contact is new then an entity ID is created and set in the output parameter.

**Effects** If the contact is new, then it is created in the given address book of the specified user.

If the contact already exists (for instance, the entity ID attribute points to a valid existing contact), then the information for that contact is updated in the given address book of the specified user using the information provided in the contact parameter.

**Exceptions** If the update operation fails then an exception is thrown.

## DeleteContacts

**Table 0–27 DeleteContacts Inputs**

Input Name	Type	Description
cIDList	tns:WSEntity[]	List of valid contact identifiers in the system.

**Table 0–28 DeleteContacts Outputs**

Type	Description
tns:WSResultStatus[]	If the delete operation succeeded, then the result status has an error code of zero, and a non-zero error code and error message otherwise.

**Effects** The contacts in cID are deleted from the system, subject to access control and privileges.

**Exceptions** None

## CalendarService

Provides methods for managing personal or team workspace calendars, task lists, meetings , invitations, tasks, task assignments, reminders and Free/Busy information.

## GetCalendars

**Table 0–29** *GetCalendars Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user or workspace identifier; if null, the logged in user is assumed.
calIDList	tns:WSEntity[]	Calendar for uID; if null the default calendar is assumed.
calFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–30** *GetCalendars Outputs*

Type	Description
tns:WSCalendar[]	Calendar information for the given user/group/workspace and calendar.

**Effects** None.

**Exceptions** None

## GetTaskAssignments

**Table 0–31** *GetTaskAssignments Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user or workspace identifier; if null, the logged in user is assumed.
taskListID	tns:WSEntity	Task list ID.
taskAssignmentIDs	tns:WSEntity[]	List of task assignment IDs to retrieve.

**Table 0–32** *GetTaskAssignments Outputs*

Type	Description
tns:WSTaskAssignment[]	Task assignments for the given user or those that correspond to the given list of task assignment IDs.

**Effects** None.

**Exceptions** None

## UpdateTaskAssignment

**Table 0–33** *UpdateTaskAssignment Inputs*

Input Name	Type	Description
task	tns:WSTaskAssignment	Updated task assignment.

**Table 0–34    UpdateTaskAssignment Outputs**

Type	Description
tns:WSTaskAssignment	Updated task assignment.

**Effects** None.

**Exceptions** None

## DeleteTaskAssignments

**Table 0–35    DeleteTaskAssignments Inputs**

Input Name	Type	Description
taskAssignmen tIDList	tns:WSEntity	List of task assignment IDs to be deleted.

**Table 0–36    DeleteTaskAssignments Outputs**

Type	Description
tns:WSResultStatus[]	If the delete operation succeeded for a particular task assignment, then the result status has an error code of zero, and a non-zero error code and error message otherwise.

**Effects** None.

**Exceptions** None

## UpdateCalendar

**Table 0–37    UpdateCalendar Inputs**

Input Name	Type	Description
uID	tns:WSEntity	Valid user or workspace identifier; if null, the logged in user is assumed.
calendar	tns:WSCalendar	New or updated calendar for uID. If new, the entity ID is not set in calendar; otherwise, it must point to a valid calendar for uID.

**Table 0–38    UpdateCalendar Outputs**

Type	Description
tns:WSCalendar	New or updated calendar for the user, with the entity ID set if new.

**Effects** If the calendar is new (for instance, entity ID is null), it is created for uID. If the calendar already exists (for instance, non-null and valid entity ID), it is updated for uID.

**Exceptions** An exception is thrown if the update fails for some reason.



## DeleteCalendar

**Table 0–39** *DeleteCalendar Inputs*

Input Name	Type	Description
calID	tns:WSEntity	Valid calendar identifier in the system.

**Table 0–40** *DeleteCalendar Outputs*

Type	Description
tns:WSResultStatus	If the delete operation succeeded, then the result status has an error code of zero, and a non-zero error code and error message otherwise.

**Effects** The calendar denoted by calID is deleted from the system, subject to access control.

**Exceptions** None

## GetEvents

**Table 0–41** *GetEvents Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group or workspace identifier; if null, the logged in user is assumed.
calID	tns:WSEntity	Valid calendar identifier for uID; if null, the default calendar is assumed.
eventIDList	tns:WSEntity[]	List of IDs of events.
startTime	xsd:dateTime	Beginning of valid time interval. If null, all events are considered.
endTime	xsd:dateTime	End of valid time interval. If the endTime is null, but the startTime is specified, then the events beginning from the startTime will be returned.
eventFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–42** *GetEvents Outputs*

Type	Description
tns:WSCalendarEvent[]	List of calendar events for the given user/group/workspace, calendar, and time interval.

**Effects** None.

**Exceptions** None

## UpdateEvent

**Table 0–43 UpdateEvent Inputs**

Input Name	Type	Description
uID	tns:WSEntity	Valid user or workspace identifier; if null, the logged in user is assumed.
calID	tns:WSEntity	Valid calendar identifier for uID; if null, the default calendar is assumed.
calEvent	tns:WSCalendarEvent	Specifies a new or existing calendar event for the given user and calendar. If new, the entity ID is not set in calEvent; otherwise, the entity ID points to a valid calendar event for the given user and calendar (uID and calID are ignored in this case).

**Table 0–44 UpdateEvent Outputs**

Type	Description
tns:WSCalendarEvent	New or updated calendar event, with the entity ID set if it was newly created.

**Effects** If calEvent is new (for instance, entity ID is null), it is created in the given calendar for uID. If calEvent already exists, (non-null and valid entity ID), it is updated.

**Exceptions** An exception is thrown if the update fails for some reason.

## DeleteEvents

**Table 0–45 DeleteEvents Inputs**

Input Name	Type	Description
eventIDList	tns:WSEntity[]	Identifies a list of calendar events to be delete from the system.

**Table 0–46 DeleteEvents Outputs**

Type	Description
tns:WSResultStatus[]	If the delete operation succeeded, then the result status has an error code of zero, and a non-zero error code and error message otherwise.

**Effects** The list of calendar events is deleted, subject to access control.

**Exceptions** None

## UpdateEventSeries

**Table 0–47 UpdateEventSeries Inputs**

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group, or workspace identifier; if null, the logged in user is assumed.

**Table 0–47 (Cont.) UpdateEventSeries Inputs**

Input Name	Type	Description
calID	tns:WSEntity	Valid calendar identifier for the user; if null, the default calendar is assumed.
eventSeries	tns:WSEventSeries	Event series to update.

**Table 0–48 UpdateEventSeries Outputs**

Type	Description
tns:WSEventSeries[]	Updated event series.

**Effects** None.

**Exceptions** None

### GetEventSeries

**Table 0–49 GetEventSeries Inputs**

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group, or workspace identifier; if null, the logged in user is assumed.
calID	tns:WSEntity	Valid calendar identifier for the user; if null, the default calendar is assumed.
eventSeriesIDList	tns:WSEntity[]	List of calendar event series in the system.
eventSeriesFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–50 GetEventSeries Outputs**

Type	Description
tns:WSEventSeries[]	List of event series for the given user and calendar.

**Effects** None.

**Exceptions** None

### DeleteEventSeries

**Table 0–51 DeleteEventSeries Inputs**

Input Name	Type	Description
eventSeriesID	tns:WSEntity	Calendar event series identifier existing in the system.

**Table 0–52    DeleteEventSeries Outputs**

Type	Description
tns:WSResultStatus	If the delete operation succeeded, then the result status has an error code of zero, and a non-zero error code and error message otherwise.

**Effects** The calendar event series is deleted, subject to access control.

**Exceptions** None

### DeleteInvitations

**Table 0–53    DeleteInvitations Inputs**

Input Name	Type	Description
invitationIDList	tns:WSEntity[]	List of invitation IDs to be deleted.

**Table 0–54    DeleteInvitations Outputs**

Type	Description
tns:WSResultStatus[]	If the delete operation succeeded, then the result status has an error code of zero, and a non-zero error code and error message otherwise.

**Effects** The list of invitations are deleted, subject to access control.

**Exceptions** None

### DeleteInvitationSeries

**Table 0–55    DeleteInvitationSeries Inputs**

Input Name	Type	Description
invSeriesID	tns:WSEntity	Invitation series ID to be deleted.

**Table 0–56    DeleteInvitationSeries Outputs**

Type	Description
tns:WSResultStatus	If the delete operation succeeded, then the result status has an error code of zero, and a non-zero error code and error message otherwise.

**Effects** The invitation series is deleted, subject to access control.

**Exceptions** None

## GetInvitationSeries

**Table 0–57** *GetInvitationSeries Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group, or workspace identifier; if null, the logged in user is assumed.
calID	tns:WSEntity	Valid calendar identifier for uID; if null, the default calendar is assumed.
invitationSeries IDList	tns:WSEntity[]	List of invitation series identifiers. If this is not null, then uID and calID are ignored.
invitationSeries Filter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–58** *GetInvitationSeries Outputs*

Type	Description
tns:WSCalendarInvitation[]	List of calendar invitation series for the given user/group/workspace and calendar.

**Effects** None.

**Exceptions** None

## GetInvitations

**Table 0–59** *GetInvitations Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group, or workspace identifier; if null, the logged in user is assumed.
calID	tns:WSEntity	Valid calendar identifier for uID; if null, the default calendar is assumed.
invitationIDList	tns:WSEntity[]	List of event identifiers. If this is not null, then uID and calID are ignored.
startTime	xsd:dateTime	Start of a valid time interval. If null, all invitations are considered.
endTime	xsd:dateTime	The end of a valid time interval. If end time is null, but start time is specified, then it will return invitations starting from the start time.
inviteFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–60** *GetInvitations Outputs*

Type	Description
tns:WSCalendarInvitation[]	List of calendar invitations for the given user/group/workspace and calendar and time interval.

**Effects** None.

**Exceptions** None

## UpdateInvitation

**Table 0–61** *UpdateInvitation Inputs*

Input Name	Type	Description
ci	tns:WSCalendarInvitation	Updated calendar invitation for the logged in user.

**Table 0–62** *UpdateInvitation Outputs*

Type	Description
tns:WSCalendarInvitation	Updated calendar event

**Effects** The calendar invitation is updated according to ci.

**Exceptions** None

## UpdateInvitationSeries

**Table 0–63** *UpdateInvitationSeries Inputs*

Input Name	Type	Description
is	tns:WSInvitationSeries	Updated calendar invitation series for the logged in user.

**Table 0–64** *UpdateInvitationSeries Outputs*

Type	Description
tns:WSCalendarInvitation	Updated calendar invitation series

**Effects** The calendar invitation series is updated according to is.

**Exceptions** None

## GetIsBusy

**Table 0–65** *GetIsBusy Inputs*

Input Name	Type	Description
uIDList	tns:WSEntity[]	List of valid users or resources; if null, the logged in user is assumed.
time	xsd:dateTime	Valid time.

**Table 0–66** *GetIsBusy Outputs*

Type	Description
xsd:boolean[]	List of busy (true/false) values corresponding to uIDList.

**Effects** None.

**Exceptions** None

## GetFreeBusy

**Table 0–67** *GetFreeBusy Inputs*

Input Name	Type	Description
uIDList	tns:WSEntity[]	List of valid users/resources in the system; if null, the logged in user is assumed.
startTime	xsd:dateTime	Start of valid time interval. Cannot be null.
endTime	xsd:dateTime	End of valid time interval. Cannot be null.

**Table 0–68** *GetFreeBusy Outputs*

Type	Description
tns:WSActorFreeBusy[]	List of freebusy information for the given users/resources specified in uIDList that are within the specified time interval.

**Effects** None.

**Exceptions** None

## GetReminders

**Table 0–69** *GetReminders Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group or workspace, identifier; if null, the logged in user is assumed.
reminderIDList	tns:WSEntity[]	List of reminder IDs.
remFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–70** *GetReminders Outputs*

Type	Description
tns:WSReminder[]	List of reminders for the user/group/workspace or the list of reminders specified by reminderIDList.

**Effects** None.

**Exceptions** None

## UpdateReminder

**Table 0–71** *UpdateReminder Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group, or workspace; if null; the logged in user is assumed.
rem	tns:WSReminder	New reminder or reminder to update. The uID will be used for new reminders. The reminder will be added to the personal workspace of that user.

**Table 0–72    *UpdateReminder Outputs***

Type	Description
tns:WSReminder	The updated (or newly created) reminder.

**Effects** No effects

**Exceptions** None

## DeleteReminder

**Table 0–73    *DeleteReminder Inputs***

Input Name	Type	Description
remID	tns:WSEntity	ID of Reminder existing in the system

**Table 0–74    *DeleteReminder Outputs***

Type	Description
tns:WSResultStatus	If the delete operation succeeded, then the result status has an error code of zero, a non-zero error code and error message otherwise.

**Effects** The reminder is deleted, subject to access control.

**Exceptions** None

## DeleteTasks

**Table 0–75    *DeleteTasks Inputs***

Input Name	Type	Description
taskIdList	tns:WSEntity[]	Task(s) existing in the system

**Table 0–76    *DeleteTasks Outputs***

Type	Description
tns:WSResultStatus	If the delete operation succeeded, then the result status has an error code of zero, a non-zero error code and error message otherwise.

**Effects** The list of tasks is deleted, subject to access control.

**Exceptions** None

## DeleteTaskLists

**Table 0–77    *DeleteTaskLists Inputs***

Input Name	Type	Description
taskListIds	tns:WSEntity[]	List of task lists existing in the system



**Table 0–78    DeleteTaskLists Outputs**

Type	Description
tns:WSResultStatus	If the delete operation succeeded, then the result status has an error code of zero, a non-zero error code and error message otherwise.

**Effects** The list of task lists is deleted, subject to access control.

**Exceptions** None

## GetTaskLists

**Table 0–79    GetTaskLists Inputs**

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group, or workspace identifier; if null, the logged in user is assumed.
taskListIDs	tns:WSEntity[]	List of task list identifiers for uID. If null, then all task lists is assumed.

**Table 0–80    GetTaskLists Outputs**

Type	Description
tns:WSTaskList[]	Details of the tasklist(s) for the user.

**Effects** None.

**Exceptions** None

## GetTasks

**Table 0–81    GetTasks Inputs**

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group, or workspace identifier; if null, the logged in user is assumed.
taskListID	tns:WSEntity	Specific task list identifier for uID. If null, then default task list is assumed.
taskIDs	tns:WSEntity[]	Specific task identifier(s) for uID to retrieve. If null, then all task are retrieved.

**Table 0–82    GetTasks Outputs**

Type	Description
tns:WSTask	Details of tasks of the specified task list for the user.

**Effects** None.

**Exceptions** None

## UpdateTask

**Table 0–83** *UpdateTask Inputs*

Input Name	Type	Description
task	tns:WSTask	Task to be updated. If the task's taskInfo's artifactID's id field is null, this will be a newly created task.

**Table 0–84** *UpdateTask Outputs*

Type	Description
tns:WSTask	Updated (or created) task.

**Effects** Updates an existing task or creates a new task in the system.

**Exceptions** None

## UpdateTaskList

**Table 0–85** *UpdateTaskList Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group, or workspace identifier; if null, the logged in user is assumed.
taskList	tns:WSTaskList	Task list to be updated. If the taskList's taskListInfo's artifactID's id field is null, this will be a newly created task list.

**Table 0–86** *UpdateTaskList Outputs*

Type	Description
tns:WSTaskList	Updated (or created) task list.

**Effects** Updates an existing task or creates a new task in the system.

**Exceptions** None

## UpdateRecurringEventSeries

**Table 0–87** *UpdateRecurringEventSeries Inputs*

Input Name	Type	Description
uID	tns:WSEntity	None
calID	tns:WSEntity	If specified, this is the calendar where the eventSeries resides. If null, the default calendar for the user is assumed.
eventSeries	tns:WSEventSeries	EventSeries to be updated or created. If eventSeriesId is null, a new EventSeries will be created. Otherwise, this will be an update.

**Table 0–88    *UpdateRecurringEventSeries Outputs***

Type	Description
tns:WSEventSeries	Newly updated or created EventSeries.

**Effects** Creates a new EventSeries in the system.

**Exceptions** None

## ConferenceService

Provides methods that manage Web conferences, including retrieving information about completed and running Web conference sessions.

### DeleteConferences

**Table 0–89    *DeleteConferences Inputs***

Input Name	Type	Description
confIDList	tns:WSEntity[]	List of existing conferences in the system.

**Table 0–90    *DeleteConferences Outputs***

Type	Description
tns:WSResultStatus[]	Status of the delete operation, with error code 0 if the operation succeeded and an error code and error message otherwise.

**Effects** None

**Exceptions** None

### DeleteTemplate

**Table 0–91    *DeleteTemplate Inputs***

Input Name	Type	Description
cTemplateID	tns:WSEntity	Existing template in the system.

**Table 0–92    *DeleteTemplate Outputs***

Type	Description
tns:WSResultStatus[]	Status of the delete operation, with error code 0 if the operation succeeded and an error code and error message otherwise.

**Effects** None

**Exceptions** None

## GetConferences

**Table 0–93** *GetConferences Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group or workspace; if null, the logged in user is assumed.
startTime	xsd:dateTime	Start time of a valid time interval.
endTime	xsd:dateTime	End time of a valid time interval.
confIDList	tns:WSEntity[]	List of valid conferences; if this parameter is not null, uID, startTime, and endTime are ignored.
confFilter	tns:WSFilter	Optional parameter specifying a predicate involving filterable attributes to reduce the size of the returned list.

**Table 0–94** *GetConferences Outputs*

Type	Description
tns:WSConference[]	List of conferences either for confIDList or uID scheduled in the given time interval.

**Effects** None

**Exceptions** None

## GetEndedSessions

**Table 0–95** *GetEndedSessions Inputs*

Input Name	Type	Description
uID	tns:WSEntity[]	User or workspace; if null, the logged in user is assumed.
confID	tns:WSEntity[]	Existing conference; if null, the default conference for uID is assumed. If null, uID is ignored.

**Table 0–96** *GetEndedSessions Outputs*

Type	Description
tns:WSConferenceSession[]	List of ended sessions for the conference.

**Effects** None

**Exceptions** None

## GetLogEntries

**Table 0–97** *GetLogEntries Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user or workspace; if null the logged in user is assumed.
confID	tns:WSEntity	Valid conference; if null, the default conference for uID is assumed.

**Table 0–97 (Cont.) GetLogEntries Inputs**

Input Name	Type	Description
p	tns:WSParticipant	Participant; if null, the logged in user is assumed.

**Table 0–98 GetLogEntries Outputs**

Type	Description
tns:WSConferenceLogEntry[]	Conference log entry for the participant.

**Effects** None

**Exceptions** None

## GetRunningSession

**Table 0–99 GetRunningSession Inputs**

Input Name	Type	Description
uID	tns:WSEntity	User or workspace; if null, the logged in user is assumed.
confID	tns:WSEntity	Existing conference; if null, the default conference for uID is assumed.
confFilter	tns:WSFilter	Optional parameter specifying a predicate involving filterable attributes to reduce the size of the returned list.

**Table 0–100 GetRunningSession Outputs**

Type	Description
tns:WSConferenceSession	Running conference session for the given conference.

**Effects** None

**Exceptions** None

## GetTemplate

**Table 0–101 GetTemplate Inputs**

Input Name	Type	Description
uID	tns:WSEntity	Valid user or workspace; if null, the logged in user is assumed.
confID	tns:WSEntity	Existing conference; if null, the default conference is assumed.
templateFilter	tns:WSFilter	Optional parameter specifying a predicate involving filterable attributes to reduce the size of the returned list.

**Table 0–102 GetTemplate Outputs**

Type	Description
tns:WSConferenceTemplate	Conference template for the given conference.

**Effects** None

**Exceptions** None

## UpdateConference

**Table 0–103** *UpdateConference Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group, or workspace; if null, the logged in user is assumed.
conf	tns:WConference	New or existing conference. If new, the entity ID is null; if the entity ID is not null, it must point to a valid conference for uID. The parameter uID is ignored if the entity ID is not null.

**Table 0–104** *UpdateConference Outputs*

Type	Description
tns:WConference	New or updated conference; the entity ID is set if it is new.

**Effects** If the conference is new, it is created, otherwise, it is updated according to data in conf.

**Exceptions** None

## UpdateConferenceSession

**Table 0–105** *UpdateConferenceSession Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group, or workspace; if null, the logged in user is assumed.
cs	tns:WConferenceSession	New or existing conference session. If new, the entity ID is null; if the entity ID is not null, it must point to a valid conference session for uID. The parameter uID is ignored if the entity ID is not null.

**Table 0–106** *UpdateConferenceSession Outputs*

Type	Description
tns:WConferenceSession	New or updated conference session; the entity ID is set if it is new.

**Effects** If the conference session is new, it is created, otherwise, it is updated according to data in cs.

**Exceptions** None

## UpdateTemplate

**Table 0–107** *UpdateTemplate Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user or workspace; if null, the logged in user is assumed.
cTemplate	tns:WSConferenceTemplate	New or existing conference template; if new, the entity ID is not set.

**Table 0–108** *UpdateTemplate Outputs*

Type	Description
tns:WSConferenceTemplate	New or updated conference template with the entity ID set if new.

**Effects** If the template is new, it is created for the user or workspace, otherwise it is updated.

**Exceptions** None

## DeviceService

Provides methods that manage devices, which are supported client software installed on computers, such as Oracle Beehive Integration for Outlook, and mobile devices.

### DeleteDevices

**Table 0–109** *DeleteDevices Inputs*

Input Name	Type	Description
deviceIDList	tns:WSEntity[]	List of valid device IDs.

**Table 0–110** *DeleteDevices Outputs*

Type	Description
tns:WSResultStatus[]	Status of the delete operation, with error code 0 if the operation succeeded and an error code and error message otherwise.

**Effects** The devices specified by deviceIDList are deleted from the system, subject to access control lists and privileges.

**Exceptions** None

### GetDevicePresence

**Table 0–111** *GetDevicePresence Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user. If null, the logged in user is assumed.
devID	tns:WSEntity	Device ID. If it is null, all devices are considered for the user. If it is specified, then uID is not required.

**Table 0–111 (Cont.) GetDevicePresence Inputs**

Input Name	Type	Description
devFilter	tns:WSFilter	Optional parameter specifying a predicate involving filterable attributes to reduce the size of the returned list.

**Table 0–112 GetDevicePresence Outputs**

Type	Description
tns:WSPresence	List of presence information for the device(s).

**Effects** None

**Exceptions** None

## GetDevices

**Table 0–113 GetDevices Inputs**

Input Name	Type	Description
uID	tns:WSEntity	Valid user. If null, the logged in user is assumed.
deviceIDList	tns:WSEntity[]	Device ID. If null, all devices are considered for the user. If this is specified, then uID is not required.
devFilter	tns:WSFilter	Optional parameter specifying a predicate involving filterable attributes to reduce the size of the returned list.

**Table 0–114 GetDevices Outputs**

Type	Description
tns:WSDevice[]	List of device information for the user.

**Effects** None

**Exceptions** None

## UpdateDevice

**Table 0–115 UpdateDevice Inputs**

Input Name	Type	Description
dev	tns:WSDevice	New or existing device in the system. If it is new, this value is null. otherwise, it is the ID for an existing device.

**Table 0–116 UpdateDevice Outputs**

Type	Description
tns:WSDevice	Newly created or updated device.

**Effects** None

**Exceptions** None



## DiscussionForumService

### DeleteDiscussionForums

**Table 0–117** *DeleteDiscussionForums Inputs*

Input Name	Type	Description
forumIDList	tns:WSEntity[]	List of forums in the system.

**Table 0–118** *DeleteDiscussionForums Outputs*

Type	Description
tns:WSResultStatus[]	Status of the delete operation, with error code 0 if the operation succeeded and an error code and error message otherwise.

**Effects** Deletes the forums specified by forumIDList.

**Exceptions** None

### DeleteMessages

**Table 0–119** *DeleteMessages Inputs*

Input Name	Type	Description
dmIDList	tns:WSEntity[]	List of messages in the system.

**Table 0–120** *DeleteMessages Outputs*

Type	Description
tns:WSResultStatus[]	Status of the delete operation, with error code 0 if the operation succeeded and an error code and error message otherwise.

**Effects** Deletes the messages specified by dmIDList.

**Exceptions** None

### DeleteTopic

**Table 0–121** *DeleteTopic Inputs*

Input Name	Type	Description
topicID	tns:WSEntity	Topic existing in the system.

**Table 0–122** *DeleteTopic Outputs*

Type	Description
tns:WSResultStatus	Status of the delete operation, with error code 0 if the operation succeeded and an error code and error message otherwise.

**Effects** Deletes the topic specified by topicID.

**Exceptions** None

### GetDiscussionForums

**Table 0–123** *GetDiscussionForums Inputs*

Input Name	Type	Description
wID	tns:WSEntity	Workspace.
forumIds	tns:WSEntity[]	List of discussion forums.
dfFilter	tns:WSFilter	Optional parameter specifying a predicate involving filterable attributes to reduce the size of the returned list.

**Table 0–124** *GetDiscussionForums Outputs*

Type	Description
tns:WSForum[]	If wID is specified, list of discussion forums in that workspace. If forumIds is specified, list of discussion forums as specified in that list.

**Effects** None

**Exceptions** None

### GetLastPost

**Table 0–125** *GetLastPost Inputs*

Input Name	Type	Description
wID	tns:WSEntity	Team workspace. If null, fID cannot be null.
fID	tns:WSEntity	Forum. If null, then wID cannot be null. If specified, fID is ignored.
topicID	tns:WSEntity	Topic. If specified, both wID and fID are ignored.
dmFilter	tns:WSEntity	None

**Table 0–126** *GetLastPost Outputs*

Type	Description
tns:WSDiscussionMessage	If topicID is specified, last post in the topic. If fID is specified, last post in the forum. If only wID is specified, last default announcement in the workspace.

**Effects** None

**Exceptions** None

## GetMessages

**Table 0–127** *GetMessages Inputs*

Input Name	Type	Description
wID	tns:WSEntity	Team workspace. If null, then fID cannot be null.
fID	tns:WSEntity	Forum. If null, wID cannot be null. If specified, wID is ignored.
topic	tns:WSTopic	Topic. If null, dmIDList must be specified.
dmIDList	tns:WSEntity[]	List of discussion messages. If specified, all other arguments are ignored.
dmFilter	tns:WSEntity	None

**Table 0–128** *GetMessages Outputs*

Type	Description
tns:WSDiscussionMessage[]	List of discussion messages specified by dmIDList, or those contained in the specified topic or forum. If fID is null, then the default announcements of the specified team workspace are returned.

**Effects** None

**Exceptions** None

## GetTopics

**Table 0–129** *GetTopics Inputs*

Input Name	Type	Description
forumID	tns:WSEntity	ID of an existing forum.
topicIDs	tns:WSEntity[]	List of topic IDs to retrieve.
forumFilter	tns:WSFilter	Optional parameter specifying a predicate involving filterable attributes to reduce the size of the returned list.

**Table 0–130** *GetTopics Outputs*

Type	Description
tns:WSTopic[]	List of topics retrieved from the specified forum or specified topic IDs.

**Effects** None

**Exceptions** None

## PostMessage

**Table 0–131** *PostMessage Inputs*

Input Name	Type	Description
parentMessage	tns:WSEntity	ID of discussion message under which to post a new message. To post a message in the root level of a topic, use the message retrieved from WSTopic.getFirstPost().
dm	tns:WSDiscussionMessage	New discussion message; its entity ID is null.

**Table 0–132** *PostMessage Outputs*

Type	Description
tns:WSDiscussionMessage	The posted message.

**Effects** The message specified by dm is posted under the message specified by parentMessage.

**Exceptions** None

## UpdateDiscussionForum

**Table 0–133** *UpdateDiscussionForum Inputs*

Input Name	Type	Description
parentForum	tns:WSEntity	Forum in which a forum may be created.
forum	tns:WSForum	New or existing topic if entityID is set.

**Table 0–134** *UpdateDiscussionForum Outputs*

Type	Description
tns:WSForum	New or updated forum

**Effects** If forum is a new forum, it is created in parentForum. If parentForum is null, then the forum is created as a default announcement. If forum already exists, it is updated; parentForum is ignored in this case.

**Exceptions** None

## UpdateTopic

**Table 0–135** *UpdateTopic Inputs*

Input Name	Type	Description
fID	tns:WSEntity	Forum.
topic	tns:WSTopic	New or existing topic.

**Table 0–136** *UpdateTopic Outputs*

Type	Description
tns:WSTopic	New or updated topic.

**Effects** If the topic is new, it is created in the specified forum. If the topic exists, it is updated. In this case, fID is ignored.

**Exceptions** None

## DocumentService

Provides methods for managing documents in personal or team workspaces.

### CancelCheckoutDocument

**Table 0–137** *CancelCheckoutDocument Inputs*

Input Name	Type	Description
docID	tns:WSEntity	None

Void return type

**Effects** None.

**Exceptions** None

### CheckinDocument

**Table 0–138** *CheckinDocument Inputs*

Input Name	Type	Description
docID	tns:WSEntity	None
versionName	xsd:string	None

**Table 0–139** *CheckinDocument Outputs*

Type	Description
tns:WSVersion	None

**Effects** None.

**Exceptions** None

### CheckoutDocument

**Table 0–140** *CheckoutDocument Inputs*

Input Name	Type	Description
docID	tns:WSEntity	None
checkoutComment	xsd:string	None

**Table 0–141** *CheckoutDocument Outputs*

Type	Description
tns:WSVersion	None

**Effects** None.

**Exceptions** None

## GetDocuments

**Table 0–142** *GetDocuments Inputs*

Input Name	Type	Description
docIDList	tns:WSEntity[]	List of document identifiers in the system.
docFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–143** *GetDocuments Outputs*

Type	Description
tns:WSDocument[]	List of documents retrieved

**Effects** None.

**Exceptions** None

## GetDocumentsInFolder

**Table 0–144** *GetDocumentsInFolder Inputs*

Input Name	Type	Description
folderID	tns:WSEntity	Folder identifier in the system.
docFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–145** *GetDocumentsInFolder Outputs*

Type	Description
tns:WSDocument[]	List of documents retrieved

**Effects** None.

**Exceptions** None

## GetContent

**Table 0–146** *GetContent Inputs*

Input Name	Type	Description
contentIDList	tns:WSEntity[]	List of document identifiers in the system.

**Table 0–147** *GetContent Outputs*

Type	Description
tns:WSContent[]	Content of documents retrieved

**Effects** None.

**Exceptions** None

## GetContentOfDocs

**Table 0–148** *GetContentOfDocs Inputs*

Input Name	Type	Description
contentIDList	tns:WSEntity[]	None

**Table 0–149** *GetContentOfDocs Outputs*

Type	Description
tns:WSContent[]	None

**Effects** None.

**Exceptions** None

## UpdateDocument

**Table 0–150** *UpdateDocument Inputs*

Input Name	Type	Description
docID	tns:WSEntity	None
doc	tns:WSDocument	New or existing document in the system. If new, the entity ID of this value is null; otherwise, the entity ID points to the document that is to be updated.
content	tns:WSContent	Specifies the content of a document. This is an optional argument and required only if the content of a document needs to be specified during creation or needs to be updated during update
conflictResolutionMode	tns:WSConflictResolutionMode	None

**Table 0–151** *UpdateDocument Outputs*

Type	Description
tns:WSDocument	The newly created or updated document, with the entity ID set if new.

**Effects** If doc is a new document (does not have an entity ID), it is created in the system with the attributes specified in doc. If doc is an existing document, it is updated according to doc.

**Exceptions** None

## DeleteDocuments

**Table 0–152** *DeleteDocuments Inputs*

Input Name	Type	Description
docIDList	tns:WSEntity[]	Document IDs to be deleted

**Table 0–153** *DeleteDocuments Outputs*

Type	Description
tns:WSResultStatus[]	None

**Effects** None

**Exceptions** None

## FolderService

Provides methods for managing messaging and document folders in personal or team workspaces.

### GetFolders

**Table 0–154** *GetFolders Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group or workspace identifier; if null the logged in user is assumed.
fIDList	tns:WSEntity[]	Valid folder ID in the system; if null all folders for uID are considered. If fID is specified then uID is not required.
folderFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–155** *GetFolders Outputs*

Type	Description
tns:WSFolder[]	Folders for the given user, group, or workspace. If fID is not null then the folder type object is returned which contains documents/subfolder info.

**Effects** None.

**Exceptions** None

### GetSubFolders

**Table 0–156** *GetSubFolders Inputs*

Input Name	Type	Description
containerID	tns:WSEntity	Valid folder ID.
folderFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list



**Table 0–157    *GetSubFolders Outputs***

Type	Description
tns:WSFolder[]	Subfolders for the given folder.

**Effects** None.

**Exceptions** None

## UpdateFolder

**Table 0–158    *UpdateFolder Inputs***

Input Name	Type	Description
uID	tns:WSEntity	User or workspace identifier. The logged in user is assumed if it is null.
folder	tns:WSFolder	ID of an existing folder. If it is null and uID is null as well then logged in uIDs default folder is assumed as this ID. If this is specified then uID is ignored. The parentID sub-attribute of the folderInfo attribute of this argument should be non-null.

**Table 0–159    *UpdateFolder Outputs***

Type	Description
tns:WSFolder	The new or updated address book is returned. If the address book is new, then an entity ID is created and set in the output parameter.

**Effects** If folder is new (does not contain an entity ID), then a new folder is created for uID. This new folder is created in the workspace of uID if parentFolderID is null; otherwise it is created as a sub-folder of parentFolderID.

If folder denotes an existing folder (contains a valid entity ID), then the existing folder is updated according to the information provided in folder. If the parentFolderID is specified then the service will then re-parent the existing folder.

**Exceptions** None

## DeleteFolders

**Table 0–160    *DeleteFolders Inputs***

Input Name	Type	Description
folderIDList	tns:WSEntity[]	List of folder identifiers in the system

**Table 0–161    *DeleteFolders Outputs***

Type	Description
tns:WSResultStatus[]	Status of the delete operation

**Effects** The specified folders are deleted

**Exceptions** An exception is thrown if the update fails for some reason.

## GeneralArtifactService

Provides methods for managing object metadata and data relationships, as well as provides cross artifact capabilities, searching, and notification capabilities.

### CopyArtifact

**Table 0–162** *CopyArtifact Inputs*

Input Name	Type	Description
newParent	tns:WSEntity	Valid artifact container in Oracle Beehive
artifact	tns:WSArtifact	Artifact object in Oracle Beehive to be copied

**Table 0–163** *CopyArtifact Outputs*

Type	Description
tns:WSArtifact	New copy of artifact

**Effects** Copies artifact to parent specified by newParent.

**Exceptions** None

### MoveArtifact

**Table 0–164** *MoveArtifact Inputs*

Input Name	Type	Description
newParent	tns:WSEntity	Valid artifact container in Oracle Beehive
artifact	tns:WSArtifact	Artifact to be moved

**Table 0–165** *MoveArtifact Outputs*

Type	Description
tns:WSArtifact	Artifact that was moved

**Effects** Moves artifact to parent specified by newParent.

**Exceptions** None

### GetTags

**Table 0–166** *GetTags Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group or workspace ID. If null the logged in user is assumed.
tagFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–167    *GetTags Outputs***

Type	Description
tns:WSTag[]	Array of tags defined for the user, group or workspace, depending on uID.

**Effects** None.

**Exceptions** None

## UpdateTag

**Table 0–168    *UpdateTag Inputs***

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group or workspace ID. If null the logged in user is assumed.
tag	tns:WSTag	New or existing tag. If new, there is no entity ID set; else, it is the ID of a valid tag.

**Table 0–169    *UpdateTag Outputs***

Type	Description
tns:WSTag	Tag that is newly created or updated.

**Effects** If the tag is new it is created for the user/group/workspace. If the tag already exists, it is updated according to the data provided in tag.

**Exceptions** None

## GetBonds

**Table 0–170    *GetBonds Inputs***

Input Name	Type	Description
entityID	tns:WSEntity	Identifier of a valid entity in the Oracle Beehive system
bondFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–171    *GetBonds Outputs***

Type	Description
tns:WSBond[]	Bonds associated with the entity ID, subject to the filter.

**Effects** None.

**Exceptions** None

## UpdateBond

**Table 0–172** *UpdateBond Inputs*

Input Name	Type	Description
bond	tns:WSBond	Valid bond in the Oracle Beehive system

**Table 0–173** *UpdateBond Outputs*

Type	Description
tns:WSBond	New or updated bond.

**Effects** The bond is updated with the information supplied if the entityID is valid. If the entityID is null, a new bond is created.

**Exceptions** None

## DeleteBonds

**Table 0–174** *DeleteBonds Inputs*

Input Name	Type	Description
entityList	tns:WSEntity[]	List of valid entities

**Table 0–175** *DeleteBonds Outputs*

Type	Description
tns:WSResultStatus[]	If the delete operation succeeded, then the result status has an error code of zero, a non-zero error code and error message otherwise.

**Effects** Entities in entityIDList are deleted, subject to access control and privileges.

**Exceptions** None

## LoadLinks

**Table 0–176** *LoadLinks Inputs*

Input Name	Type	Description
entityID	tns:WSEntity	Valid entity
linkProjection	tns:WSProjection	None

**Table 0–177** *LoadLinks Outputs*

Type	Description
tns:WSLink[]	List of links associated with entityID.

**Effects** None.

**Exceptions** None

## GetLinksInFolder

**Table 0–178** *GetLinksInFolder Inputs*

Input Name	Type	Description
folderID	tns:WSEntity	Folder identifier
linkFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–179** *GetLinksInFolder Outputs*

Type	Description
tns:WSLink[]	List of links contained in the specified folder.

**Effects** None.

**Exceptions** None

## GetLinksToEntity

**Table 0–180** *GetLinksToEntity Inputs*

Input Name	Type	Description
entityID	tns:WSEntity	Entity identifier
linkFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–181** *GetLinksToEntity Outputs*

Type	Description
tns:WSLink[]	List of links to the specified entity.

**Effects** None.

**Exceptions** None

## UpdateLink

**Table 0–182** *UpdateLink Inputs*

Input Name	Type	Description
link	tns:WSLink	Valid link. Its reference and linkInfo.parentID properties must point to valid entities.

**Table 0–183** *UpdateLink Outputs*

Type	Description
tns:WSLink	New or updated link.

**Effects** The link is updated with the information supplied if the entity ID is valid. If the entity ID is null, a new link is created.

**Exceptions** None

## DeleteLinks

**Table 0–184** *DeleteLinks Inputs*

Input Name	Type	Description
linkList	tns:WSEntity[]	List of valid entities

**Table 0–185** *DeleteLinks Outputs*

Type	Description
tns:WSResultStatus[]	If the delete operation succeeded, then the result status has an error code of zero, a non-zero error code and error message otherwise.

**Effects** Deletes the specified links.

**Exceptions** None

## GetExternalArtifacts

**Table 0–186** *GetExternalArtifacts Inputs*

Input Name	Type	Description
externalArtifactIDList	tns:WSEntity[]	List of valid external artifacts

**Table 0–187** *GetExternalArtifacts Outputs*

Type	Description
tns:WSExternalArtifact[]	External artifacts associated with externalArtifactIDList, subject to externalArtFilter.

**Effects** None

**Exceptions** None

## UpdateExternalArtifact

**Table 0–188** *UpdateExternalArtifact Inputs*

Input Name	Type	Description
parent	tns:WSEntity	Valid entity
externalArtifact	tns:WSExternalArtifact	New external artifact if the entity ID of this value does not exist, an existing external artifact otherwise

**Table 0–189** *UpdateExternalArtifact Outputs*

Type	Description
tns:WSExternalArtifact	New or updated external artifact.

**Effects** If externalArtifact is new (does not contain an entity ID) then a new external artifact is created in parent. If externalArtifact exists (contains a valid entity ID) then it is updated according to the provided information in externalArtifact.

**Exceptions** None

## DeleteExternalArtifacts

**Table 0–190** *DeleteExternalArtifacts Inputs*

Input Name	Type	Description
externalArtifact IDList	tns:WSEntity[]	List of external artifacts

**Table 0–191** *DeleteExternalArtifacts Outputs*

Type	Description
tns:WSResultStatus[]	If the delete operation succeeded, then the result status has an error code of zero, a non-zero error code and error message otherwise.

**Effects** Deletes the specified external artifacts.

**Exceptions** None

## GetEffectiveLocks

**Table 0–192** *GetEffectiveLocks Inputs*

Input Name	Type	Description
entity	tns:WSEntity	Specifies the entity on which locks are being queried.

**Table 0–193** *GetEffectiveLocks Outputs*

Type	Description
tns:WSLock[]	List of effective locks on the entity specified in the input. To determine whether a lock is directly applied on an entity, call WSEntity.getLocks(). For example, WSDocument.getLocks().

**Effects** None

**Exceptions** None

## UpdateLock

**Table 0–194** *UpdateLock Inputs*

Input Name	Type	Description
lock	tns:WSLock	Lock identifier that is being modified or created.

**Table 0–195    UpdateLock Outputs**

Type	Description
tns:WSLock	Updated or created lock.

**Effects** None

**Exceptions** None

## DeleteLocks

**Table 0–196    DeleteLocks Inputs**

Input Name	Type	Description
lockList	tns:WSEntity[]	List of locks to be deleted.

**Table 0–197    DeleteLocks Outputs**

Type	Description
tns:WSResultStatus[]	None

**Effects** None

**Exceptions** None

## DeleteTags

**Table 0–198    DeleteTags Inputs**

Input Name	Type	Description
tagIDList	tns:WSEntity[]	List of tags to be deleted.

**Table 0–199    DeleteTags Outputs**

Type	Description
tns:WSResultStatus[]	None

**Effects** None

**Exceptions** None

## UpdateSubscription

**Table 0–200    UpdateSubscription Inputs**

Input Name	Type	Description
eID	tns:WSEntity	ID of any entity that is accessible to the logged in user.
sub	tns:WSSubscription	New or existing subscription for the user. If the subscription is new, there is no entity ID set; otherwise entity ID points to a valid subscription for the entity.



**Table 0–201    UpdateSubscription Outputs**

Type	Description
tns:WSSubscription	The new or updated subscription information is returned. If the subscription is new then an entity ID is created by the system and set in the output parameter.

**Effects** If the subscription is new (has no entity ID set), it is created for the entity. If the subscription already exists (with non-null valid entity ID) it is updated for the entity according to sub.

**Exceptions** None

## DeleteSubscriptions

**Table 0–202    DeleteSubscriptions Inputs**

Input Name	Type	Description
subIDList	tns:WSEntity[]	List of IDs of valid subscriptions to be deleted.

**Table 0–203    DeleteSubscriptions Outputs**

Type	Description
tns:WSResultStatus[]	If the delete operation succeeded, then the result status has an error code of zero, and a non-zero error code and error message otherwise.

**Effects** The subscriptions are deleted from the system, subject to access control and privileges.

**Exceptions** None

## GetSubscription

**Table 0–204    GetSubscription Inputs**

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group, or workspace identifier; if null the logged in user is assumed.
notID	tns:WSEntity	ID of notification in the system. If it is specified then uID is not required.
subFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–205    GetSubscription Outputs**

Type	Description
tns:WSSubscription	The subscription corresponding to the notification.

**Effects** None

**Exceptions** None

## GetSubscriptionList

**Table 0–206** *GetSubscriptionList Inputs*

Input Name	Type	Description
eID	tns:WSEntity	Any entity identifier in the system; cannot be null.
subFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–207** *GetSubscriptionList Outputs*

Type	Description
tns:WSSubscription[]	List of subscriptions for the given entity for the logged in user.

**Effects** None

**Exceptions** None

## GetSubscriptionTemplates

**Table 0–208** *GetSubscriptionTemplates Inputs*

Input Name	Type	Description
subscriptionType	xsd:string	None

**Table 0–209** *GetSubscriptionTemplates Outputs*

Type	Description
tns:WSSubscriptionTemplate[]	None

**Effects** None

**Exceptions** None

## FindArtifacts

**Table 0–210** *FindArtifacts Inputs*

Input Name	Type	Description
searchString	xsd:string	Metadata search string
filter	tns:WSFilter	Predicate, which involves filterable attributes to reduce the size of the returned list
maxCount	xsd:int	Maximum artifacts to return in one call.
start	xsd:int	The first number of a range of integers; lowest value is 1. The method will retrieve this range.
end	xsd:int	The last number of a range of integers; the largest value is maxCount. The method will retrieve this range.

**Table 0–211 FindArtifacts Outputs**

Type	Description
tns:WSSearchResult[]	Items that match the search criteria.

**Effects** None.

**Exceptions** None

## FindTags

**Table 0–212 FindTags Inputs**

Input Name	Type	Description
tagNameList	xsd:string[]	List of tag names that exist in the system for the user

**Table 0–213 FindTags Outputs**

Type	Description
tns:WSTag[]	All items matching the search criteria

**Effects** None.

**Exceptions** None

## FindAllTags

**Table 0–214 FindAllTags Inputs**

Input Name	Type	Description
uID	tns:WSEntity	User, group, or workspace ID
artifactType	tns:WSEntity	Type of artifact

**Table 0–215 FindAllTags Outputs**

Type	Description
tns:WSTag[]	Items matching the search criteria.

**Effects** None.

**Exceptions** None

## FindCategories

**Table 0–216 FindCategories Inputs**

Input Name	Type	Description
categoryNameList	xsd:string[]	List of category names that exist in the system

**Table 0–217** *FindCategories Outputs*

Type	Description
tns:WSCategory[]	All items matching the search criteria

**Effects** None.

**Exceptions** None

### FindAllCategories

**Table 0–218** *FindAllCategories Inputs*

Input Name	Type	Description
uID	tns:WSEntity	User, group, or workspace ID
artifactType	tns:WSEntity	Type of artifact

**Table 0–219** *FindAllCategories Outputs*

Type	Description
tns:WSCategory[]	Items matching the search criteria. If an artifact type is specified, then all categories defined for that type are returned.

**Effects** None.

**Exceptions** None

### FindRelatedArtifacts

**Table 0–220** *FindRelatedArtifacts Inputs*

Input Name	Type	Description
artifactID	tns:WSEntity	Artifact in the system
artifactType	xsd:string	Artifact type

**Table 0–221** *FindRelatedArtifacts Outputs*

Type	Description
tns:WSArtifact[]	Items related to artifactID and of the specified type.

**Effects** None.

**Exceptions** None

### FindAllArtifactsWithTags

**Table 0–222** *FindAllArtifactsWithTags Inputs*

Input Name	Type	Description
uID	tns:WSEntity	None

**Table 0–222 (Cont.) FindAllArtifactsWithTags Inputs**

Input Name	Type	Description
artifactType	xsd:string	None
tagName	xsd:string	None

**Table 0–223 FindAllArtifactsWithTags Outputs**

Type	Description
tns:WSArtifact[]	None

**Effects** No effects

**Exceptions** None

### FindArtifactsModifiedSince

**Table 0–224 FindArtifactsModifiedSince Inputs**

Input Name	Type	Description
uID	tns:WSEntity	User ID
date	xsd:dateTime	Date
artifactType	xsd:string	Artifact type, such as email or document

**Table 0–225 FindArtifactsModifiedSince Outputs**

Type	Description
tns:WSArtifact[]	Artifacts that have been modified since the specified date and have the specified artifact type.

**Effects** None.

**Exceptions** None

### FindArtifactsWithTags

**Table 0–226 FindArtifactsWithTags Inputs**

Input Name	Type	Description
uID	tns:WSEntity	User ID
artifactType	tns:WSEntityType	Artifact type, such as email or document
tagNames	xsd:string[]	Name of tag

**Table 0–227 FindArtifactsWithTags Outputs**

Type	Description
tns:WSArtifact[]	Artifacts with the specified tag.

**Effects** None.

**Exceptions** None

## FindArtifactsLargerThan

**Table 0–228** *FindArtifactsLargerThan Inputs*

Input Name	Type	Description
size	xsd:int	Size of the artifact

**Table 0–229** *FindArtifactsLargerThan Outputs*

Type	Description
tns:WSArtifact[]	Artifacts larger than the specified size.

**Effects** None.

**Exceptions** None

## GroupService

Provides methods for directly managing personal, workspace, or system groups without having to navigate through scope details such as enterprise, organization and workspace.

### GetGroup

**Table 0–230** *GetGroup Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Group identifier.
projection	tns:WSProjection	Specify FULL projection type so that the group returned by this method will also have a list of its members (the returned group's memberIDList property will contain a list of its members)

**Table 0–231** *GetGroup Outputs*

Type	Description
tns:WSGroup	Information about the group specified by uID.

**Effects** None

**Exceptions** None

### GetGroups

**Table 0–232** *GetGroups Inputs*

Input Name	Type	Description
groupFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–233    *GetGroups Outputs***

Type	Description
tns:WSGroup[]	List of groups in the system that satisfy the predicates specified by groupFilter.

**Effects** None

**Exceptions** None

### GetGroupsForUser

**Table 0–234    *GetGroupsForUser Inputs***

Input Name	Type	Description
uID	tns:WSEntity	Identifier of user.
projection	tns:WSProjection	None

**Table 0–235    *GetGroupsForUser Outputs***

Type	Description
tns:WSGroup[]	List of groups to which the user, specified by uID, belongs.

**Effects** None

**Exceptions** None

### GetEffectiveGroupsForUser

**Table 0–236    *GetEffectiveGroupsForUser Inputs***

Input Name	Type	Description
uID	tns:WSEntity	Identifier of user.
projection	tns:WSProjection	None

**Table 0–237    *GetEffectiveGroupsForUser Outputs***

Type	Description
tns:WSGroup[]	List of groups to which the user, specified by uID, directly or indirectly belongs.

**Effects** None

**Exceptions** None

## GetAllUsersGroup

**Table 0–238** *GetAllUsersGroup Inputs*

Input Name	Type	Description
projection	tns:WSProjection	None

**Table 0–239** *GetAllUsersGroup Outputs*

Type	Description
tns:WSGroup[]	The all users group of the enterprise.

**Effects** None

**Exceptions** None

## CreateGroup

**Table 0–240** *CreateGroup Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Workspace identifier.
gID	tns:WSGroup	Group identifier of a new group.

**Table 0–241** *CreateGroup Outputs*

Type	Description
tns:WSGroup	Group that was created in uID.

**Effects** None

**Exceptions** None

## DeleteGroup

**Table 0–242** *DeleteGroup Inputs*

Input Name	Type	Description
gID	tns:WSEntity	Group identifier.

**Table 0–243** *DeleteGroup Outputs*

Type	Description
tns:WSResultStatus	If the delete operation succeeded, the result status returned has an error code 0, and a non-zero error code and error message otherwise.

**Effects** The group identified by gID is deleted from the system.

**Exceptions** None



## UpdateGroup

**Table 0–244** *UpdateGroup Inputs*

Input Name	Type	Description
grp	tns:WSGroup	Detailed information about an existing group. It must contain the entity ID of an existing group.

**Table 0–245** *UpdateGroup Outputs*

Type	Description
tns:WSGroup	The updated group is returned. If the group is new then an entity ID is created and set in the output parameter.

**Effects** None

**Exceptions** An exception is thrown if the update fails for some reason.

## MembershipService

Provides methods to retrieve, modify, and delete users and groups in the system. To retrieve the members of a group, specify a FULL projection type with the GetGroup method. To retrieve the members of a workspace, specify a FULL projection in a filter with the GetWorkspaces method.

### GetUsers

**Table 0–246** *GetUsers Inputs*

Input Name	Type	Description
uIDList	tns:WSEntity[]	List of user IDs. If null, then the filter is applied to retrieve the user list. If uIDList is not null, then the filter is ignored. Both uIDList and userFilter cannot be null.
userFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list.

**Table 0–247** *GetUsers Outputs*

Type	Description
tns:WSUser[]	If uIDList is specified, then the user information of the uIDList elements are returned. Otherwise, the filter is applied to retrieve user information.

**Effects** None.

**Exceptions** None

### UpdateUser

**Table 0–248** *UpdateUser Inputs*

Input Name	Type	Description
wsUser	tns:WSUser	New or updated information about a user in the system.

**Table 0–249    *UpdateUser Outputs***

Type	Description
tns:WSUser	The updated information for a user in the system.

**Effects** Updates the user specified by wsUser. The logged in user must have administrator privileges to update users.

**Exceptions** If the logged in user does not have administrator privileges, an access denied exception is thrown.

## DeleteUser

**Table 0–250    *DeleteUser Inputs***

Input Name	Type	Description
uID	tns:WSEntity	ID of an existing user in the system.

**Table 0–251    *DeleteUser Outputs***

Type	Description
tns:WSResultStatus	Status of the delete operation, with error code 0 if the operation succeeded and an error code and error message otherwise.

**Effects** The user specified by uID is deleted from the system.

**Exceptions** None

## GetDelegatedPrincipals

**Table 0–252    *GetDelegatedPrincipals Inputs***

Input Name	Type	Description
uID	tns:WSEntity	User.
userFilter	tns:WSEntity	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–253    *GetDelegatedPrincipals Outputs***

Type	Description
tns:WSUser[]	List of users that have delegated to the uID.

**Effects** None

**Exceptions** None

## WhoAml

**Table 0–254** WhoAml Outputs

Type	Description
tns:WSUser	Logged in user.

**Effects** None

**Exceptions** None

## MessageService

Provides methods for managing personal or team workspace email and instant messages.

### GetContentData

**Table 0–255** GetContentData Inputs

Input Name	Type	Description
emailID	tns:WSEntity	Valid user; if null, the logged in user is assumed.
partIdentifier	xsd:base64Binary	None
size	xsd:int	None

**Table 0–256** GetContentData Outputs

Type	Description
xsd:base64Binary	None

**Effects** None.

**Exceptions** None

### GetInbox

**Table 0–257** GetInbox Inputs

Input Name	Type	Description
msgFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list. In this filter's projection, you may specify these attributes specific to message boxes: MESSAGEBOX_NEW_MESSAGE_COUNT, MESSAGEBOX_MESSAGE_COUNT, MESSAGEBOX_SUBFOLDER_LIST, and MESSAGEBOX_UNREAD_MESSAGE_COUNT.

**Table 0–258** GetInbox Outputs

Type	Description
tns:WSMessageBox	Retrieves the inbox of the logged in user.

**Effects** None.

**Exceptions** None

## GetMessageBoxes

**Table 0–259** *GetMessageBoxes Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group, or workspace. If null, the logged in user is assumed.
mdIDList	tns:WSEntity[]	List of valid message box for the user, group, or workspace. If null, default message box is assumed. If mbID is specified then uID is ignored.
msgFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list. In this filter's projection, you may specify these attributes specific to message boxes: MESSAGEBOX_NEW_MESSAGE_COUNT, MESSAGEBOX_MESSAGE_COUNT, MESSAGEBOX_SUBFOLDER_LIST, and MESSAGEBOX_UNREAD_MESSAGE_COUNT.

**Table 0–260** *GetMessageBoxes Outputs*

Type	Description
tns:WSMessageBox[]	List of message boxes specified by mbIDList is returned. If mbID is null then the default message box for the user, group or workspace is returned.

**Effects** None.

**Exceptions** None

## UpdateMessageBox

**Table 0–261** *UpdateMessageBox Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group, or workspace. If null, the logged in user is assumed.
parentMBid	tns:WSEntity	Identifier of an existing message box; if null the inbox for uID is assumed. If this specified then uID is ignored.
mbox	tns:WSMessageBox	New or existing message box for the user, group, or workspace; if new, entity ID is not set.

**Table 0–262** *UpdateMessageBox Outputs*

Type	Description
tns:WSMessageBox	New or updated message box.

**Effects** If mbox is new (for instance, does not contain an entity ID) then a new message box is created in the workspace of uID if parentMBid is null; otherwise it is created as a sub-message box of parentMBid. If mbox denotes an existing message box (for

instance, contains a valid entity ID) then mbox is updated according to the information provided in mbox.

**Exceptions** None

## DeleteMessageBoxes

**Table 0–263** *DeleteMessageBoxes Inputs*

Input Name	Type	Description
mbIDList	tns:WSEntity[]	List of valid message boxes in the system.

**Table 0–264** *DeleteMessageBoxes Outputs*

Type	Description
tns:WSResultStatus[]	If the delete operation succeeded, then the result status has an error code of zero, and a non-zero error code and error message otherwise.

**Effects** Message boxes identified by mbIDList are deleted from the system, subject to access control and privileges.

**Exceptions** None

## GetEmailMessages

**Table 0–265** *GetEmailMessages Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group or workspace. If null, the logged in user is assumed.
msgBoxID	tns:WSEntity	ID of the inbox or sub-inbox. If left unspecified, the default inbox is used.
msgIDList	tns:WSEntity[]	List of message identifiers. If null, all messages for uID are considered.
msgFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–266** *GetEmailMessages Outputs*

Type	Description
tns:WSMessage[]	Details of the messages identified by msgIDList

**Effects** None.

**Exceptions** None

## GetNewMessages

**Table 0–267** *GetNewMessages Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group, or workspace; if null, the logged in user is assumed.
msgBoxID	tns:WSEntity	ID of the inbox or sub-inbox. If left unspecified, the default inbox is used.
msgFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–268** *GetNewMessages Outputs*

Type	Description
tns:WSMessage[]	List of new messages for the given user.

**Effects** No effects

**Exceptions** No exceptions thrown

## GetUnreadMessages

**Table 0–269** *GetUnreadMessages Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group or workspace. If null, the logged in user is assumed.
msgBoxID	tns:WSEntity	ID of the inbox or sub-inbox. If left unspecified, the default inbox is used.
msgFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–270** *GetUnreadMessages Outputs*

Type	Description
tns:WSMessageHeader[]	List of unread message headers for the given user.

**Effects** None.

**Exceptions** None

## UpdateMessages

**Table 0–271** *UpdateMessages Inputs*

Input Name	Type	Description
msgList	tns:WSMessage[]	List of messages to be updated

**Table 0-272    *UpdateMessages Outputs***

Type	Description
tns:WSMessage[]	List of updated messages.

**Effects** None.

**Exceptions** No exceptions thrown

## SendMessage

**Table 0-273    *SendMessage Inputs***

Input Name	Type	Description
uID	tns:WSEntity	Entity from which the message is sent, which may be a user or workspace). If it is a workspace, the user must have proper permissions. If null, the logged in user is assumed.
msg	tns:WSMessage	Newly created message to be sent.

**Table 0-274    *SendMessage Outputs***

Type	Description
tns:WSResultStatus	If the send operation succeeded an error code of zero is returned in the status, and a non-zero error code and error message otherwise.

**Effects** Sends the message from the logged in user.

**Exceptions** None

## DeleteEmailMessages

**Table 0-275    *DeleteEmailMessages Inputs***

Input Name	Type	Description
msgIDList	tns:WSEntity[]	Array of email message IDs.

**Table 0-276    *DeleteEmailMessages Outputs***

Type	Description
tns:WSResultStatus[]	If the delete operation succeeded, then the result status has an error code of zero, and a non-zero error code and error message otherwise.

**Effects** The list of given messages are deleted, subject to access control and privileges.

**Exceptions** None

## SendInstantMessage

**Table 0–277** *SendInstantMessage Inputs*

Input Name	Type	Description
instantMsg	tns:WSInstantMessage	Newly created message to be sent.

**Table 0–278** *SendInstantMessage Outputs*

Type	Description
tns:WSResultStatus	If the send operation succeeded an error code of zero is returned in the status, and a non-zero error code and error message otherwise.

**Effects** Sends the message from the logged in user

**Exceptions** None

## GetInstantMessage

**Table 0–279** *GetInstantMessage Inputs*

Input Name	Type	Description
clientSideID	xsd:string	Currently unused
conversationID	xsd:string	Currently unused

**Table 0–280** *GetInstantMessage Outputs*

Type	Description
tns:WSInstantMessage[]	All instant messages, which are sent to the logged-in user in a 30 second time frame.

**Effects** Saves a draft message in the Drafts folder for the uID.

**Exceptions** None

## SaveDraft

**Table 0–281** *SaveDraft Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Existing user identifier in the system.
msg	tns:WSMessage	Message to save.
draftFolderId	tns:WSEntity	The folder in which to save the draft message. If null, a folder named "Drafts" (case ignored) will be used if it exists at the root level of the user's personal workspace. The name of this folder may be specified (as opposed to its ID).



**Table 0–282** *SaveDraft Outputs*

Type	Description
tns:WSResultStatus	None

**Effects** None

**Exceptions** None

## PreferenceService

Provides methods to manage user preferences stored on the server and leveraged by Oracle Beehive clients.

### ActivatePresenceProfile

**Table 0–283** *ActivatePresenceProfile Inputs*

Input Name	Type	Description
prefProfileId	tns:WSEntity	ID of preference profile.

**Table 0–284** *ActivatePresenceProfile Outputs*

Type	Description
tns:WSPreferenceProfile	Preference profile that was activated.

**Effects** None

**Exceptions** None

### DeletePreferenceProfiles

**Table 0–285** *DeletePreferenceProfiles Inputs*

Input Name	Type	Description
prefProfIDList	tns:WSEntity[]	List of preference profiles to delete.

**Table 0–286** *DeletePreferenceProfiles Outputs*

Type	Description
tns:WSResultStatus[]	If the delete operation succeeded, then the result status has an error code of zero, and a non-zero error code and error message otherwise.

**Effects** None.

**Exceptions** None

## DeletePreferences

**Table 0–287** *DeletePreferences Inputs*

Input Name	Type	Description
prefProfIDList	tns:WSEntity[]	List of preferences to delete.

**Table 0–288** *DeletePreferences Outputs*

Type	Description
tns:WSResultStatus[]	If the delete operation succeeded, then the result status has an error code of zero, and a non-zero error code and error message otherwise.

**Effects** None.

**Exceptions** None

## GetActivePreferenceProfile

**Table 0–289** *GetActivePreferenceProfile Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Identifier of an existing user.

**Table 0–290** *GetActivePreferenceProfile Outputs*

Type	Description
tns:WSPreferenceProfile	Active preference profile of the specified user.

**Effects** None

**Exceptions** None

## GetPreferenceProfiles

**Table 0–291** *GetPreferenceProfiles Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Identifier of an existing user.
prefFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

**Table 0–292** *GetPreferenceProfiles Outputs*

Type	Description
tns:WSPreferenceProfile[]	List of preference profiles of the specified user.

**Effects** None

**Exceptions** None

## UpdatePreference

**Table 0–293** *UpdatePreference Inputs*

Input Name	Type	Description
prefHolder	tns:WSEntity	User in the system; if null, the logged in user is assumed.
prefProfile	tns:WSEntity	New or existing preference profile for the given user.

**Table 0–294** *UpdatePreference Outputs*

Type	Description
tns:WSPreference	New or updated preference object. If new, there is no entity ID set in pref. with the output entity ID set if new.

**Effects** If pref is a new preference object, then it will be added to prefProfile. If pref is an existing preference object, it will be modified in prefProfile.

**Exceptions** None

## UpdatePreferenceProfile

**Table 0–295** *UpdatePreferenceProfile Inputs*

Input Name	Type	Description
uID	tns:WSEntity	User identifier in the system; if null, the logged in user is assumed.
prefProf	tns:WSPreferenceProfile	New or existing preference profile for the given user. If new, there is no entity ID set in prefProf; otherwise the entity ID must point to a valid preference profile for the user.

**Table 0–296** *UpdatePreferenceProfile Outputs*

Type	Description
tns:WSPreferenceProfile	The new or updated preference profile is returned, with the output entity ID set if new.

**Effects** If prefProfile is a new profile for the user (for instance, its entity ID is null) then this profile is created for the user. If prefProfile is an existing profile for the user (with non-null and valid entity ID) then this profile is updated for the user.

**Exceptions** An exception is thrown if the update operation fails for some reason.

## PresenceService

Provides methods to manage a user's presence information, subscribe to other user's presence, and view the XMPP roster

## GetPresence

**Table 0–297** *GetPresence Inputs*

Input Name	Type	Description
watchable	tns:WSEntity	Valid user and/or group identifier; if null, the logged in user is assumed.

**Table 0–298** *GetPresence Outputs*

Type	Description
tns:WSPresence[]	Presence info for the user or group.

**Effects** None.

**Exceptions** None

## GetPresences

**Table 0–299** *GetPresences Inputs*

Input Name	Type	Description
watchables	tns:WSEntity[]	List of valid user and/or group identifiers; if null, the logged in user is assumed.

**Table 0–300** *GetPresences Outputs*

Type	Description
tns:WSPresence[]	List of presence info for the users or groups.

**Effects** None.

**Exceptions** None

## GetRosterPresences

**Table 0–301** *GetRosterPresences Outputs*

Type	Description
tns:WSPresence[]	None

**Effects** None.

**Exceptions** None

## GetSubscriptionRoster

**Table 0–302** *GetSubscriptionRoster Outputs*

Type	Description
tns:WSSubscriptionRoster[]	None

**Effects** None.

**Exceptions** None

## RequestPresenceSubscription

**Table 0–303** *RequestPresenceSubscription Inputs*

Input Name	Type	Description
watchavke	tns:WSEntity	None

**Table 0–304** *RequestPresenceSubscription Outputs*

Type	Description
:	Void

**Effects** None

**Exceptions** None

## SetPresence

**Table 0–305** *SetPresence Inputs*

Input Name	Type	Description
contactMethod	tns:WSPresenceContactMethod	None

**Table 0–306** *SetPresence Outputs*

Type	Description
:	Void

**Effects** None

**Exceptions** None

## SetSubscriptionStatus

**Table 0–307** *SetSubscriptionStatus Inputs*

Input Name	Type	Description
watchable	tns:WSEntity	Valid user or group identifier.
status	tns:WSEntity	Status of specified user or group.

**Table 0–308** *SetSubscriptionStatus Outputs*

Type	Description
:	Void

**Effects** None

**Exceptions** None

## DeletePresenceSubscription

**Table 0–309** *DeletePresenceSubscription Inputs*

Input Name	Type	Description
watchable	tns:WSEntity	Valid user or group identifier.

**Table 0–310** *DeletePresenceSubscription Outputs*

Type	Description
:	Void

**Effects** None

**Exceptions** None

## UpdateActivities

**Table 0–311** *UpdateActivities Inputs*

Input Name	Type	Description
activitiesToAdd	tns:WSActivity[]	List of activities to add.
activitiesToRemove	tns:WSActivity[]	List of activities to remove.

**Table 0–312** *UpdateActivities Outputs*

Type	Description
:	Void

**Effects** None

**Exceptions** None

## WorkspaceService

Provides methods for managing personal or team workspaces and their top-level folders.

### GetWorkspaces

**Table 0–313** *GetWorkspaces Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Either a user identifier or a valid workspace ID in the system.
wspType	xsd:string	None

**Table 0–313 (Cont.) GetWorkspaces Inputs**

Input Name	Type	Description
wspFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list. You may specify one or more of the following attributes: isNew, isread, isunread, createdBy_id, createdon_date, modifiedon_date, modifiedby_id, label_id, name, delivered_time, size, to. Specify FULL projection type in this filter so that the workspaces returned by this method will also have a list of its members (each of the returned workspace's memberIDList property will contain a list of its members)

**Table 0–314 GetWorkspaces Outputs**

Type	Description
tns:WSWorkspace[]	Details of the workspace identified by wID, or list of workspaces visible for a uID.

**Effects** None.

**Exceptions** None

## GetWorkspaceTemplates

**Table 0–315 GetWorkspaceTemplates Inputs**

Input Name	Type	Description
uID	tns:WSEntity	Either a user identifier or a valid workspace ID in the system.
wspTemplateFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list.

**Table 0–316 GetWorkspaceTemplates Outputs**

Type	Description
tns:WSWorkspaceTemplate[]	None

**Effects** None

**Exceptions** None

## GetTrashItems

**Table 0–317 GetTrashItems Inputs**

Input Name	Type	Description
uID	tns:WSEntity	User or workspace.

**Table 0–318    *GetTrashItems Outputs***

Type	Description
tns:WSArtifact[]	If uID is null or a user, then a list of artifacts in the Trash folder of a user's personal workspace is returned  If uID is a workspace, then a list of artifacts in the Trash folders of the team workspaces visible to the logged in user is returned.

**Effects** None.

**Exceptions** No exceptions thrown

### UnDeletelItems

**Table 0–319    *UnDeletelItems Inputs***

Input Name	Type	Description
artifactList	tns:WSEntity[]	List of artifacts in the Trash folder

**Table 0–320    *UnDeletelItems Outputs***

Type	Description
tns:WSResultStatus[]	If the undelete operation succeeded, then the result status has an error code of zero, a non-zero error code and error message otherwise.

**Effects** Undeletes the specified items.

**Exceptions** None

### PurgeTrash

**Table 0–321    *PurgeTrash Inputs***

Input Name	Type	Description
wIDList	tns:WSEntity[]	Workspace IDs whose trash will be purged.

**Table 0–322    *PurgeTrash Outputs***

Type	Description
tns:WSResultStatus[]	If the PurgeTrash operation succeeded, then the result status has an error code of zero, a non-zero error code and error message otherwise.

**Effects** Purges the Trash folder

**Exceptions** None



## DeleteWorkspaces

**Table 0–323** *DeleteWorkspaces Inputs*

Input Name	Type	Description
wIDList	tns:WSEntity[]	List of valid workspace identifiers in the system.

**Table 0–324** *DeleteWorkspaces Outputs*

Type	Description
tns:WSResultStatus[]	If the delete operation succeeded, then the result status has an error code of zero, and a non-zero error code and error message otherwise.

**Effects** The workspaces identified by wIDList are deleted, subject to access control and privileges.

**Exceptions** None

## UpdateWorkspace

**Table 0–325** *UpdateWorkspace Inputs*

Input Name	Type	Description
uID	tns:WSEntity	Valid user or group ID in the system. If null the logged in user is assumed.
wksp	tns:WSWorkspace	New or existing workspace. If it is new, then there should be no entity ID set in w; otherwise, the entity ID points to the workspace that is to be updated.

**Table 0–326** *UpdateWorkspace Outputs*

Type	Description
tns:WSWorkspace	The newly created or updated workspace. If new the entity ID is returned in workspace

**Effects** If w is new (has no entity ID set), then this workspace is created for the user or group. If it already exists, it is updated according to the data provided in w.

**Exceptions** None

## UpdateWorkspaceMembers

**Table 0–327** *UpdateWorkspaceMembers Inputs*

Input Name	Type	Description
workspace	tns:WSEntity	Existing workspace.
workspaceParticipants	tns:WSWorkspaceParticipant[]	List of workspace participants

**Table 0–328    UpdateWorkspaceMembers Outputs**

Type	Description
tns:WSWorkspace	Updated workspace with new list of workspace participants.

**Effects** None

**Exceptions** None

## Types

This section lists the types used by Oracle Beehive Web services methods.

### WSActivity

The activity in which the user is engaged. Usually this refers to some calendar activity. Other users who consume this information can decide whether it is appropriate to initiate a conversation or communicate with that user.

**Table 0–329    WSActivity Attributes**

Attribute	Type	Description
startTime	xsd:dateTime	Start of a valid time interval.
note	xsd:string	
id	xsd:string	
activityType	xsd:WSActivityType	CONFERENCE, HOLIDAY, MEAL, tns, ON_THE_PHONE, OTHER, OUT_OF_OFFICE, STEERING, TRAVEL, VACATION
reference	tns:WSEntity	
endTime	xsd:dateTime	End of a valid time interval.

### WSActivityType

Enumeration, may have a value of CONFERENCE, HOLIDAY, MEAL, MEETING, ON\_THE\_PHONE, OTHER, OUT\_OF\_OFFICE, STEERING, TRAVEL, VACATION, or UNSET.

### WSActorFreeBusy

WSActorFreeBusy Description

**Table 0–330    WSActorFreeBusy Attributes**

Attribute	Type	Description
freeBusyIntervals	tns:WSFreeBusyInterval[]	
actor	tns:WSEntity	

### WSAddressBook

Container of contacts. It can contain sub-address books in a hierarchical manner.

**Table 0–331 WSAddressBook Attributes**

Attribute	Type	Description
isDefault	xsd:boolean	Whether it is the default address book
contactInfo	tns:WSEntity[]	List of contacts in the address book
subAddressBookInfo	tns:WSArtifact[]	List of sub-address books
addressBookInfo	tns:WSArtifact	Artifact level information about the address book

**Filterable Parameters**

addressBookInfo, subAbookInfo

**WSArtifact**

The WSArtifact type contains more detailed information about a particular object. The WSArtifact type does not represent a specific entity or object.

**Table 0–332 WSArtifact Attributes**

Attribute	Type	Description
parentID	tns:WSEntity	Identifies the parent of the artifact
owner	tns:WSEntity	No longer used.
categoryApplicationList	tns:WSCategoryApplication[]	
viewerPropertyList	tns:WSProperty[]	
isRead	xsd:boolean	Whether the artifact has been read.
creationDate	xsd:dateTime	Date the object was created in the system.
isNew	xsd:boolean	True if the artifact is new; a new artifact is one that has come into existence since the last time a user viewed the artifact's folder.
propertyList	tns:WSProperty[]	List of properties associated with the entity.
categoryList	tns:WSEntity[]	List of identifiers of categories associated with the artifact.
isLink	xsd:boolean	Whether the artifact is a link.
creator	tns:WSEntity	Entity that created the artifact.
tagIDList	tns:WSEntity[]	List of identifiers of the tags associated with the artifact.
lastModifiedDate	xsd:dateTime	Date and time this artifact was last modified.
artifactID	tns:WSEntity	Entity information for the artifact.
size	xsd:int	Size of the artifact

**Table 0–332 (Cont.) WSArtifact Attributes**

Attribute	Type	Description
hasReminder	xsd:boolean	True if the object has a reminder associated with it.
lastModifiedBy	tns:WSEntity	Entity who last modified this object.

**Filterable Parameters**

This list includes parentID, hasReminder, and creationDate

**WSAttribute****Table 0–333 WSAttribute Attributes**

Attribute	Type	Description
defaultValue	xsd:string	
type	xsd:string	
description	xsd:string	
allowedValues	tns:WSProperty[]	
isMandatory	xsd:boolean	
name	xsd:string	
id	xsd:string	

**WSAttributeApplication****Table 0–334 WSAttributeApplication Attributes**

Attribute	Type	Description
value	xsd:string	
attributeTemplate	tns:WSAttribute	

**WSAttributeDefinition****Table 0–335 WSAttributeDefinition Attributes**

Attribute	Type	Description
maximumValue	xsd:anyType	
value	xsd:anyType	
defaultValue	xsd:anyType	
aggregate	xsd:boolean	
valueType	tns:WSPropertyValueType	
minimumValue	xsd:anyType	
description	xsd:string	

**Table 0–335 (Cont.) WSAttributeDefinition Attributes**

Attribute	Type	Description
allowedValues	tns:WSProperty[]	
maximumInclusive	xsd:boolean	
name	xsd:string	
minimumInclusive	xsd:boolean	
id	xsd:string	

## WSAttributeName

Enumeration, may have a value of ALL\_ATTRIBUTES, PARENT\_ID, CATEGORY\_ID\_LIST, CATEGORY\_APPLICATIONS, TAG\_ID\_LIST, LOCKS, VERSION\_TYPE, CREATOR, LAST\_MODIFIED\_BY, WORKSPACE\_DISCUSSION\_ID\_LIST, WORKSPACE\_DEFAULTS, WORKSPACE\_CALENDAR\_ID\_LIST, WORKSPACE\_INBOX\_ID, WORKSPACE\_ADDRESSBOOK\_ID\_LIST, WORKSPACE\_TASK\_ID\_LIST, WORKSPACE\_SCOPE\_INFO, WORKSPACE\_REMINDER\_LIST, WORKSPACE\_LIBRARY\_ID\_LIST, WORKSPACE\_MEMBERS, WORKSPACE\_MEMBERS\_WITH\_ROLES, FOLDER\_CONTAINED\_ARTIFACTS\_LIST, FOLDER\_SUBFOLDER\_LIST, EMAIL\_PARENT\_ID, EMAIL\_PROPERTY\_LIST, EMAIL\_SPAWNED\_MESSAGE\_ID\_LIST, EMAIL\_RECEIVER\_ID\_LIST, EMAIL\_CCRECEIVER\_ID\_LIST, EMAIL\_BCCRECEIVER\_ID\_LIST, EMAIL\_REPLYTO\_ID, EMAIL\_INREPLYTOMESSAGE\_ID, EMAIL\_FLAGS, EMAIL\_BODY, EMAIL\_BODY\_WITH\_MAIN\_STREAM, EMAIL\_BODY\_WITH\_STREAM, DOCUMENT\_VERSION, DOCUMENT\_VERSION\_HISTORY, DOCUMENT\_CHECKEDOUT\_BY, DOCUMENT\_WITH\_STREAM, BOND\_BONDED\_ELEMENTS, USER\_GROUPS, USER\_WORKSPACES, USER\_CONTACT\_DETAILS, LINK\_REFERENCE, MESSAGEBOX\_NEW\_MESSAGE\_COUNT, MESSAGEBOX\_MESSAGE\_COUNT, MESSAGEBOX\_SUBFOLDER\_LIST, or MESSAGEBOX\_UNREAD\_MESSAGE\_COUNT

## WSAttributeTemplate

**Table 0–336 WSAttributeTemplate Attributes**

Attribute	Type	Description
maximumValue	xsd:anyType	
value	xsd:anyType	
defaultValue	xsd:anyType	
prompted	xsd:boolean	
valueMissing	xsd:boolean	
final	xsd:boolean	
id	xsd:string	
allowedValues	tns:WSProperty[]	
forceDefault	xsd:boolean	
minimumValue	xsd:anyType	
attributeDefinition	tns:WSAttributeDefinition	

**Table 0–336 (Cont.) WSAttributeTemplate Attributes**

Attribute	Type	Description
mandatory	xsd:boolean	
maximumInclusive	xsd:boolean	
minimumInclusive	xsd:boolean	

## WSBond

Defines a relationship. Each entity can be bonded to any number (including zero) of other entities by a variety of bond types.

**Table 0–337 WSBond Attributes**

Attribute	Type	Description
bondType	xsd:string	DISCUSS_THIS, RELATED_MATERIALS, FOLLOW_UP
bondedEntityList	tns:WSEntity[]	List of entities bonded together by the relationship defined by bondType.
bondInfo	tns:WSArtifact	Additional information about the bond.

### Filterable Parameters

All attributes

## WSBuddyList

List of contacts in any instant messaging client. It is a people list with whom a user communicates frequently through instant messaging and chats. The user can also keep track of the presence status of those on the list. It allows third-party and client applications to obtain and add data to it or retrieve data from it.

**Table 0–338 WSBuddyList Attributes**

Attribute	Type	Description
groupIDList	tns:WSEntity[]	List of groups in the buddy list
userIDList	tns:WSEntity[]	List of users in the buddy list

## WSCalendar

Container of time management artifacts. It holds a user's calendar event objects and calendar invitations. Calendar events are created and/or owned by the user. There could be other participants that can be added to the calendar event object. For every participant added to the event, the system automatically creates a calendar invitation object; this invitation object resides in the participant's calendar. The participant then updates the invitation object to update the participation status (for example, accept or reject). Because the invitation object has a reference to the originating calendar event object, a two-way asynchronous synchronization and update of information can happen. A user can have multiple calendars. The default calendar is the one where events get created by default. Also, the default calendar is accessible from the personal workspace of the user.

**Table 0–339 WSCalendar Attributes**

Attribute	Type	Description
timeZone	xsd:string	Time zone of the calendar.
preferenceProfiles	tns:WSPreferenceProfile	Preference profile used for this calendar.
includeInFreeBusy	xsd:boolean	If true, include in calculations of freebusy.
calendarType	xsd:string	Not used
ownerID	tns:WSEntity	Information about the owner of this calendar.
calendarInfo	tns:WSArtifact	Artifact information for the calendar.
properties	tns:WSProperty[]	Properties associated with the calendar. Properties are key=value pairs.
sensitivity	tns:WSSensitivity	Security attribute. A user may manage the access privileges of a calendar with this attribute.

**Filterable Parameters**

All attributes except sensitivity and preferenceProfile

**WSCalendarEvent**

Time slot entries in an user's calendar. They denote, among other things, a time slot for a particular purpose. The purpose could be a meeting, a Web conference, or any other activity. Calendar events created in a particular user's calendar become entities that are owned by that user. The owner can then add any other user or group as participants to the calendar event.

**Table 0–340 WSCalendarEvent Attributes**

Attribute	Type	Description
participantList	tns:WSParticipant[]	Participants invited to attend this event.
participantInvitationList	tns:WSCalendarInvitation[]	The invitations of the participants invited to attend this event.
viewerPrivateProperties	tns:WSProperty[]	List of properties for the viewer of the calendar.
groupPrivateProperties	tns:WSProperty[]	List of properties for the group to which the viewer belongs.
status	xsd:string	Status of the event, for instance TENTATIVE, CANCELED, or CONFIRMED.
propertyList	tns:WSProperty[]	Properties associated with this meeting.
endTime	xsd:dateTime	End of valid time interval.

**Table 0–340 (Cont.) WSCalendarEvent Attributes**

Attribute	Type	Description
iCalPriority	xsd:int	The priority, if this is an iCalendar event.
URI	xsd:AnyURI	URI associated with this meeting.
recurringEventSeries	tns:WSEventSeries	The event series associated with this event.
startTime	xsd:dateTime	Start of valid time interval.
location	tns:WSLocation	Location of the meeting.
eventType	tns:WSCalendarEventType	Either MEETING or DAY EVENT
priority	tns:WSPriority	Priority of calendar event.
duration	xsd:dateTime	Valid time interval.
sensitivity	tns:WSSensitivity	Security attribute. A user may manage the access privileges of a calendar event with this attribute.
eventInfo	tns:WSArtifact	Information about the event object.

**Filterable Parameters**

All attributes except sensitivity, recurrenceRule, and participantInvitationList

**WSCalendarEventParticipant****Table 0–341 WSCalendarEventParticipant Attributes**

Attribute	Type	Description
resolvedName	xsd:string	
addressable	tns:WSEntity	
role	tns:WSParticipantRole	
status	tns:WSCalendarEventParticipantStatus	
resolvedAddress	xsd:string	
external	xsd:boolean	

**WSCalendarEventParticipantStatus**

Enumeration. Value of ACCEPTED, DECLINED, NEEDS\_ACTION, TENTATIVE, or UNSET

**WSCalendarEventStatus**

Enumeration. Value of CANCELLED, CONFIRMED, TENTATIVE, or UNSET



## WSCalendarEventTransparency

Enumeration. Value of DEFAULT\_TRANSPARENCY, OPAQUE, OUT\_OF\_OFFICE, TENTATIVE, TRANSPARANT, or UNSET

## WSCalendarEventType

Enumeration. Value of MEETING, DAY\_EVENT, HOLIDAY, or UNSET

## WSCalendarExceptionToSeries

Enumeration. Value of SERIES\_ONLY, SERIES\_AND\_ALL\_OCCURENCES, SERIES\_AND\_NON\_DIRECTLY\_MODIFIED\_EXCEPTIONS or UNSET

## WSCalendarInvitation

Objects that are created for every calendar event. They are located in the participant's default calendar. These calendar invitation objects are owned by the participant. The participant can update the status or act on the invitation. The calendar invitation contains a reference to the calendar event that was created by the event host.

**Table 0–342**    *WSCalendarInvitation Attributes*

Attribute	Type	Description
inviteePriority	tns:WSPriority	Priority of the invited participant.
isRecurring	xsd:boolean	True if this event reoccurs.
inviteInfo	tns:WSArtifact	Artifact information for the invitation.
recurringEventSeries	tns:WSEventSeries	The event series with which this calendar invitation is associated.
inviteeICalPriority	xsd:int	Priority for the invitee if this is an iCalendar event.
startTime	xsd:dateTime	Start of valid time interval.
source	tns:WSCalendarEvent	The event associated with this invitation.
invitee	tns:WSParticipant	Information about the invited participant.
properties	tns:WSProperty[]	Properties associated with the invitation.
inviteeParticipantStatus	xsd:string	Status of the invited participant, one of ACCEPTED, DECLINED, NEEDS_ACTION, or TENTATIVE
endTime	xsd:dateTime	End of valid time interval.

### Filterable Parameters

All attributes

## WSCalendarSeriesUpdateMode

Enumeration. Value of DIRECTLY\_MODIFIED\_EXCEPTION, INDIRECTLY\_MODIFIED\_EXCEPTION, NON\_EXCEPTION, or UNSET

## WSCategory

Classifies an entity under a structured taxonomy. The metadata administrator primarily creates and makes them available to all users within the enterprise. Categories are hierarchical in nature. The names of the categories within a category hierarchy must be unique. A category holds the category level attributes shared by all entities, classified by the category. It uses one category application object per entity to hold the instance level attributes.

**Table 0–343 WSCategory Attributes**

Attribute	Type	Description
subCategoryList	tns:WSArtifact[]	Artifact information about the subcategories of this category.
categoryInfo	tns:WSArtifact	Artifact information about this category.
attributeTemplateList	tns:WSAttributeTemplate[]	
superCategory	tns:WSArtifact	Artifact information about the category to which this category belongs.
attributeDefinitionList	tns:WSAttributeDefinition[]	

### Filterable Parameters

All attributes

## WSCategoryApplication

**Table 0–344 WSCategoryApplication Attributes**

Attribute	Type	Description
category	tns:WSEntity	
attributeApplicationList	tns:WSAttributeApplication	

## WSCommunity

Set of users, groups, and organizations that share a common set of workspaces and are governed by a common set of policies.

**Table 0–345 WSCommunity Attributes**

Attribute	Type	Description
userList	tns:WSUser[]	Users that belong to community.
groupList	tns:WSGroup[]	Groups that belong to community.
organizationList	tns:WSOrganization[]	Organizations that belong to community.

**Table 0–345 (Cont.) WSCommunity Attributes**

Attribute	Type	Description
workspaceList	tns:WSWorkspace[]	Workspaces shared by members of community.

## WSConference

A conference contains the current status and other information. A conference can be associated with one or more sessions, but only one of the sessions can be running at any moment. Attendees privileges in a conference are specified by an associated conference setting object.

**Table 0–346 WSConference Attributes**

Attribute	Type	Description
confInfo	tns:WSArtifact	Additional information about the conference
URL	xsd:anyURI	
status	xsd:string	

## WSConferenceLogEntry

**Table 0–347 WSConferenceLogEntry Attributes**

Attribute	Type	Description
property	tns:WSProperty	
session	tns:WSConferenceSession	
conference	tns:WSConference	
entryTime	xsd:dateTime	
participant	tns:WSParticipant	

## WSConferenceSession

Contains the transcripts for a conference. A conference session can end with a different result, which is captured in the session ending status. The conference transcript is made available after the end of a conference session. A transcript can contain visual and audio information collected during the session. The transcript document is a media file, identified by the media type of the document, which can be replayed by third-party media players.

**Table 0–348 WSConferenceSession Attributes**

Attribute	Type	Description
conferenceSessionEndStatus	xsd:string	
recording	tns:WSDocument[]	
conference	tns:WSConference	
startTime	xsd:dateTime	
endTime	xsd:dateTime	

## WSConferenceSetting

A conference setting includes a conference configuration (properties) and user rights. Conference settings are comprised of a conference roles section and a conference property list. The conference roles section contains a list of roles created only for the lifetime of the conference instance. The roles are used to assign permissions to participants. A conference property is a special type of property that can hold special values, such as permissions and participants.

**Table 0–349** *WSConferenceSetting Attributes*

Attribute	Type	Description
keys	xsd:string[]	
conferenceRoles	tns:WSRole[]	
confTemplate	tns:WSEntity	
conferenceProperties	tns:WSProperty[]	

## WSConferenceTemplate

Specifies a set of initial conference settings. It can hold a predefined set of groups, properties, and permissions.

**Table 0–350** *WSConferenceTemplate Attributes*

Attribute	Type	Description
conferenceSettings	tns:WSConferenceSetting	
templateInfo	tns:WSArtifact	
attributeList	tns:WSAttribute[]	

## WSConflictResolutionMode

Enumeration, may have a value of OVERWRITE, ABORT, CREATE\_UNIQUE, or VERSION\_OVERWRITE.

## WSContact

An artifact that refers to an addressable entity. It is an entry in an address book.

**Table 0–351** *WSContact Attributes*

Attribute	Type	Description
workPhones	xsd:string[]	List of work phone numbers.
im_usernames	xsd:string[]	Instant Message usernames.
userID	tns:WSEntity	If this contact is an internal user, entity information for that user.
userType	tns:WSUserType	INTERNAL, EXTERNAL, or UNSET
type	tns:WSContactType	One of USER, GROUP, PERSON, or RESOURCE
homePhone	xsd:string	Home phone number for this contact.

**Table 0–351 (Cont.) WSContact Attributes**

Attribute	Type	Description
resource	tns:WSResource	If the Contact is an internal Resource, information for that Resource.
firstName	xsd:string	First name of contact.
lastName	xsd:string	Last name of contact.
info	tns:WSArtifact	Artifact information for this contact.
groupID	tns:WSEntity	If this contact is a Group, entity information for that group.
emailAddresses	xsd:string[]	E-mail address of contact.
cellPhones	xsd:string[]	List of cell phone numbers of contact.

**Table 0–351 (Cont.) WSCContact Attributes**

Attribute	Type	Description
mailingAddresses	xsd:string[]	<p>List of mailing addresses of this contact.</p> <p>Note: In order for Oracle Beehive Extensions for Outlook and Oracle Beehive Integration for Zimbra to recognize mailing addresses added by Oracle Beehive Web Services, it must be in the following format:</p> <p>l1=&lt;address-line-1&gt;?l2=&lt;address-line-2&gt;?box=&lt;post-box-number&gt;?cy=&lt;city&gt;?st=&lt;state&gt;?code=&lt;postal-code&gt;?c=&lt;country&gt;</p> <p>For example, the following is a valid address:</p> <p>l1=1, Main Street</p> <p>The question mark (?) must not be encoded when used as a separator; it must be encoded in all other cases. For example, the following is a valid address:</p> <p>l1=1?l2=Which Street%3F?code=12345</p> <p>However, the following is not a valid address because the question mark appearing as part of the street name "Which Street?" is not encoded:</p> <p>l1=1?l2=Which Street??code=12345</p> <p>The address appears as Home Address in Oracle Beehive Extensions for Outlook and Oracle Beehive Integration for Zimbra.</p> <p>You may not add multiple addresses for a user with Oracle Beehive Web Services.</p>
isBuddy	xsd:boolean	True if this contact is on the logged in user's buddy list.
isAutoCreated	xsd:boolean	True if a contact is auto-created. Auto-created contacts are automatically managed by the server based on the org-chart and cannot be created or deleted by clients. These contacts are mostly read-only. Currently, you can only change the nickname.

**Filterable Parameters**

All attributes except userID, groupID, and isBuddy

**WSContactType**

Enumeration, may have a value of USER, GROUP, RESOURCE, PERSON, or UNSET. A USER contact type represents contacts that are person contacts or bookmarked person contacts. A PERSON contact represents non-bookmarked person contacts.

**WSContent**

Content of an entity, such as a document, notification, or message.

**Table 0–352** *WSContent Attributes*

Attribute	Type	Description
emailContentData	tns:WSMessage	
contentEncoding	xsd:string	Encoding of content.
contentId	xsd:string	
contentType	xsd:string	INLINE, MESSAGE, MULTIPART, REFERENCE
partList	tns:WSContent[]	List of parts of data, of any assortment of media types, which constitutes this content object.
containingDocumentID	tns:WSEntity	Entity, such as document, that consists of this content object.
partIdentifier	xsd:base64Binary	
multiContentType	xsd:string	ALTERNATIVE, MIXED, PARALLEL, RELATED
characterEncoding	xsd:string	Character encoding of content.
contentLanguage	xsd:string	Language of content.
data	xsd:base64Binary	Data of content.
contentDispositionType	xsd:string	
size	xsd:long	Size of document in bytes.
mediaType	xsd:string	Type of media.
name	xsd:string	

**WSDateTime**

WSDateTime Description

**Table 0–353** *WSDateTime Attributes*

Attribute	Type	Description
floating	xsd:boolean	
timestamp	xsd:dateTime	
dateOnly	xsd:boolean	

## WSDevice

A terminal from which a person can interact with the system, either to collaborate, communicate or manage resources. There is no assumption as to how the device is connected to the system.

**Table 0–354** *WSDevice Attributes*

Attribute	Type	Description
device	tns:WSEntity	
deviceClass	xsd:string	
os	xsd:string	
manufacturer	xsd:string	
model	xsd:string	
propertyList	tns:WSProperty[]	
provisioningStatus	tns:WSProvisioningStatus	
processor	xsd:string	
deviceId	xsd:string	

## WSDiscussionMessage

Discussion semantics to the message. The parent of the discussion message is the topic that contains the discussion message.

**Table 0–355** *WSDiscussionMessage Attributes*

Attribute	Type	Description
dmInfo	tns:WSArtifact	Additional information about this discussion message.
annotationList	tns:WSMessageAnnotation[]	
inReplyToMessageID	tns:WSEntity	

### Filterable Parameters

All attributes

## WSDocument

Specific type of self-contained artifact (usually a file) that represents the result of certain end-user applications (such as a word processor)

**Table 0–356** *WSDocument Attributes*

Attribute	Type	Description
documentInfo	tns:WSArtifact	Additional information about this document.
checkoutComments	xsd:string	
versionHistory	tns:WSVersion[]	List of version of document
isCheckedOut	xsd:boolean	
version	tns:WSVersion	Current version of document.
mimeMultipartType	xsd:string	MIME content-type string.



**Table 0–356 (Cont.) WSDocument Attributes**

Attribute	Type	Description
checkedOutBy	tns:WSEntity	
contentID	tns:WSEntity	Content of document.
representativeVersion	tns:WSVersion	Version of document users will typically access.
partDocumentIDList	tns:WSEntity[]	Multipart document's part IDs of simple contents.
locks	tns:WSLock[]	
path	xsd:string	
size	xsd:long	Size of document.
author	tns:WSUser	Author of document.
sensitivity	tns:WSSensitivity	Security attribute. A user may manage the access privileges of a document with this attribute.

## WSEmailParticipant

**Table 0–357 WSEmailParticipant Attributes**

Attribute	Type	Description
resolvedName	xsd:string	
addressable	tns:WSEntity	
resolvedAddress	xsd:string	
external	xsd:boolean	

## WSEnterprise

Top-level scope for all entities.

**Table 0–358 WSEnterprise Attributes**

Attribute	Type	Description
attributeList	tns:WSProperty[]	Attributes associated with enterprise.
timezoneList	xsd:string[]	List of time zones associated with enterprise.

## WSEntity

Contains basic information of an Oracle Beehive object such as its name, its description, and a unique ID to locate it in the system. The entity object for Web services provides this basic information that can be passed to another method to get its details.

**Table 0–359 WSEntity Attributes**

Attribute	Type	Description
type	xsd:string	One of the following: ADDRESSBOOK, BOND, CALENDAR, CALENDAREVENT, CALENDARINVITATION, CATEGORY, CONFERENCE, CONFERENCETEMPLATE, CONTACT, DEVICE, DISCUSSIONMESSAGE, DOCUMENT, EMAILMESSAGE, ENTERPRISE, EXTERNALARTIFACT, EXTERNALUSER, EVENTSERIES, FAXVOICE, FOLDER, FORUM, GROUP, INSTANTMESSAGE, LINK, LOCK, NOTIFICATION, ORGANIZATION, PREFERENCEPROFILE, PREFERENCESET, PRESENCE, REMINDER, RESOURCE, ROLE, SUBSCRIPTION, TAG, TASK, TASKASSIGNMENT, TASKLIST, TOPIC, TRASH, TRASHITEM, USER, WORKSPACE
description	xsd:string	A user-supplied description for the entity. It can be changed.
name	xsd:string	A user-supplied name for the entity. It can be changed.
id	xsd:string	A unique identifier for the entity. Created by the Oracle Beehive system when the entity is created and can never be changed, duplicated, or re-used.

**Filterable Parameters**

An enumeration for the parameters that can be used to filter a return list of this type in Web service methods

**WSEntityType**

Enumeration, may have a value of ADDRESSBOOK, BOND, CALENDAR, CALENDAREVENT, CALENDARINVITATION, CATEGORY, CONFERENCE, CONFERENCETEMPLATE, CONTACT, DEVICE, DISCUSSIONMESSAGE, DOCUMENT, EMAILMESSAGE, ENTERPRISE, EXTERNALARTIFACT, EXTERNALUSER, EVENTSERIES, FAXVOICE, FOLDER, FORUM, GROUP, INSTANTMESSAGE, INVITATIONSERIES, LINK, LOCK, NOTIFICATION, ORGANIZATION, PREFERENCEPROFILE, PREFERENCESET, PRESENCE, REMINDER, RESOURCE, ROLE, ROLEDEFINITION, SUBSCRIPTION, SUBSCRIPTIONLIST, TAG, TASK, TASKASSIGNMENT, TASKLIST, TEMPLATE,

TOPIC, TRASH, TRASHITEM, USER, WORKSPACE, WORKSPACETEMPLATE, or UNSET.

## WSEventSeries

A set of recurring calendar events.

**Table 0–360** *WSEventSeries Attributes*

Attribute	Type	Description
eventSeriesId	tns:WSEntity	Entity information for the EventSeries.
exclusionRule	xsd:string	Rule to exclude particular dates.
events	tns:WSCalendarEvent[]	Events in this series.
recurrenceDate	xsd:dateTime	Recurrence date.
recurrenceRule	xsd:string	iCalendar date time recurrence rule string.
exclusionDate	xsd:dateTime	
eventType	xsd:string	Type of the event, one of TASK, CALENDAR_EVENT, CALENDAR_INVITATION

### Filterable Parameters

All attributes

## WSExternalArtifact

Artifact that is located outside of the system. Oracle Beehive simply records its location (such as a URI) instead of the content itself.

**Table 0–361** *WSExternalArtifact Attributes*

Attribute	Type	Description
artifactInfo	tns:WSArtifact	Additional information about the external artifact.
uri	xsd:AnyURI	Location (URI) of external artifact).
contentEncoding	xsd:string	Content encoding of external artifact.
mediaType	xsd:string	Media type.

### Filterable Parameters

All attributes

## WSFilter

Used to subset a result list returned by a "get" Web service method such as getAllContacts in AddressBookService. A filter consists of two (optional) lists of predicates and a list of sort criteria. The MatchAllList list of predicates returns true if all of its predicates are satisfied. The MatchAnyList list of predicates returns true if any of its predicates are satisfied. If anyAllListRelation is AND, then an entity will be

selected by the filter only if MatchAllList and MatchAnyList return true. If anyAllListRelation is OR, then an entity will be selected by the filter only if MatchAllList or MatchAnyList return true.

**Table 0–362** *WSFilter Attributes*

Attribute	Type	Description
anyAllListRelation	tns:WSLogicalOperator	May have a value of AND or OR. Default is AND.
matchAnyList	tns:WSPredicate[]	Returns true if any of its predicates are satisfied. It is equivalent to a logical OR operator.
projection	tns:WSProjection	Defines the amount of data that is returned with an object.
matchAllList	tns:WSPredicate	Returns true if all of its predicates are satisfied. It is equivalent to a logical AND operator.
sortCriteriaList	tns:WSSortCriteria	Sort criteria for the result list of the filter

## WSFolder

Container of other entities. It may contain other artifacts and sub-folders.

**Table 0–363** *WSFolder Attributes*

Attribute	Type	Description
folderInfo	tns:WSArtifact	Additional information about this folder.
versioningType	xsd:string	Type of versioning used by folder.
containedArtifactList	tns:WSArtifact[]	Artifacts that this folder contains.
preferenceProfileList	tns:WSPreferenceProfile[]	List of preference profiles associated with this folder.
locks	tns:WSLock[]	List of preference profiles associated with this folder.
filterExpression	xsd:string	Expression to filter contents of folder.
subFolderInfoList	tns:WSArtifact[]	Additional information about subfolders of this folder.
sensitivity	tns:WSSensitivity	Security attribute. A user may manage the access privileges of a folder with this attribute.

### Filterable Parameters

All attributes

## WSForum

Collection of other discussion topics and other forums

**Table 0–364** *WSForum Attributes*

Attribute	Type	Description
forumInfo	tns:WSArtifact	Additional information about this forum.
msgCount	xsd:int	Number of messages in forum.

**Filterable Parameters**

All attributes

**WSFreeBusyInterval**

Returned when querying a user's free-busy information for a time period. The tuple of time interval periods and their statuses are stored in this data type and can be accessed through the accessor methods of the appropriate Web service.

**Table 0–365** *WSFreeBusyInterval Attributes*

Attribute	Type	Description
freeBusyType	xsd:string	Indicates whether this interval is a time during which the user or resource is FREE, BUSY, OUT_OF_OFFICE, or other valid status.
startTime	xsd:dateTime	Start of valid time interval.
endTime	xsd:dateTime	End of valid time interval.

**WSGroup**

A defined collection of users or resources.

**Table 0–366** *WSGroup Attributes*

Attribute	Type	Description
subGroupInfo	tns:WSEntity[]	Immutable. List of identifiers for the subgroups of this group
creator	tns:WSEntity	Entity who created the group.
groupInfo	tns:WSEntity	Immutable. Unique identifier for the group.
parentID	tns:WSEntity	
lastModifiedDate	xsd:dateTime	Date and time that the group was last modified.
creationDate	xsd:dateTime	Date and time when the group was created.
memberIDList	tns:WSEntity[]	Immutable. Members (users or groups) who belong to this Group.
lastModifiedBy	tns:WSEntity	Entity that last modified the group.
dynamicQueryString	xsd:string	If not null, then this group is dynamic

**Filterable Parameters**

groupInfo, SubGroupInfo

**WSInstantMessage**

Message that is part of a one-on-one, synchronous and text-based conversation.

**Table 0–367** *WSInstantMessage Attributes*

Attribute	Type	Description
type	xsd:string	BROADCAST, CHAT, FILETRANSFER, SYSTEM
msg	tns:WSMessage	Message text of instant message.
conversationID	xsd:string	Currently unused.
clientsideID	xsd:string	Currently unused.

**Filterable Parameters**

All attributes

**WSInvitationSeries**

WSInvitationSeries Description

**Table 0–368** *WSInvitationSeries Attributes*

Attribute	Type	Description
recurrenceStartDate	tns:WSDateTime	
inviteePriority	tns:WSPriority	
invitationSeriesId	tns:WSEntity	
inviteeEffectiveTransparency	tns:WSCalendarEventTransparency	
invitee	tns:WSEntity	
invitations	tns:WSEntity[]	
originalInclusionRule	xsd:string	
includeTimes	tns:WSDateTime[]	
inviteeParticipantStatus	tns:WSCalendarEventParticipantStatus	
organizer	tns:WSCalendarEventParticipant	
explicitlyModifiedOn	xsd:dateTime	
ICalPriority	xsd:int	
inviteeICalPriority	xsd:int	
URI	xsd:anyURI	
location	tns:WSLocation	
priority	tns:WSPriority	
primary	xsd:boolean	

**Table 0–368 (Cont.) WSInvitationSeries Attributes**

Attribute	Type	Description
attachments	xsd:boolean	
transparency	tns:WSCalendarEventTransparency	
status	tns:WSCalendarEventStatus	
multipleInstances	xsd:boolean	
inviteeTransparency	tns:WSCalendarEventTransparency	
excludeTimes	tns:WSDatetime[]	
participants	tns:WSCalendarEventParticipant[]	
seriesUpdateMode	tns:WSCalendarSeriesUpdateMode	
source	tns:WSEntity	
eventType	tns:WSCalendarEventType	
recurrenceRule	xsd:string	
duration	xsd:long	
multipleParticipants	xsd:boolean	

## WSLink

Symbolic link to an artifact for the purpose of including the same artifact in more than one folder or workspace. Privileges on the referenced artifact are inherited only from the primary folder or workspace of the referenced artifact (not from the link or container of the link).

**Table 0–369 WSLink Attributes**

Attribute	Type	Description
linkInfo	tns:WSArtifact	Additional information about this link.
reference	tns:WSEntity	Reference to entity to which the link refers.

### Filterable Parameters

All attributes

## WSLocale

A locale designates a particular combination of language and region.

**Table 0–370 WSLocale Attributes**

Attribute	Type	Description
language	xsd:string	Language of locale.
script	xsd:string	Script of locale.
region	xsd:string	Region of locale.

**Table 0–370 (Cont.) WSLocale Attributes**

Attribute	Type	Description
variant	xsd:string	Variant of locale.
extension	xsd:string	Extension of locale.
privateUse	xsd:string	
directionality	xsd:string	The direction in which the language of the local is written, such as left-to-right for English or right-to-left for Hebrew.

## WSLocation

**Table 0–371 WSLocation Attributes**

Attribute	Type	Description
timeZone	xsd:dateTime	Time zone of location.
longitude	xsd:float	Longitude of location.
description	xsd:string	Description of location.
propertyList	tns:WSProperty[]	List of properties associated with location.
latitude	xsd:float	Latitude of location.
name	xsd:string	Name of location.
altitude	xsd:float	Altitude of location.

## WSLock

Identifiable object of a certain lock type. A user may hold multiple locks of different types on the same entity. If the entity is a container of artifacts (such as an AddressBook for Contacts), then the lock is applied to all artifacts in the container

**Table 0–372 WSLock Attributes**

Attribute	Type	Description
lockType	xsd:string	ALL, DAV, UserRequest
lockedEntity	tns:WSEntity	Null if unlocking a locked one
creator	tns:WSEntity	Creator of lock.
timeout	xsd:dateTime	Duration of time before lock is removed from entity.
lockID	tns:WSEntity	ID of lock.
createdOn	xsd:dateTime	Date and time lock was created.

### Filterable Parameters

All attributes



## WSLogicalOperator

Enumerated type. May have one of the following values: EQ (equals), NEQ (not equal to), LE (less than or equal to), GE (greater than or equal to), LT (less than), GT (greater than), CONTAINS, DOESNOTCONTAIN, STARTSWITH, or ENDSWITH

## WSMessage

Unit of conversation. It holds the contents in the body attribute.

**Table 0–373** *WSMessage Attributes*

Attribute	Type	Description
receiverIDList	tns:WSEmailParticipant[]	List of those to whom the message is sent.
replyToList	tns:WSEmailParticipant[]	List of those to whom to send a reply.
inReplyToMessageID	xsd:string[]	If the message is a reply to an earlier message, this will refer to that earlier message.
bccReceiverIDList	tns:WSEmailParticipant[]	List of those to whom the message is BCC'd.
senderID	tns:WSEmailParticipant	Who sent the message.
messageType	tns:WSMessageType	One of EMAIL, VOICE, SMS, INSTANT, DISCUSSION, or FAX.
ccReceiverIDList	tns:WSEmailParticipant[]	List of those to whom the message is CC'ed.
msgInfo	tns:WSArtifact	Artifact information about this message.
flags	tns:WSMessageFlag[]	
subject	xsd:string	Subject of the message.
size	xsd:long	Size of the message in bytes.
priority	tns:WSPriority	
body	tns:WSContent	Body of message.
deliveredTime	xsd:dateTime	Date and time at which the message was delivered.

### Filterable Parameters

All attributes

## WSMessageAnnotation

**Table 0–374** *WSMessageAnnotation Attributes*

Attribute	Type	Description
name	xsd:string	Name of message annotation.
attributeList	tns:WSProperty[]	Attributes associated with message annotation.

## WSMessageBox

**Table 0–375** *WSMessageBox Attributes*

Attribute	Type	Description
msgCount	xsd:int	Number of messages in the message box (or inbox).
msgBoxInfo	tns:WSArtifact	Additional information about this message box.
newMsgCount	xsd:int	Number of messages that are new.
msgSubBoxList	tns:WSEntity[]	Subfolders within the message box.
unreadMsgCount	xsd:int	Number of messages that are unread.

### Filterable Parameters

All attributes

## WSMessageFlag

Enumeration, may have a value of ANSWERED, FORWARDED, HIDDEN, MARKED\_DELETE, MARKED\_DRAFT, MARKED\_FOR\_FOLLOWUP, MDN\_PROCESSED, REDIRECTED, or UNSET.

## WSMessageHeader

Holds the header information and meta-data for the message. The sent time of the message is represented by the user's created-on time of the artifact.

**Table 0–376** *WSMessageHeader Attributes*

Attribute	Type	Description
receiverIDList	tns:WSParticipant[]	List of participants that received this message.
optionalMessageID	xsd:string	Currently unused.
msgInfo	tns:WSArtifact	Additional information about this message header.
flags	xsd:string[]	
size	xsd:long	Size of message.
priority	tns:WSPriority	Message priority.
senderID	tns:WSEntity	ID of user who sent the message.
messageType	tns:WSMessageType	One of EMAIL, VOICE, SMS, INSTANT, DISCUSSION, or FAX.
deliveredTime	xsd:dateTime	Date and time when message was delivered.

**Filterable Parameters**

All attributes

**WSMessageType**

One of EMAIL, VOICE, SMS, INSTANT, DISCUSSION, or FAX.

**WSNotificationChannel**

Enumeration, may have a value of DEFAULT\_CHANNEL, EXTERNAL\_EMAIL, IM, INBOX, NOTIFICATION\_LIST, SMS, VOICE, or UNSET.

**WSOrganization**

An organization is a scope under an enterprise. Organizations can be nested to form hierarchies.

**Table 0–377**    *WSOrganization Attributes*

Attribute	Type	Description
scopeInfo	tns:WSScope	Information about this organization.
parent	tns:WSCommunity	Community under which this organization is nested.

**WSParticipant**

Represents someone involved in messaging activity, conferences or calendar events.

**Table 0–378**    *WSParticipant Attributes*

Attribute	Type	Description
participantID	tns:WSEntity	Entity associated with the user, if the participant is a user.
groupID	tns:WSEntity	Group entity associated with this participant, if this participant is a group.
participantStatus	xsd:string	Status of the participant: ACCEPTED, DECLINED, NEEDS_ACTION, TENTATIVE.
externalUserKey	xsd:string	Information for users outside of the system. For example, it may be the email address for an outside user.
role	tns:WSParticipantRole	Role of this Participant: CHAIR, NON-PARTICIPANT, OPTIONAL_PARTICIPANT, REQUIRED_PARTICIPANT.
properties	tns:WSProperty[]	Properties related to this participant.
workspaceID	tns:WSEntity	Workspace associated with this Participant, if the Participant is a Workspace.

## WSParticipantRole

Enumeration. Value of either CHAIR, NON\_PARTICIPANT, OPTIONAL\_PARTICIPANT, REQUIRED\_PARTICIPANT, or UNSET.

## WSPermission

A permission is a privilege granted on an entity.

**Table 0–379 WSPermission Attributes**

Attribute	Type	Description
roleGrantee	tns:WSRole	Role that is being granted the permission
grantPrivileges	xsd:string	Type of access, one of DELETE, DISCOVER, EXECUTE, READ, WRITE
targetWSEntity	tns:WSEntity	Entity to which the user will be granted or denied privileges
userGrantee	tns:WSUser	User that is being granted the permission
denyPrivileges	xsd:string	Type of access, one of DELETE, DISCOVER, EXECUTE, READ, WRITE

## WSPhysicalLocationType

Enumeration. Value of Coordinates, PhysicalSpecification, or UNSET

## WSPredicate

Condition on a certain parameter for the purpose of filtering result sets. A predicate is defined as a triple of parameter (for example, NAME), parameter value (for example, JOE), and operation (for example, CONTAINS).

**Table 0–380 WSPredicate Attributes**

Attribute	Type	Description
paramValue	xsd:string	The value you want to find
paramName	xsd:string	Enumeration consisting of the kind of value you want to filter. One of the following: isnew, isread, isunread, createdby_id, createdon_date, modifiedon_date, modifiedby_id, label_id, name, delivered_time, size, to
operation	tns:WSLogicalOperator	Operation, one of EQ, NEQ, LE, GE, LT, GT, CONTAINS, DOESNOTCONTAIN, STARTSWITH, ENDSWITH

## WSPreference

A user's system, system behavior, UI, presentation, and interactivity customizations are stored in preference objects.

**Table 0–381** *WSPreferenceAttributes*

Attribute	Type	Description
prefProperties	tns:WSPreferenceProperty[]	List of properties associated with the preference object.
prefID	tns:WSEntity	Entity associated with the preference object.
sensitivity	tns:WSSensitivity	Security attribute. A user may manage the access privileges of a preference object with this attribute.

**Filterable Parameters**

All attributes except sensitivity

**WSPreferenceProfile**

A set of preferences, which a user has specified, that are related to a particular type of activity or circumstance. Users can maintain multiple preference profiles and switch between them. For instance, a user may have a regular preference profile and a business travel preference profile. That user may switch profiles depending on the time of usage.

**Table 0–382** *WSPreferenceProfileAttributes*

Attribute	Type	Description
isActive	xsd:boolean	True if the user wants to use the preferences specified by this preference profile.
preferences	tns:WSPreference[]	List of preference the user has specified.
sensitivity	tns:WSSensitivity	Security attribute. A user may manage the access privileges of a preference profile with this attribute.
profileID	tns:WSEntity	Entity associated with this preference profile.

**Filterable Parameters**

All attributes

**WSPreferenceProperty**

A user's system, system behavior, UI, presentation, and interactivity customizations are stored in preference objects.

**Table 0–383** *WSPreferencePropertyAttributes*

Attribute	Type	Description
value	xsd:string	
valueType	xsd:string	
values	xsd:string[]	
overWritable	xsd:boolean	

**Table 0–383 (Cont.) WSPreferenceProperty Attributes**

Attribute	Type	Description
name	xsd:string	

## WSPresence

Information about a user's location, status, availability, connectedness and ability to converse at a given point of time is captured as a user's presence. Since the Oracle Beehive system has access to such information about internal users only, only their presence is made available by Oracle Beehive. For external users, such information is not available.

**Table 0–384 WSPresence Attributes**

Attribute	Type	Description
activities	tns:WSActivity[]	
modificationTimeStamp	xsd:dateTime	Date and time when the presence was last updated.
contactMethods	tns:WSPresenceContactMethod[]	

### Filterable Parameters

All attributes

## WSPresenceContactMethod

**Table 0–385 WSPresenceContactMethod Attributes**

Attribute	Type	Description
statusTimestamp	xsd:dateTime	
creationTimestamp	xsd:dateTime	
reachabilityStatus	tns:WSPresenceStatusType	
URI	xsd:string	
note	xsd:string	
priority	xsd:int	
URIScheme	xsd:string	

## WSPresenceStatusType

Enumeration, may have a value of NOT\_REACHABLE, REACHABLE, CHATTY, AWAY, EXTENDED\_AWAY, DO\_NOT\_DISTURB, or UNSET.

## WSPriority

A ranking of the importance of entities in a set. The higher the priority, the more important the entity. For example, when the entity is a message and the priority is high, it is an urgent message.

**Table 0–386** *WSPriority Attributes*

Attribute	Type	Description
value	xsd:string	Either HIGH, NORMAL, or LOW.

## WSProjection

Defines the amount of data that is returned with an object.

**Table 0–387** *WSProjection Attributes*

Attribute	Type	Description
projection	tns:WSProjectionType	
requiredAttributes	tns:WSAttributeName[]	

## WSProjectionType

Enumeration, may have a value of FULL, META, BASIC, or EMPTY.

## WSProperty

A name-value pair representing some characteristic of an entity.

**Table 0–388** *WSProperty Attributes*

Attribute	Type	Description
value	xsd:string	Value of the property
valueType	tns:WSPropertyValueTypes	Type of value, one of BOM_IDENTIFIABLE, BOOLEAN, COLLECTION_COLLAB_PROPERTY, DATE, DATETIME, FLOAT, DOUBLE, LONG, INTEGER, STRING, TIMESTAMP, URI
values	xsd:string[]	
name	xsd:string	Name of the property

## WSPropertyValueTypes

Enumeration, value of either BOM\_IDENTIFIABLE, BOOLEAN, COLLECTION\_COLLAB\_PROPERTY, DATE, DATETIME, FLOAT, DOUBLE, LONG, INTEGER, STRING, TIMESTAMP, or URI

## WSProvisioningStatus

Enumeration, may have a value of ACTIVE, ENABLED, LOCKED, LOCKDOWN, DISABLED, MARKED\_FOR\_DELETE, DELETE\_IN\_PROGRESS, or UNSET.

## WSQuota

Controls the total amount of space that can be allocated within a scope. Since the deletion of artifacts only moves the artifacts to workspace trash, the deleted artifacts still count against a workspace quota until archived or expunged.

**Table 0–389** *WSQuota Attributes*

Attribute	Type	Description
hardQuota	xsd:int	Once the hard quota is reached, no more create or update (with more bytes) operations will be allowed until the quota breach is resolved
quotaThreshold	xsd:int	Amount quota may be exceeded before a warning is sent.
softQuota	xsd:int	When the soft quota is breached, a warning event will be raised

## WSReminder

Entity that is used to trigger a reminder action at some timed event. It associates a timed trigger on the entity, which can set off an event. A timed trigger can be defined relative to the time of an operation, such as create, update, check-in, or checkout or to a scheduled event (usually a future event), such as the start time of an calendar occurrence, due date of a task, or end time of a conference. A timed trigger can also be defined by absolute time, independent of any operation.

**Table 0–390** *WSReminder Attributes*

Attribute	Type	Description
ruleID	tns:WSEntity	Rule associated with reminder.
relativeTriggerOffset	xsd:dateTime	Duration of time to wait after occurrence of an event (defined by ruleID) to trigger the reminder action.
nextAbsoluteTrigger	xsd:dateTime	An additional date and time to trigger the reminder action.
reminderInfo	tns:WSArtifact	Additional information associated with this reminder.
absoluteTrigger	xsd:dateTime	Date and time to trigger the reminder action.
status	xsd:string	PENDING or PROCESSED
associatedArtifactID	tns:WSEntity	Specifies the associated artifact's entity ID. (Reminders can be associated to any artifact in the system.)

### Filterable Parameters

All attributes

## WSReminderAction



**Table 0–391** *WSReminderAction Attributes*

Attribute	Type	Description
type	tns:WSReminderActionType	
notificationChannels	tns:WSReminderCondition	

## WSReminderActionType

Enumeration, may have a value of NOTIFICATION\_REMINDER\_ACTION, CLIENT\_REMINDER\_ACTION, or UNSET.

## WSReminderCondition

Enumeration, may have a value of NOTALLPARTICIPANTSACCEPTED, NOTALLREQUIREDPARTICIPANTSACCEPTED, or UNSET.

## WSReminderRule

**Table 0–392** *WSReminderRule Attributes*

Attribute	Type	Description
reminderAction	tns:WSReminderAction	
reminderCondition	tns:WSReminderCondition	

## WSResource

An allocatable entity limited in capacity for performing an action or being acted upon (resources are usually not sharable at one time).

**Table 0–393** *WSResource Attributes*

Attribute	Type	Description
externalResourceCapacity	xsd:int	Capacity of external resource, for example, the capacity of a conference room.
isInternalResource	xsd:boolean	True if it is an internal resource, which is a non-user participant in the system.
location	tns:WSLocation	Location of resource.
internalResourceInfo	tns:WSArtifact	If it is an internal resource, artifact information associated with the resource.
resourceId	tns:WSEntity	Entity information for this resource.

## WSResultStatus

Status of executing Web service methods

**Table 0–394** *WSResultStatus Attributes*

Attribute	Type	Description
errorMsg	xsd:string	Error message (if there is any)

**Table 0–394 (Cont.) WSResultStatus Attributes**

Attribute	Type	Description
statusCode	xsd:string	0 indicates success

## WSRole

A role is a named set of permissions

**Table 0–395 WSRole Attributes**

Attribute	Type	Description
memberList	tns:WSGroup[]	Member ID list for the role
permission	tns:WSPermission	List of permissions associated with the role
name	xsd:string	Name of the role

## WSRoleDefinition

**Table 0–396 WSRoleDefinition Attributes**

Attribute	Type	Description
permissions	tns:WSPermission	
roleDefinitionId	tns:WSEntity	

## WSRule

Instruction of the form IF/THEN. A rule is evaluated to either true or false by evaluating the associated condition(s), and appropriate action can be taken depending on the result.

**Table 0–397 WSRule Attributes**

Attribute	Type	Description
conjunctive	xsd:boolean	
rule	xsd:string	
ruleDefinition	tns:WSEntity	
description	xsd:string	
actions	tns:WSEntity[]	Actions to be taken depending on the result of conditions.
conditions	tns:WSEntity[]	List of conditions to evaluate to true or false.

## WSScope

Defines a logical, administrative region. A workspace in Oracle Beehive is a scope.

**Table 0–398** *WSScope Attributes*

Attribute	Type	Description
policyList	tns:WSEntity[]	List of policies that define the scope.
template	tns:WSEntity	
description	xsd:string	Description of scope.
roleList	tns:WSRole[]	List of roles associated with this scope.
quota	tns:WSQuota	Quota of scope.
propertyList	tns:WSProperty[]	List of properties associated with the scope
categoryList	tns:WSCategory[]	Categories associated with scope.

**Filterable Parameters**

All attributes

**WSSearchResult**

The result of a keyword-based search in the Oracle Beehive system

**Table 0–399** *WSSearchResult Attributes*

Attribute	Type	Description
searchResultList	tns:WSSearchResultItem[]	List of items that match the search criteria
hitCount	xsd:int	Number of hits in the result

**WSSearchResultItem**

Denotes an item in a keyword-based search result in Oracle Beehive.

**Table 0–400** *WSSearchResultItem Attributes*

Attribute	Type	Description
searchPropertyList	tns:WSProperty[]	List of properties associated with the artifact
artifact	tns:WSArtifact	An artifact that matches the search criteria
searchResultID	tns:WSEntity	Identifier for the search result

**WSSensitivity**

There are four levels of sensitivity, confidential, private, public, and normal, for any artifact, entity address, or attribute.

Public: Artifacts or attributes marked as Public are accessible to all users in the system, including external users with guest access to the workspace.

Normal: Artifacts or attributes marked as Normal are by default accessible to other users. Other users have access according to the standard access control definition.

**Confidential:** Artifacts or attributes marked as Confidential are not viewable by default to other users in the system unless as denoted through delegation.

**Private:** Artifacts or attributes marked as Private are not accessible to any other users on the system including the designates. For instance, an e-mail message (from a family member), calendar event (a doctor's appointment), and document (a personal resume) can be marked as private and not exposed to any other user unless the owner changes the sensitivity level from Private to some other setting.

**Table 0–401 WSSensitivity Attributes**

Attribute	Type	Description
sensitivityValue	xsd:string	One of CONFIDENTIAL, NORMAL, PRIVATE, PUBLIC

## WSSortCriteria

Sort criteria of a result list. The sort criteria is defined in terms of a parameter (for example, NAME) and the sort order (for example, ASCENDING or DESCENDING).

**Table 0–402 WSSortCriteria Attributes**

Attribute	Type	Description
sortOrder	xsd:string	ASCENDING, DESCENDING
paramName	xsd:string	Name of the field being sorted.

## WSSubscription

User-defined instruction for the Oracle Beehive system to perform a certain action if an event and a given set of conditions occur. It allows a user to prescribe how to automatically react or notify the user when some events occur on an entity.

**Table 0–403 WSSubscription Attributes**

Attribute	Type	Description
conjunctive	xsd:boolean	
attributeValues	tns:WSProperty[]	
subscriptionTemplate	tns:WSEntity	
subscriptionState	xsd:string	Either ENABLED or DISABLED
subscriber	tns:WSEntity	
attachedTo	tns:WSEntity	
subscriptionInfo	tns:WSArtifact	Additional information about the subscription.
subscriptionRules	tns:WSRule[]	
ruleIDList	tns:WSEntity[]	List of rules defined in the subscription
priority	xsd:int	

**Filterable Parameters**

All attribute names

**WSSubscriptionRoster****Table 0–404** *WSSubscriptionRoster Attributes*

Attribute	Type	Description
entries	tns:WSSubscriptionRosterElement[]	

**WSSubscriptionRosterElement****Table 0–405** *WSSubscriptionRosterElement Attributes*

Attribute	Type	Description
watchable	tns:WSEntity	
status	xsd:string	

**WSSubscriptionTemplate**

Template used to create subscriptions.

**Table 0–406** *WSSubscriptionTemplate Attributes*

Attribute	Type	Description
attributeIDList	tns:WSEntity[]	List of attributes of the subscription to be created with this template.
templateInfo	tns:WSArtifact	Additional information about the subscription to be created.
ruleIDList	tns:WSEntity[]	List of rules defined in the subscription to be created.

**WSTag**

Represents a name that can be directly attached to an entity for the purpose of classifying the entity. A single tag can be applied to any number of entities and any entity can support any number of tags.

**Table 0–407** *WSTag Attributes*

Attribute	Type	Description
tagInfo	tns:WSArtifact	Additional information about the tag.

**Filterable Parameters**

All attributes

## WSTask

Represents an action-item or assignment such as an item of work assigned to an individual.

**Table 0–408 WSTask Attributes**

Attribute	Type	Description
taskInfo	tns:WSArtifact	Additional information about this task.
taskStatus	xsd:string	CANCELLED, COMPLETED, IN_PROCESS, NEEDS_ACTION
ICalPriority	xsd:int	Integer in the range zero to nine. Zero is undefined. One is highest priority; nine is lowest.
propertyList	tns:WSProperty[]	Properties associated with this task.
URI	xsd:AnyURI	URI associated with this task.
startTime	xsd:dateTime	Time that task should be started.
location	tns:WSLocation	Location of task.
dueTime	xsd:dateTime	Time that task is due.
taskSeries	tns:WSEventSeries	If this task is associated with an EventSeries this will be populated.
priority	tns:WSPriority	Priority of task.
assignees	xsd:Participant[]	Participants who are assigned this task.

### Filterable Parameters

All attributes

## WSTaskAssignment

An assignment of a task to a participant, with related details such as start and end times, priority, and percent complete.

**Table 0–409 WSTaskAssignment Attributes**

Attribute	Type	Description
taskAssignmentInfo	tns:WSArtifact	
assigneeDueTime	xsd:dateTime	Due time of the task specified by the assignee.
assigneePercentComplete	xsd:int	How much of the assignment is complete, from 0 to 100.
assigneePriority	tns:WSPriority	
taskDueTime	xsd:dateTime	Due time of the task.
assigneeStartTime	xsd:dateTime	Start time of the task specified by the assignee.

**Table 0–409 (Cont.) WSTaskAssignment Attributes**

Attribute	Type	Description
taskStartTime	xsd:dateTime	Start time of the task.
assigneeCompleted	xsd:dateTime	If the assignment is complete, this will contain the date when it was completed.
assigneeICalPriority	xsd:int	Integer in the range zero to nine. A value of zero specifies an undefined priority. A value of one is the highest priority, nine the lowest.
properties	tns:WSProperty[]	
assigneeParticipantStatus	tns:WSTaskParticipantStatus	ACCEPTED, COMPLETED, DECLINED, NEEDS_ACTION, TENTATIVE, WAITING_ON_OTHER
recurringEventSeries	tns:WSEventSeries	Event series associated with this task assignment.
source	xsd:string	ID of the TaskAssignment source.
assignee	tns:WSEntity	Participant assigned to the task specified by this task assignment.
assigneeTimeSpent	xsd:dateTime	Amount of time spent by the assignee on the task.
assigneeTimeAllocated	xsd:dateTime	Amount of time allocated by the assignee to spend on the task.
priority	tns:WSPriority	Priority of the task assignment.

**Filterable Parameters**

All attributes

**WSTaskList**

Container of task management artifacts such as tasks and task assignments.

**Table 0–410 WSTaskList Attributes**

Attribute	Type	Description
assignments	tns:WSTaskAssignment[]	Task assignments associated with this task list.
timeZone	xsd:TimeZone	Time zone of task list.
taskIDs	tns:WSTask[]	Tasks associated with this task list.
owner	tns:WSEntity	Owner of task list.
taskListInfo	tns:WSArtifact	Additional information about this task list.

**Filterable Parameters**

All attributes

**WSTaskParticipantStatus**

Enumeration, may have a value of ACCEPTED, COMPLETED, DECLINED, IN\_PROCESS, NEEDS\_ACTION, WAITING\_ON\_OTHER, or UNSET.

**WSTimedTrigger****Table 0–411** *WSTimedTrigger Attributes*

Attribute	Type	Description
type	tns:WSTimedTriggerType	
time	xsd:long	

**WSTimedTriggerType**

Enumeration, may have a value of ABSOLUTE\_TRIGGER, START\_TRIGGER, END\_TRIGGER, or UNSET.

**WSTopic**

Represents a conversation among forum members; it is structured as a collection of discussions messages. The discussions semantics may impose that the topic messages be sorted in chronological order or threaded by reply.

**Table 0–412** *WSTopic Attributes*

Attribute	Type	Description
topicInfo	tns:WSArtifact	Additional information about this topic.
discussionMessageList	tns:WSEntity[]	List of messages contained in topic.
relatedArtifactID	tns:WSEntity	
msgCount	xsd:int	Number of messages in topic.

**Filterable Parameters**

All attributes

**WSUser**

An entity that can perform actions upon other entities, usually a human. Processes or the services themselves may need to act on the system, but they are considered to be system actors and not users.

**Table 0–413** *WSUser Attributes*

Attribute	Type	Description
systemActor	xsd:boolean	
userWorkspaces	tns:WSEntity[]	Workspaces to which the user belongs



**Table 0–413 (Cont.) WSUser Attributes**

Attribute	Type	Description
userType	tns:WSUserType	Immutable. Type of user.
contactDetails	tns:WSContact	Contact information of the user.
primaryEmailAddress	xsd:string	Primary e-mail address of the user.
userRoles	tns:WSRole[]	Roles that the user has in Oracle Beehive
userIdentifier	xsd:string	Immutable. Entity ID for the user.
userInfo	tns:WSEntity	Immutable. Entity ID for the user.
firstName	xsd:string	First name of the user.
lastName	xsd:string	Last name of the user.

**Filterable Parameters**

userInfo, userType, firstName, lastName, emailAddress

**WSUserType**

WSUserType Description

**Table 0–414 WSUserType Attributes**

Attribute	Type	Description
value	xsd:string	Either INTERNAL, EXTERNAL, or UNSET

**WSVersion**

Defines attributes of a versionable artifact, which represents a specific version of an artifact.

**Table 0–415 WSVersion Attributes**

Attribute	Type	Description
versionedDocID	tns:WSEntity	ID of entity (such as a document) associated with this version object.
predecessorID	tns:WSEntity	Previous (older) version.
successorID	tns:WSEntity	Next (newer) version
description	xsd:string	Description of version.
versionLabel	xsd:string	Version label.
versionNumber	xsd:int	Version number.

**WSVersioningType**

Enumeration, may have a value of AUTO, DISABLED, MANUAL, or UNSET.

## WSWorkspace

Scope that defines a logical scope of work for users. All workspaces must have a name. Each workspace is addressable individually. A workspace may be a personal workspace for an individual user or a group workspace for a set of members.

**Table 0–416** *WSWorkspace Attributes*

Attribute	Type	Description
directoryListed	xsd:boolean	
addressbookIDList	tns:WSEntity[]	List of address books associated with workspace.
defaultCalendar	tns:WSCalendar	
categoryList	tns:WSEntity[]	
tagIDList	tns:WSEntity[]	List of tags associated with workspace.
versioningType	tns:WSVersioningType	Versioning type.
defaultAddressBook	tns:WSAddressBook	
isPublished	xsd:boolean	True if workspace has been published.
scopeInfo	tns:WSScope	Scope information of workspace.
emailAddress	xsd:string	E-mail address associated with workspace.
defaultTaskList	tns:WSTaskList	
workspaceID	tns:WSEntity	Entity denoting the workspace. The name attribute of the entity is not optional and must be present.
lastModifiedBy	tns:WSEntity	User or entity who last modified workspace.
parentID	tns:WSEntity	Identifier of the parent community
discussionForumIDList	tns:WSEntity[]	List of discussion forums associated with workspace.
categoryApplicationList	tns:WSCategoryApplication[]	List of categories associated with workspace.
calendarIDList	tns:WSEntity[]	List of calendars associated with workspace.
inboxID	tns:WSEntity	Workspace inbox.
workspaceType	xsd:WSWorkspaceType	Type of the workspace.
creationDate	xsd:dateTime	Date and time workspace was created.
tasklistIDList	tns:WSEntity[]	List of tasklist(s associated with workspace.
memberIDList	tns:WSWorkspaceParticipant[]	List of tasklist(s associated with workspace.
defaultForum	tns:WSForum	
creator	tns:WSEntity	

**Table 0–416 (Cont.) WSWorkspace Attributes**

Attribute	Type	Description
lastModifiedDate	xsd:dateTime	Last date and time workspace was modified.
locks	tns:WSLock[]	List of reminders associated with workspace.
reminderList	tns:WSEntity[]	List of reminders associated with workspace.
libraryIDList	tns:WSEntity[]	Folder for artifacts

**Filterable Parameters**

All attributes

**WSWorkspaceParticipant****Table 0–417 WSWorkspaceParticipant Attributes**

Attribute	Type	Description
roleDefinitions	tns:WSRoleDefinition[]	
roles	tns:WSRole	
member	tns:WSEntity	

**WSWorkspaceTemplate**

Defines a workspace

**Table 0–418 WSWorkspaceTemplate Attributes**

Attribute	Type	Description
workspaceTemplateType	xsd:string	
contactInfo	xsd:string	
workspaceTemplateInfo	tns:WSArtifact	
authorCreationTime	xsd:dateTime	
copyrightInfo	xsd:string	
templateID	xsd:string	
author	xsd:string	
transportableFormat	xsd:base64Binary	



---

# Oracle Beehive Business Reporting

This chapter describes the business reporting features of Oracle Beehive. It includes the following sections:

- [Overview of Business Views](#)
- [Types of Views](#)
- [Workspace Service Views](#)
- [Device Management Service Views](#)
- [Collaboration Service Views](#)
- [Message Delivery Service Views](#)
- [Mobility Service Views](#)
- [User Directory Service Views](#)
- [Creating Schema with Access Only to Oracle Beehive Views](#)

## Overview of Business Views

Oracle Database is the information store for Oracle Beehive and, in addition to other data, contains a set of business views. A business view, or a view, is a tailored presentation of Oracle Beehive data.

Database administrators can query a view to obtain statistical reports of Oracle Beehive data. For instance, you can query a view to obtain reports about workspace count, quota usage, quota limits, and so on.

---

**Note:** Business views are different from data provided by Oracle Enterprise Manager Grid Control. A business view provides information about how users use resources rather than about system uptime or memory usage.

---

This section includes the following topics

- [Accessing a View](#)
- [Saving a View](#)

## Accessing a View

Database administrators access views by logging on to Oracle Database as the BEE\_CODE user. The BEE\_CODE user is an Oracle Beehive user who makes database

transactions. The BEE\_CODE user can then query the view using a SQL query tool such as SQL\*Plus.

For example, to query the view `bee_team_wspc_members_rv` to get a count of all distinct user-typed members for all team workspaces, perform the following steps:

1. Start SQL\*Plus at the command prompt:

```
prompt> sqlplus /nolog
```

2. Connect to Oracle Database as the BEE\_CODE user:

```
SQL> CONNECT BEE_CODE/PASSWORD
```

3. You can either query a view or use the DESCRIBE command to get a list of all the columns in a view:

```
DESCRIBE bee_team_wspc_members_rv
```

4. The output of the command is as follows:

Name	Null?	Type
-----	-----	-----
WORKSPACE_EID	NOT NULL	RAW(22)
WORKSPACE_TYPE		CHAR(4)
NUM_MEMBERS		NUMBER

### Performance Issues

A view requires no storage other than storage for the definition of the view. A view is not allocated any storage space because it does not actually contain any data. Instead, a view references a base table each time you query the view and displays the output.

So, administrators should take into consideration the fact that querying a view can be time-consuming and affect database performance while the view is being queried.

## Saving a View

By default, the output of a view appears at the command prompt. However, you can store this output to a file by using the SQL\*Plus command `SPOOL`.

---

**Note:** If you are not using SQL\*Plus but another SQL query tool, then you can use a command offered by the SQL query tool you are using and which is similar in functionality to `SPOOL`.

---

For example, to append the output to the existing file `NOTES`, use the following command:

```
SPOOL NOTES APPEND
```

**See Also:** *SQL\*Plus User's Guide and Reference* for more information about the `SPOOL` command

## Types of Views

This section describes the following kinds of views:

- **Regular Reporting Views:** The names of these views end with `_rv`. Results of regular views do not contain sensitive information.

- **Sensitive Reporting Views:** The names of these views end with `_s_rv`.

Most business views have both an `_rv` and an `_s_rv` version; the only difference is that entity names are included in the `_s_rv` version.

## Regular Reporting Views

You can query a regular view to obtain non-sensitive information about the data contained in Oracle Beehive. Results of regular views do not contain sensitive information such as entity names of objects like files and workspaces. As per the naming conventions, regular views end with `_rv`.

[Example 5-1](#) shows information in a regular view.

### Example 5-1 Information In a Regular View

Name	Null?	Type
ENTERPRISE_EID	NOT NULL	RAW(22)
VISIBILITY	NOT NULL	CHAR(1 CHAR)
NUM_WORKSPACES		NUMBER
QUOTA_USED		NUMBER
OVER_QUOTA		CHAR(1)
NUM_ORGANIZATIONS		NUMBER

## Sensitive Reporting Views

You can query a sensitive view to obtain sensitive information about the data contained in Oracle Beehive. Results of these views include entity names of objects such as files and workspaces.

For example, a sensitive view might contain information about the company budget and only a senior level executive of the finance division or the administrator can query it. As per the naming conventions, sensitive views end with an `_s_rv`.

[Example 5-2](#) shows the information in a sensitive view.

### Example 5-2 Information In a Sensitive View

Name	Null?	Type
ENTERPRISE_EID	NOT NULL	RAW(22)
ENTERPRISE_NAME		VARCHAR2(1000 CHAR)
VISIBILITY	NOT NULL	CHAR(1 CHAR)
NUM_WORKSPACES		NUMBER
QUOTA_USED		NUMBER
OVER_QUOTA		CHAR(1)
NUM_ORGANIZATIONS		NUMBER

---

**Note:** You can restrict access to sensitive views by creating a database user with restricted privileges. Such a user will have access only to regular views and not to sensitive views.

---

**See Also:** Chapter 11, "Administering User Privileges, Roles, and Profiles" in the *Oracle Database Security Guide*.

# Workspace Service Views

This section contains a comprehensive list of business views related to the Workspace Service. This section also includes sample queries that you can run for each view.

This section lists the following views:

- [bee\\_enterprises](#)
- [bee\\_enterprise\\_bonds](#)
- [bee\\_enterprise\\_users](#)
- [bee\\_organizations](#)
- [bee\\_team\\_wspc\\_members](#)
- [bee\\_wspc\\_member\\_roles](#)
- [bee\\_wspc\\_role\\_groups](#)
- [bee\\_wspc\\_role\\_users](#)
- [bee\\_wspc\\_trash\\_totals](#)
- [bee\\_user\\_markers](#)
- [bee\\_user\\_marker\\_totals](#)
- [bee\\_workspaces](#)
- [bee\\_workspace\\_documents](#)
- [bee\\_workspace\\_trash](#)

## bee\_enterprises

This view provides a workspace count, organization count, quota usage, quota state, and basic properties such as name, EID, and visibility for all enterprises.

### bee\_enterprises\_s\_rv

This sensitive view displays the following information:

Name	Null?	Type
-----		-----
ENTERPRISE_EID	NOT NULL	RAW(22)
ENTERPRISE_NAME		VARCHAR2(1000 CHAR)
VISIBILITY	NOT NULL	CHAR(1 CHAR)
NUM_WORKSPACES		NUMBER
QUOTA_USED		NUMBER
OVER_QUOTA		CHAR(1)
NUM_ORGANIZATIONS		NUMBER

### bee\_enterprises\_rv

This regular view displays the following information:

Name	Null?	Type
-----		-----
ENTERPRISE_EID	NOT NULL	RAW(22)
VISIBILITY	NOT NULL	CHAR(1 CHAR)
NUM_WORKSPACES		NUMBER
QUOTA_USED		NUMBER
OVER_QUOTA		CHAR(1)
NUM_ORGANIZATIONS		NUMBER



**Suggested Queries for This View**

1. Total space consumed by all enterprises:

```
SELECT SUM(quota_used) FROM bee_enterprises_rv;
```

2. Total space utilization by all enterprises which are not active/visible:

```
SELECT SUM(quota_used) AS total_size FROM bee_enterprises_rv
WHERE visibility != 'V';
```

3. Average quota usage (in bytes) per enterprise:

```
SELECT AVG(quota_used) FROM bee_enterprises_rv;
```

4. Space consumed by each enterprise:

```
SELECT enterprise_eid, quota_used, over_quota
FROM bee_enterprises_rv;
```

5. Number of enterprises over hard quota limits:

```
SELECT COUNT(*) FROM bee_enterprises_rv
WHERE over_quota = 'Y';
```

6. Number of organizations per enterprise:

```
SELECT enterprise_eid, num_organizations FROM bee_enterprises_rv;
```

7. Number of workspaces per enterprise:

```
SELECT enterprise_eid, num_workspaces FROM bee_enterprises_rv;
```

8. Average number of organizations per enterprise:

```
SELECT AVG(num_organizations) FROM bee_enterprises_rv;
```

9. Average number of workspaces per enterprise:

```
SELECT AVG(num_workspaces) FROM bee_enterprises_rv;
```

**bee\_enterprise\_bonds**

This view provides bond properties such as EID, name, type, root type, root EID, and modification time for each enterprise, allowing for aggregation queries on bonds per enterprise.

**bee\_enterprise\_bonds\_s\_rv**

This sensitive view displays the following information:

Name	Null?	Type
ENTERPRISE_EID	NOT NULL	RAW(22)
ENTERPRISE_NAME		VARCHAR2(1000 CHAR)
ENTERPRISE_VISIBILITY	NOT NULL	CHAR(1 CHAR)
BOND_EID	NOT NULL	RAW(22)
BOND_NAME	NOT NULL	VARCHAR2(1000 CHAR)
BOND_TYPE	NOT NULL	VARCHAR2(1 CHAR)
BOND_ROOT_EID	NOT NULL	RAW(22)
BOND_ROOT_TYPE	NOT NULL	VARCHAR2(4 CHAR)
BOND_MODIFIED_ON	NOT NULL	TIMESTAMP(6)
BOND_DELETED	NOT NULL	CHAR(1 CHAR)

**bee\_enterprise\_bonds\_rv**

This regular view displays the following information:

Name	Null?	Type
ENTERPRISE_EID	NOT NULL	RAW(22)
ENTERPRISE_VISIBILITY	NOT NULL	CHAR(1 CHAR)
BOND_EID	NOT NULL	RAW(22)
BOND_TYPE	NOT NULL	VARCHAR2(1 CHAR)
BOND_ROOT_EID	NOT NULL	RAW(22)
BOND_ROOT_TYPE	NOT NULL	VARCHAR2(4 CHAR)
BOND_MODIFIED_ON	NOT NULL	TIMESTAMP(6)
BOND_DELETED	NOT NULL	CHAR(1 CHAR)

**Suggested Queries for This View**

1. Total number of bonds in each enterprise:

```
SELECT enterprise_eid,
       enterprise_visibility,
       COUNT(bond_eid) as num_bonds
FROM   bee_enterprise_bonds_rv
GROUP BY enterprise_eid, enterprise_visibility
ORDER BY num_bonds DESC;
```

2. Number of bonds of type *related materials* in each enterprise. Use F for *follow up* and D for *discuss this* bonds:

```
SELECT enterprise_eid,
       enterprise_visibility,
       COUNT(bond_eid) as num_related_bonds
FROM   bee_enterprise_bonds_rv
WHERE  bond_type = 'R'
GROUP BY enterprise_eid, enterprise_visibility
ORDER BY num_related_bonds DESC;
```

3. Number of bonds with root of type document in each enterprise:

```
SELECT enterprise_eid,
       enterprise_visibility,
       COUNT(bond_eid) as num_docroot_bonds
FROM   bee_enterprise_bonds_rv
WHERE  bond_root_type = 'adoc'
GROUP BY enterprise_eid, enterprise_visibility
ORDER BY num_docroot_bonds DESC;
```

4. Average number of bonds per enterprise:

```
SELECT AVG(num_bonds) as avg_num_bonds
FROM   (SELECT COUNT(bond_eid) as num_bonds
        FROM   bee_enterprise_bonds_rv
        GROUP BY enterprise_eid);
```

5. Number of bonds of each type in the system:

```
SELECT DECODE(bond_type, 'R', 'RELATED_MATERIALS',
              'F', 'FOLLOW_UP',
              'D', 'DISCUSS_THIS') as bond_type,
       COUNT(bond_eid) as num_bonds
```

```
FROM   bee_enterprise_bonds_rv
GROUP BY bond_type
ORDER BY bond_type;
```

**6. Number of bonds with root of each entity type in the system:**

```
SELECT bond_root_type as bond_root_type,
       COUNT(bond_eid) as num_bonds
FROM   bee_enterprise_bonds_rv
GROUP BY bond_root_type
ORDER BY bond_root_type;
```

## bee\_enterprise\_users

This view provides a user count for all enterprises.

### bee\_enterprise\_users\_s\_rv

This sensitive view displays the following information:

Name	Null?	Type
-----	-----	-----
ENTERPRISE_EID	NOT NULL	RAW(22)
ENTERPRISE_NAME		VARCHAR2(1000 CHAR)
VISIBILITY	NOT NULL	CHAR(1 CHAR)
NUM_USERS		NUMBER

### bee\_enterprise\_users\_rv

This regular view displays the following information:

Name	Null?	Type
-----	-----	-----
ENTERPRISE_EID	NOT NULL	RAW(22)
VISIBILITY	NOT NULL	CHAR(1 CHAR)
NUM_USERS		NUMBER

## Suggested Queries for This View

**1. Average number of users per enterprise:**

```
SELECT AVG(num_users) FROM bee_enterprise_users_rv;
```

**2. Ten most populous enterprises and their user counts:**

```
SELECT * FROM
  (SELECT enterprise_eid, visibility, num_users
   FROM   bee_enterprise_users_rv
   ORDER BY num_users DESC)
WHERE  ROWNUM <= 10;
```

## bee\_organizations

This view provides a total workspace count, child workspace count, quota usage, flags for "top\_level" and "has\_child\_orgs", and basic properties such as name, EID, enterprise name, and enterprise EID for all organizations.

**bee\_organizations\_s\_rv**

This sensitive view displays the following information:

Name	Null?	Type
ORGANIZATION_EID	NOT NULL	RAW(22)
ORGANIZATION_NAME	NOT NULL	VARCHAR2(1000 CHAR)
VISIBILITY	NOT NULL	CHAR(1 CHAR)
ENTERPRISE_EID	NOT NULL	RAW(22)
ENTERPRISE_NAME		VARCHAR2(1000 CHAR)
TOTAL_WORKSPACES		NUMBER
CHILD_WORKSPACES		NUMBER
QUOTA_USED		NUMBER
IS_TOPLEVEL_ORG		CHAR(1)
HAS_CHILD_ORGS		CHAR(1)
OVER_QUOTA		CHAR(1)

**bee\_organizations\_rv**

This regular view displays the following information:

Name	Null?	Type
ORGANIZATION_EID	NOT NULL	RAW(22)
VISIBILITY	NOT NULL	CHAR(1 CHAR)
ENTERPRISE_EID	NOT NULL	RAW(22)
TOTAL_WORKSPACES		NUMBER
CHILD_WORKSPACES		NUMBER
QUOTA_USED		NUMBER
IS_TOPLEVEL_ORG		CHAR(1)
HAS_CHILD_ORGS		CHAR(1)
OVER_QUOTA		CHAR(1)

**Suggested Queries for This View**

1. Total space consumed by all organizations:

```
SELECT SUM(quota_used) FROM bee_organizations_rv;
```

2. Average quota usage (in bytes) per organization:

```
SELECT AVG(quota_used) FROM bee_organizations_rv;
```

3. Average quota usage (in bytes) per *leaf* organization::

```
SELECT AVG(quota_used) FROM bee_organizations_rv
WHERE has_child_orgs = 'N';
```

4. Average quota usage (in bytes) per *root* organization:

```
SELECT AVG(quota_used) FROM bee_organizations_rv
WHERE is_toplevel_org = 'Y';
```

5. Number of organizations over hard quota limits:

```
SELECT COUNT(*) FROM bee_organizations_rv
WHERE over_quota = 'Y';
```

**6. Total number of workspaces in each organization:**

```
SELECT organization_eid, total_workspaces FROM bee_organizations_rv;
```

**7. Number of direct child workspaces in each organization:**

```
SELECT organization_eid, child_workspaces FROM bee_organizations_rv;
```

**8. Number of workspaces in each *root* organization:**

```
SELECT organization_eid, total_workspaces FROM bee_organizations_rv
WHERE is_toplevel_org = 'Y';
```

**9. Average number of workspaces per organization:**

```
SELECT AVG(total_workspaces) FROM bee_organizations_rv;
```

**10. Average number of direct child workspaces per organization:**

```
SELECT AVG(child_workspaces) FROM bee_organizations_rv;
```

**11. Average number of workspaces per "leaf" organization:**

```
SELECT AVG(total_workspaces) FROM bee_organizations_rv
WHERE has_child_orgs = 'N';
```

**12. Distribution percentile for number of child workspaces in each organization.  
Percentile of all organizations with fewer or equal child workspaces:**

```
SELECT organization_eid, child_workspaces,
       (CUME_DIST() OVER (ORDER BY child_workspaces NULLS FIRST)) as percentile
FROM   bee_organizations_rv;
```

**13. Number of direct child workspaces for median, or other percentile organization(s):**

```
SELECT organization_eid, child_workspaces, 0.50 as percentile
FROM   bee_organizations_rv
WHERE  child_workspaces = (SELECT (PERCENTILE_DISC(0.50) WITHIN GROUP
                                (ORDER BY child_workspaces ASC NULLS FIRST)) FROM bee_organizations_rv);
```

**bee\_team\_wspc\_members**

This view provides a count of all distinct user-typed members for all team workspaces, traversing the group hierarchy as needed.

**bee\_team\_wspc\_members\_s\_rv**

This sensitive view displays the following information:

Name	Null?	Type
-----	-----	-----
WORKSPACE_EID	NOT NULL	RAW(22)
WORKSPACE_NAME	NOT NULL	VARCHAR2(1000 CHAR)
WORKSPACE_TYPE		CHAR(4)
NUM_MEMBERS		NUMBER

**bee\_team\_wspc\_members\_rv**

This regular view displays the following information:

Name	Null?	Type
------	-------	------

Name	Null?	Type
WORKSPACE_EID	NOT NULL	RAW(22)
WORKSPACE_TYPE		CHAR(4)
NUM_MEMBERS		NUMBER

### Suggested Queries for This View

1. Average number of user members per team workspace:

```
SELECT AVG(num_members) FROM bee_team_wspc_members_rv;
```

2. Ten most populous team workspaces and their membership counts:

```
SELECT * FROM
  (SELECT workspace_eid, workspace_type, num_members
   FROM bee_team_wspc_members_rv
   ORDER BY num_members DESC)
WHERE ROWNUM <= 10;
```

3. Distribution percentile of number of members in each workspace. Percentile of all workspaces with fewer or equal members:

```
SELECT workspace_eid, num_members,
       (CUME_DIST() OVER (ORDER BY num_members NULLS FIRST)) as percentile
FROM bee_team_wspc_members_rv;
```

4. Number of members for median or other percentile workspaces:

```
SELECT workspace_eid, num_members, 0.50 as percentile
FROM bee_team_wspc_members_rv
WHERE num_members = (SELECT (PERCENTILE_DISC(0.50) WITHIN GROUP
                          (ORDER BY num_members ASC NULLS FIRST)) FROM bee_team_wspc_members_rv);
```

## bee\_user\_markers

This view lists all markers, marker names, marker types, and marker classifications for markers owned by each user in the system. It allows for queries of marker counts/types by user and enterprise.

Oracle Beehive clients, such as Oracle Beehive Zimbra, represent markers as tags. For example, in Oracle Beehive Zimbra, you may tag your tasks, mail, contacts, folders, and other entities. You may use predefined tags that come with Oracle Beehive Zimbra or create your own. This enables you to group and organize your entities in categories you have defined yourself.

### bee\_user\_markers\_s\_rv

This sensitive view displays the following information:

Name	Null?	Type
USER_EID	NOT NULL	RAW(22)
USER_NAME		VARCHAR2(1000 CHAR)
USER_CREATED_ON		TIMESTAMP(6)
USER_MODIFIED_ON		TIMESTAMP(6)
USER_VISIBILITY		VARCHAR2(1)
ENTERPRISE_EID	NOT NULL	RAW(22)
ENTERPRISE_NAME		VARCHAR2(1000 CHAR)
MARKER_EID		RAW(22)
MARKER_NAME		VARCHAR2(1000 CHAR)

MARKER_TYPE	VARCHAR2 (4 CHAR)
MARKER_CLASSIFICATION	VARCHAR2 (1 CHAR)
MARKER_VISIBILITY	CHAR (1 CHAR)

**bee\_user\_markers\_rv**

This regular view displays the following information:

Name	Null?	Type
USER_EID	NOT NULL	RAW (22)
USER_CREATED_ON		TIMESTAMP (6)
USER_MODIFIED_ON		TIMESTAMP (6)
USER_VISIBILITY		VARCHAR2 (1)
ENTERPRISE_EID	NOT NULL	RAW (22)
MARKER_EID		RAW (22)
MARKER_TYPE		VARCHAR2 (4 CHAR)
MARKER_CLASSIFICATION		VARCHAR2 (1 CHAR)
MARKER_VISIBILITY		CHAR (1 CHAR)

**Suggested Queries for This View**

1. Number of markers of type label defined/owned by each user:

```
SELECT user_eid, enterprise_eid,
       COUNT(marker_eid) as num_labels
FROM   bee_user_markers_rv
WHERE  marker_type = 'labl'
GROUP BY user_eid, enterprise_eid
ORDER BY num_labels DESC;
```

2. Number of markers of type category defined/owned by each user:

```
SELECT user_eid, enterprise_eid,
       COUNT(marker_eid) as num_categories
FROM   bee_user_markers_rv
WHERE  marker_type = 'catg'
GROUP BY user_eid, enterprise_eid
ORDER BY num_categories DESC;
```

**bee\_user\_marker\_totals**

This view provides a total marker count per user.

**bee\_user\_marker\_totals\_s\_rv**

This sensitive view displays the following information:

Name	Null?	Type
USER_EID	NOT NULL	RAW (22)
USER_NAME		VARCHAR2 (1000 CHAR)
ENTERPRISE_EID	NOT NULL	RAW (22)
ENTERPRISE_NAME		VARCHAR2 (1000 CHAR)
NUM_MARKERS		NUMBER

**bee\_user\_marker\_totals\_rv**

This regular view displays the following information:

Name	Null?	Type
USER_EID	NOT NULL	RAW(22)
ENTERPRISE_EID	NOT NULL	RAW(22)
NUM_MARKERS		NUMBER

### Suggested Queries for This View

1. Total number of markers defined/owned by each user:

```
SELECT * FROM bee_user_marker_totals_rv
ORDER BY num_markers DESC;
```

2. Average number of markers defined/owned by each user:

```
SELECT AVG(num_markers) FROM bee_user_marker_totals_rv;
```

3. Distribution percentile for number of markers for each user. Percent of all users with fewer or equal markers:

```
SELECT user_eid, num_markers,
       (CUME_DIST() OVER (ORDER BY num_markers NULLS FIRST)) as percentile
FROM   bee_user_marker_totals_rv
ORDER BY num_markers DESC;
```

4. Number of markers for median or other percentile users:

```
SELECT user_eid, num_markers, 0.50 as percentile
FROM   bee_user_marker_totals_rv
WHERE  num_markers =
       (SELECT (PERCENTILE_DISC(0.50) WITHIN GROUP
              (ORDER BY num_markers ASC NULLS FIRST))
        FROM bee_user_marker_totals_rv);
```

## bee\_workspaces

This view provides information such as quota usage, quota limits, quota state, and basic properties such as workspace type, enterprise name, parent EID, and last modified time for all workspaces in the system.

### bee\_workspaces\_s\_rv

This sensitive view displays the following information:

Name	Null?	Type
WORKSPACE_EID	NOT NULL	RAW(22)
WORKSPACE_NAME	NOT NULL	VARCHAR2(1000 CHAR)
PARENT_EID	NOT NULL	RAW(22)
ENTERPRISE_EID	NOT NULL	RAW(22)
ENTERPRISE_NAME		VARCHAR2(1000 CHAR)
WORKSPACE_TYPE	NOT NULL	VARCHAR2(4 CHAR)
MODIFIED_ON	NOT NULL	TIMESTAMP(6)
VISIBILITY	NOT NULL	CHAR(1 CHAR)
QUOTA_USED		NUMBER(38)
QUOTA_STATE		VARCHAR2(1 CHAR)
SOFT_QUOTA		NUMBER(38)
HARD_QUOTA		NUMBER(38)
PCT_SOFT_USAGE		NUMBER
PCT_HARD_USAGE		NUMBER



**bee\_workspaces\_rv**

This regular view displays the following information:

Name	Null?	Type
WORKSPACE_EID	NOT NULL	RAW(22)
PARENT_EID	NOT NULL	RAW(22)
ENTERPRISE_EID	NOT NULL	RAW(22)
WORKSPACE_TYPE	NOT NULL	VARCHAR2(4 CHAR)
MODIFIED_ON	NOT NULL	TIMESTAMP(6)
VISIBILITY	NOT NULL	CHAR(1 CHAR)
QUOTA_USED		NUMBER(38)
QUOTA_STATE		VARCHAR2(1 CHAR)
SOFT_QUOTA		NUMBER(38)
HARD_QUOTA		NUMBER(38)
PCT_SOFT_USAGE		NUMBER
PCT_HARD_USAGE		NUMBER

**Suggested Queries for This View****1. Total number of workspaces:**

```
SELECT COUNT(*) AS total_workspaces FROM bee_workspaces_rv;
```

**2. Total number of team workspaces:**

```
SELECT COUNT(*) AS team_workspaces FROM bee_workspaces_rv
WHERE workspace_type = 'wstm';
```

**3. Total number of personal workspaces:**

```
SELECT COUNT(*) AS personal_workspaces FROM bee_workspaces_rv
WHERE workspace_type = 'wspr';
```

**4. Number of workspaces exceeding soft quota limits:**

```
SELECT COUNT(*) AS workspaces_over_soft_quota FROM bee_workspaces_rv
WHERE quota_state = 'S';
```

**5. Number of team workspaces exceeding soft quota limits:**

```
SELECT COUNT(*) AS team_wspcs_over_soft_quota FROM bee_workspaces_rv
WHERE quota_state = 'S'
AND workspace_type = 'wstm';
```

**6. Number of personal workspaces exceeding soft quota limits:**

```
SELECT COUNT(*) AS pers_wspcs_over_soft_quota FROM bee_workspaces_rv
WHERE quota_state = 'S'
AND workspace_type = 'wspr';
```

**7. Number of workspaces exceeding hard quota limits:**

```
SELECT COUNT(*) AS workspaces_over_hard_quota FROM bee_workspaces_rv
WHERE quota_state = 'H';
```

**8. Number of team workspaces exceeding hard quota limits:**

```
SELECT COUNT(*) AS team_wspcs_over_hard_quota FROM bee_workspaces_rv
WHERE quota_state = 'H'
AND workspace_type = 'wstm';
```

**9. Number of personal workspaces exceeding hard quota limits:**

```
SELECT COUNT(*) AS pers_wspscs_over_hard_quota FROM bee_workspaces_rv
WHERE  quota_state = 'H'
AND    workspace_type = 'wspr';
```

**10. Total space utilization by all workspaces:**

```
SELECT SUM(quota_used) AS total_size FROM bee_workspaces_rv;
```

**11. Total space utilization by all workspaces which are inactive/visible:**

```
SELECT SUM(quota_used) AS total_size FROM bee_workspaces_rv
WHERE  visibility != 'V';
```

**12. Percentile (fractional) of soft quota used by all workspaces which have explicit soft quota limits:**

```
SELECT (SUM(quota_used) / SUM(soft_quota)) AS pct_soft_usage
FROM   bee_workspaces_rv
WHERE  soft_quota >= 0;
```

**13. Percentile (fractional) of hard quota used by all workspaces which have explicit hard quota limits:**

```
SELECT (SUM(quota_used) / SUM(hard_quota)) AS pct_hard_usage
FROM   bee_workspaces_rv
WHERE  hard_quota >= 0;
```

**14. Ten largest workspaces (by data size):**

```
SELECT * FROM
  (SELECT workspace_eid, workspace_type, data_size
   FROM bee_workspaces_rv
   ORDER BY data_size DESC)
WHERE  ROWNUM <= 10;
```

**15. Ten most recently modified workspaces:**

```
SELECT * FROM
  (SELECT workspace_eid, workspace_type, modified_on
   FROM bee_workspaces_rv
   ORDER BY modified_on DESC)
WHERE  ROWNUM <= 10;
```

**16. Quota usage distribution percentile for all workspaces. Percentile of all workspaces with lower or equal quota usage for each workspace:**

```
SELECT workspace_eid, quota_used,
       (CUME_DIST() OVER (ORDER BY quota_used NULLS FIRST)) as percentile
FROM   bee_workspaces_rv;
```

**17. Quota usage for median (or other percentile) workspace(s):**

```
SELECT workspace_eid, quota_used, 0.50 as percentile
FROM   bee_workspaces_rv w
WHERE  quota_used = (SELECT (PERCENTILE_DISC(0.50) WITHIN GROUP
                          (ORDER BY quota_used ASC NULLS FIRST)) FROM bee_workspaces_rv);
```

## bee\_workspace\_documents

This view provides document properties such as creation time, modification time, size, document type, media (MIME) type, and visibility for all documents mapped to their

containing workspaces. It allows for queries of document sizes/counts by workspace, document type, date ranges, and so on.

This view includes all documents contained in heterogeneous folders but not documents in trash or archive folders. Also, this view includes un-versioned or family document entities while filtering out version entities as redundant.

### **bee\_workspace\_documents\_s\_rv**

This sensitive view displays the following information:

Name	Null?	Type
-----	-----	-----
WORKSPACE_EID	NOT NULL	RAW(22)
WORKSPACE_NAME	NOT NULL	VARCHAR2(1000 CHAR)
WORKSPACE_TYPE	NOT NULL	VARCHAR2(4 CHAR)
WORKSPACE_MODIFIED_ON	NOT NULL	TIMESTAMP(6)
WORKSPACE_VISIBILITY	NOT NULL	CHAR(1 CHAR)
DOCUMENT_EID	NOT NULL	RAW(22)
DOCUMENT_NAME		VARCHAR2(1000 CHAR)
DOCUMENT_CREATED_ON	NOT NULL	TIMESTAMP(6)
DOCUMENT_MODIFIED_ON	NOT NULL	TIMESTAMP(6)
DOCUMENT_VISIBILITY	NOT NULL	CHAR(1 CHAR)
DOCUMENT_DATA_SIZE	NOT NULL	NUMBER(38)
DOCUMENT_IS_HIDDEN	NOT NULL	CHAR(1 CHAR)
DOCUMENT_MEDIA_TYPE		VARCHAR2(4000 CHAR)
DOCUMENT_DATA_TYPE		VARCHAR2(8)

### **bee\_workspace\_documents\_rv**

This regular view displays the following information:

Name	Null?	Type
-----	-----	-----
WORKSPACE_EID	NOT NULL	RAW(22)
WORKSPACE_TYPE	NOT NULL	VARCHAR2(4 CHAR)
WORKSPACE_MODIFIED_ON	NOT NULL	TIMESTAMP(6)
WORKSPACE_VISIBILITY	NOT NULL	CHAR(1 CHAR)
DOCUMENT_EID	NOT NULL	RAW(22)
DOCUMENT_CREATED_ON	NOT NULL	TIMESTAMP(6)
DOCUMENT_MODIFIED_ON	NOT NULL	TIMESTAMP(6)
DOCUMENT_VISIBILITY	NOT NULL	CHAR(1 CHAR)
DOCUMENT_DATA_SIZE	NOT NULL	NUMBER(38)
DOCUMENT_IS_HIDDEN	NOT NULL	CHAR(1 CHAR)
DOCUMENT_MEDIA_TYPE		VARCHAR2(4000 CHAR)
DOCUMENT_DATA_TYPE		VARCHAR2(8)

### **Suggested Queries for This View**

1. Total number of documents for each workspace:

```
SELECT workspace_eid, workspace_type,
       COUNT(document_eid) AS num_documents
FROM   bee_workspace_documents_rv
GROUP BY workspace_eid, workspace_type
ORDER BY num_documents DESC;
```

2. Total number of BLOB-stored documents for each workspace:

```
SELECT workspace_eid, workspace_type,
       COUNT(document_eid) AS num_blob_documents
```

```
FROM bee_workspace_documents_rv
WHERE document_data_type = 'BLOB'
GROUP BY workspace_eid, workspace_type
ORDER BY num_blob_documents DESC;
```

**3. Total number of XDB/BLOB-stored documents for each workspace:**

```
SELECT workspace_eid, workspace_type,
       COUNT(document_eid) AS num_xdbblob_documents
FROM bee_workspace_documents_rv
WHERE document_data_type = 'XDB_BLOB'
GROUP BY workspace_eid, workspace_type
ORDER BY num_xdbblob_documents DESC;
```

**4. Total number of BFILE documents for each workspace:**

```
SELECT workspace_eid, workspace_type,
       COUNT(document_eid) AS num_bfile_documents
FROM bee_workspace_documents_rv
WHERE document_data_type = 'BFILE'
GROUP BY workspace_eid, workspace_type
ORDER BY num_bfile_documents DESC;
```

**5. Total number of documents with media type "text" for each workspace:**

```
SELECT workspace_eid, workspace_type,
       COUNT(document_eid) AS num_text_documents
FROM bee_workspace_documents_rv
WHERE document_media_type like 'text%'
GROUP BY workspace_eid, workspace_type
ORDER BY num_text_documents DESC;
```

**6. Total number of documents for all team workspaces and all personal workspaces:**

```
SELECT DECODE(workspace_type, 'wstm', 'TEAM', 'wspr', 'PERSONAL')
       as workspace_type,
       COUNT(document_eid) AS num_documents
FROM bee_workspace_documents_rv
GROUP BY workspace_type
ORDER BY num_documents DESC;
```

**7. Total space consumed by documents for each workspace:**

```
SELECT workspace_eid, workspace_type,
       SUM(document_data_size) AS total_doc_size
FROM bee_workspace_documents_rv
GROUP BY workspace_eid, workspace_type
ORDER BY total_doc_size DESC;
```

**8. Average document size for each workspace:**

```
SELECT workspace_eid, workspace_type,
       AVG(document_data_size) AS avg_doc_size
FROM bee_workspace_documents_rv
GROUP BY workspace_eid, workspace_type
ORDER BY avg_doc_size DESC;
```

**9. Total size of documents for all team workspaces and all personal workspaces:**

```
SELECT DECODE(workspace_type, 'wstm', 'TEAM', 'wspr', 'PERSONAL')
       as workspace_type,
       SUM(document_data_size) AS total_doc_size
FROM bee_workspace_documents_rv
```

```
GROUP BY workspace_type
ORDER BY total_doc_size DESC;
```

#### 10. Ten workspaces with the most recently modified documents:

```
SELECT * FROM
  (SELECT * FROM
    (SELECT workspace_eid, workspace_type,
      MAX(document_modified_on) as latest_doc_modified_on
    FROM bee_workspace_documents_rv
    GROUP BY workspace_eid, workspace_type)
  ORDER BY latest_doc_modified_on DESC)
WHERE ROWNUM <= 10
```

### bee\_workspace\_trash

This view provides trash item properties such as item EID, item name, original entity EID, original name, original parent type, deleted by EID, deleted by type, deleted on, and data size for every item in a workspace trash folder. This view can be used to determine current trash usage statistics across workspaces and for specific workspaces.

#### bee\_workspace\_trash\_s\_rv

This sensitive view displays the following information:

Name	Null?	Type
WORKSPACE_EID	NOT NULL	RAW(22)
WORKSPACE_NAME	NOT NULL	VARCHAR2(1000 CHAR)
WORKSPACE_TYPE	NOT NULL	VARCHAR2(4 CHAR)
WORKSPACE_VISIBILITY	NOT NULL	CHAR(1 CHAR)
ITEM_EID		RAW(22)
ITEM_NAME		VARCHAR2(1000 CHAR)
ITEM_ENTITY_EID		RAW(22)
ITEM_ENTITY_TYPE		VARCHAR2(4 CHAR)
ITEM_ORIGINAL_NAME		VARCHAR2(1000 CHAR)
ITEM_ORIGINAL_PARENT_EID		RAW(22)
ITEM_ORIGINAL_PARENT_TYPE		VARCHAR2(4 CHAR)
DELETED_BY_EID		RAW(22)
DELETED_BY_TYPE		VARCHAR2(4 CHAR)
DELETED_ON		TIMESTAMP(6)
DATA_SIZE		NUMBER(38)

#### bee\_workspace\_trash\_rv

This regular view displays the following information:

Name	Null?	Type
WORKSPACE_EID	NOT NULL	RAW(22)
WORKSPACE_TYPE	NOT NULL	VARCHAR2(4 CHAR)
WORKSPACE_VISIBILITY	NOT NULL	CHAR(1 CHAR)
ITEM_EID		RAW(22)
ITEM_ENTITY_EID		RAW(22)
ITEM_ENTITY_TYPE		VARCHAR2(4 CHAR)
ITEM_ORIGINAL_PARENT_EID		RAW(22)
ITEM_ORIGINAL_PARENT_TYPE		VARCHAR2(4 CHAR)
DELETED_BY_EID		RAW(22)
DELETED_BY_TYPE		VARCHAR2(4 CHAR)

DELETED_ON	TIMESTAMP(6)
DATA_SIZE	NUMBER(38)

Suggested Queries for This View

1. Total number of trash items in each workspace:

```
SELECT workspace_eid, workspace_type,
       COUNT(item_eid) as num_trash_items
FROM   bee_workspace_trash_rv
GROUP BY workspace_eid, workspace_type
ORDER BY num_trash_items DESC;
```

2. Total number of trash items of type document in each workspace:

```
SELECT workspace_eid, workspace_type,
       COUNT(item_eid) as num_trash_items
FROM   bee_workspace_trash_rv
WHERE  item_entity_type = 'adoc'
GROUP BY workspace_eid, workspace_type
ORDER BY num_trash_items DESC;
```

3. All trash items larger than 10MB in size:

```
SELECT workspace_eid, workspace_type,
       item_eid, item_entity_eid, item_entity_type,
       data_size
FROM   bee_workspace_trash_rv
WHERE  data_size > 10485760
ORDER BY data_size DESC;
```

4. All trash items deleted before January 25, 2007:

```
SELECT workspace_eid, workspace_type,
       item_eid, item_entity_eid, item_entity_type,
       deleted_by_eid, deleted_on, data_size
FROM   bee_workspace_trash_rv
WHERE  deleted_on < TO_TIMESTAMP('25-Jan-07 12:00:00.000000', 'DD-Mon-RR
HH24:MI:SS.FF')
ORDER BY deleted_on;
```

bee\_wspc\_member\_roles

This view lists every direct accessor and accessor type for each (workspace, role name) pair. It does not traverse the group hierarchy.

bee\_wspc\_member\_roles\_s\_rv

This sensitive view displays the following information:

Name	Null?	Type
-----	-----	-----
WORKSPACE_EID	NOT NULL	RAW(22)
WORKSPACE_NAME	NOT NULL	VARCHAR2(1000 CHAR)
WORKSPACE_TYPE	NOT NULL	VARCHAR2(4 CHAR)
ROLE_NAME	NOT NULL	VARCHAR2(1000 CHAR)
ACCESSOR_EID	NOT NULL	RAW(22)
ACCESSOR_TYPE	NOT NULL	VARCHAR2(4)

**bee\_wspc\_member\_roles\_rv**

This regular view displays the following information:

Name	Null?	Type
WORKSPACE_EID	NOT NULL	RAW(22)
WORKSPACE_TYPE	NOT NULL	VARCHAR2(4 CHAR)
ROLE_NAME	NOT NULL	VARCHAR2(1000 CHAR)
ACCESSOR_EID	NOT NULL	RAW(22)
ACCESSOR_TYPE	NOT NULL	VARCHAR2(4)

**Suggested Queries for This View**

1. Number of direct accessors of type user for each (workspace, role) pair:

```
SELECT workspace_eid, role_name,
       COUNT(accessor_eid) as num_user_accessors
FROM   bee_wspc_member_roles_rv
WHERE  accessor_type = 'user'
GROUP BY workspace_eid, role_name
ORDER BY workspace_eid, role_name;
```

2. Number of direct accessors of type group for each (workspace, role) pair:

```
SELECT workspace_eid, role_name,
       COUNT(accessor_eid) as num_group_accessors
FROM   bee_wspc_member_roles_rv
WHERE  accessor_type = 'grup'
GROUP BY workspace_eid, role_name
ORDER BY workspace_eid, role_name;
```

**bee\_wspc\_role\_groups**

This view lists the complete distinct user count for each group accessor for each (workspace, role name) pair. It traverses the group hierarchy to calculate user totals.

**bee\_wspc\_role\_groups\_s\_rv**

This sensitive view displays the following information:

Name	Null?	Type
WORKSPACE_EID	NOT NULL	RAW(22)
WORKSPACE_NAME	NOT NULL	VARCHAR2(1000 CHAR)
ROLE_NAME	NOT NULL	VARCHAR2(1000 CHAR)
ACCESSOR_EID	NOT NULL	RAW(22)
NUM_GROUP_USERS		NUMBER

**bee\_wspc\_role\_groups\_rv**

This regular view displays the following information:

Name	Null?	Type
WORKSPACE_EID	NOT NULL	RAW(22)
ROLE_NAME	NOT NULL	VARCHAR2(1000 CHAR)
ACCESSOR_EID	NOT NULL	RAW(22)
NUM_GROUP_USERS		NUMBER

**Suggested Queries for This View**

1. Number of users for each accessor of type group for each (workspace, role) pair:

```
SELECT * FROM bee_wspc_role_groups_rv
ORDER BY workspace_eid, role_name, accessor_eid;
```

2. Total number of group users for each (workspace, role) pair:

```
SELECT workspace_eid, role_name,
       SUM(num_group_users) as num_group_users
FROM   bee_wspc_role_groups_rv
GROUP BY workspace_eid, role_name
ORDER BY workspace_eid, role_name;
```

**bee\_wspc\_role\_users**

This view provides a total user count for each (workspace, role name) pair, including all users assigned both directly and indirectly by groups. Note that the group hierarchy is traversed to compute the user totals.

**bee\_wspc\_role\_users\_s\_rv**

This sensitive view displays the following information:

Name	Null?	Type
-----	-----	-----
WORKSPACE_EID	NOT NULL	RAW(22)
WORKSPACE_NAME	NOT NULL	VARCHAR2(1000 CHAR)
WORKSPACE_TYPE	NOT NULL	VARCHAR2(4 CHAR)
ROLE_NAME	NOT NULL	VARCHAR2(1000 CHAR)
TOTAL_ROLE_USERS		NUMBER

**bee\_wspc\_role\_users\_rv**

This regular view displays the following information:

Name	Null?	Type
-----	-----	-----
WORKSPACE_EID	NOT NULL	RAW(22)
WORKSPACE_TYPE	NOT NULL	VARCHAR2(4 CHAR)
ROLE_NAME	NOT NULL	VARCHAR2(1000 CHAR)
TOTAL_ROLE_USERS		NUMBER

**Suggested Queries for This View**

1. Total number of users (both direct and from groups) for each (workspace,role) pair:

```
SELECT * from bee_wspc_role_users_rv
ORDER BY workspace_eid, role_name;
```

2. Ten workspaces with the most workspace-coordinator members:

```
SELECT * FROM
  (SELECT * from bee_wspc_role_users_rv
   WHERE role_name = 'workspace-coordinator'
   ORDER BY total_role_users DESC)
WHERE ROWNUM <= 10;
```



## bee\_wspc\_trash\_totals

This view totals up trash size for each workspace and can be used to determine trash size statistics across workspaces.

### bee\_wspc\_trash\_totals\_s\_rv

This sensitive view displays the following information:

Name	Null?	Type
WORKSPACE_EID	NOT NULL	RAW(22)
WORKSPACE_NAME	NOT NULL	VARCHAR2(1000 CHAR)
WORKSPACE_TYPE	NOT NULL	VARCHAR2(4 CHAR)
WORKSPACE_VISIBILITY	NOT NULL	CHAR(1 CHAR)
TOTAL_TRASH_SIZE		NUMBER

### bee\_wspc\_trash\_totals\_rv

This regular view displays the following information:

Name	Null?	Type
WORKSPACE_EID	NOT NULL	RAW(22)
WORKSPACE_TYPE	NOT NULL	VARCHAR2(4 CHAR)
WORKSPACE_VISIBILITY	NOT NULL	CHAR(1 CHAR)
TOTAL_TRASH_SIZE		NUMBER

## Suggested Queries for This View

1. Total space consumed by all trash items in each workspace:

```
SELECT workspace_eid, workspace_type, total_trash_size
FROM   bee_wspc_trash_totals_rv
ORDER BY total_trash_size DESC;
```

2. Ten team workspaces with most trash space consumption:

```
SELECT * FROM
  (SELECT workspace_eid, workspace_type, total_trash_size
   FROM   bee_wspc_trash_totals_rv
   WHERE  workspace_type = 'wstm'
   ORDER BY total_trash_size DESC)
WHERE ROWNUM <= 10;
```

3. Average space consumed by trash items per workspace:

```
SELECT AVG(total_trash_size) as average_trash_size
FROM   bee_wspc_trash_totals_rv;
```

4. Distribution percentile for total space consumed by trash items in each workspace. Percent of all workspaces with less than or equal trash space consumption:

```
SELECT workspace_eid, workspace_type, total_trash_size,
       (CUME_DIST() OVER (ORDER BY total_trash_size NULLS FIRST)) as percentile
FROM   bee_wspc_trash_totals_rv;
```

Total trash space consumption for median (or other percentile) workspace(s):

```
SELECT workspace_eid, workspace_type, total_trash_size, 0.50 as percentile
FROM   bee_wspc_trash_totals_rv
WHERE  total_trash_size = (SELECT (PERCENTILE_DISC(0.50) WITHIN GROUP
                                (ORDER BY total_trash_size ASC NULLS FIRST)) FROM bee_wspc_trash_
```

```
totals_rv);
```

## Device Management Service Views

This section describes the following views:

- [dm\\_applications](#)
- [dm\\_devices](#)
- [dm\\_device\\_apps](#)
- [dm\\_device\\_cmds](#)
- [dm\\_device\\_logs](#)
- [dm\\_dev\\_profs\\_to\\_dev\\_types](#)

### dm\_applications

This view provides information about client applications.

#### dm\_applications\_rv

Name	Null?	Type
-----	-----	-----
EID	NOT NULL	RAW(22)
SITE_ID	NOT NULL	NUMBER(5)
ENTERPRISE_ID	NOT NULL	NUMBER(5)
DESCRIPTION		VARCHAR2(4000)
NAME		VARCHAR2(1000 CHAR)
ISPLATFORM		CHAR(1 CHAR)
VENDOR		VARCHAR2(1000 CHAR)
OS		VARCHAR2(1000 CHAR)
PROCESSOR		VARCHAR2(1000 CHAR)
DEVICECLASS		VARCHAR2(1000 CHAR)

### dm\_devices

This view provides information about devices and their users.

#### dm\_devices\_rv

Name	Null?	Type
-----	-----	-----
EID	NOT NULL	RAW(22)
SITE_ID	NOT NULL	NUMBER(5)
ENTERPRISE_ID	NOT NULL	NUMBER(5)
OWNER_EID	NOT NULL	RAW(22)
OWNER_TYPE	NOT NULL	VARCHAR2(4)
OWNER_SITE_ID	NOT NULL	NUMBER(5)
CREATOR_EID	NOT NULL	RAW(22)
CREATOR_TYPE	NOT NULL	VARCHAR2(4)
CREATED_ON	NOT NULL	TIMESTAMP(0)
NAME		VARCHAR2(1000 CHAR)
DEVICE_ID	NOT NULL	VARCHAR2(1000 CHAR)
DEV_INF_DTD_VERSION		VARCHAR2(1000 CHAR)
SOFTWARE_VERSION		VARCHAR2(1000 CHAR)
OS		VARCHAR2(256 CHAR)
PROCESSOR		VARCHAR2(256 CHAR)
DEVICE_CLASS		VARCHAR2(256 CHAR)

MANUFACTURER	VARCHAR2 (256 CHAR)
MODEL	VARCHAR2 (256 CHAR)
DEVICE_INFO	SYS.XMLTYPE
STATUS	VARCHAR2 (20 CHAR)

## dm\_device\_apps

This view provides information about device applications.

### dm\_device\_apps\_rv

Name	Null?	Type
DEV_EID	NOT NULL	RAW (22)
DEV_SITE_ID	NOT NULL	NUMBER (5)
DEV_ENPR_ID	NOT NULL	NUMBER (5)
DEV_DEVICE_ID	NOT NULL	VARCHAR2 (1000 CHAR)
APP_EID	NOT NULL	RAW (22)
VER_VERSION		VARCHAR2 (1000 CHAR)
VER_VERSIONNUM	NOT NULL	NUMBER
PATCH_PATCHNUM	NOT NULL	NUMBER

## dm\_device\_cmds

This view provides information about device commands.

### dm\_device\_cmds\_rv

Name	Null?	Type
EID	NOT NULL	RAW (22)
SITE_ID	NOT NULL	NUMBER (5)
ENTERPRISE_ID	NOT NULL	NUMBER (5)
CREATED_ON	NOT NULL	TIMESTAMP (6)
EXECUTED_ON		TIMESTAMP (0)
EXECUTION_TIME		NUMBER (20)
ACTION	NOT NULL	VARCHAR2 (32 CHAR)
DEVICE_EID	NOT NULL	RAW (22)
DEVICE_SITE_ID	NOT NULL	NUMBER (5)
STATUS	NOT NULL	VARCHAR2 (32 CHAR)
STATUS_MSG		VARCHAR2 (1000 CHAR)

## dm\_device\_logs

This view provides information about device logs.

### dm\_device\_logs\_rv

Name	Null?	Type
EID	NOT NULL	RAW (22)
ENTERPRISE_ID	NOT NULL	NUMBER (5)
SITE_ID	NOT NULL	NUMBER (5)
DEVICE_ID	NOT NULL	RAW (22)
UPLOADED_ON		TIMESTAMP (0)
NAME	NOT NULL	VARCHAR2 (500 CHAR)
APP_NAME	NOT NULL	VARCHAR2 (1000 CHAR)
STREAMID		RAW (22)
LOG_SIZE		NUMBER
VISIBILITY_STATUS		CHAR (1)

## dm\_dev\_profs\_to\_dev\_types

This view provides information about device profiles.

### dm\_dev\_profs\_to\_dev\_types\_rv

Name	Null?	Type
-----		
DEV_TYPE_EID	NOT NULL	RAW(22)
DEV_TYPE_SITE_ID	NOT NULL	NUMBER(5)
DEV_TYPE_ENPR_ID	NOT NULL	NUMBER(5)
DEV_PROF_EID	NOT NULL	RAW(22)
DEV_PROF_SITE_ID	NOT NULL	NUMBER(5)
DEV_PROF_ENPR_ID	NOT NULL	NUMBER(5)
DEV_INF_DTD_VERSION		VARCHAR2(1000 CHAR)
DEV_NAME		VARCHAR2(1000 CHAR)
DEV_TYPE_SOFTWARE_VERSION		VARCHAR2(1000 CHAR)
DEV_TYPE_OS		VARCHAR2(256 CHAR)
DEV_TYPE_PROCESSOR		VARCHAR2(256 CHAR)
DEV_TYPE_DEVICE_CLASS		VARCHAR2(256 CHAR)
DEV_TYPE_MANUFACTURER		VARCHAR2(256 CHAR)
DEV_TYPE_MODEL		VARCHAR2(256 CHAR)
DEV_TYPE_STATUS		VARCHAR2(256 CHAR)
DEV_PROF_NAME		VARCHAR2(1000 CHAR)

## Collaboration Service Views

This section describes the following views related to e-mail messages:

- [es\\_message\\_headers](#)
- [es\\_message](#)

## es\_message\_headers

This view provides information about e-mail message headers.

### es\_message\_headers\_rv

Name	Null?	Type
-----		
MESSAGE_EID	NOT NULL	RAW(22)
FROM_STR		VARCHAR2(4000)

## es\_message

This view provides messages about e-mail messages.

### es\_message\_rv

Name	Null?	Type
-----		
OWNER_EID	NOT NULL	RAW(22)
FOLDER_EID	NOT NULL	RAW(22)
MESSAGE_EID	NOT NULL	RAW(22)
RECEIVED_DATE		DATE
MESSAGE_SIZE		NUMBER
SUBJECT	NOT NULL	VARCHAR2(1000 CHAR)

## Message Delivery Service Views

This section describes the view `mds_delivery_status`.

### `mds_delivery_status`

This view provides information about the delivery status of e-mail messages.

#### `mds_delivery_status_rv`

Name	Null?	Type
-----	-----	-----
EID	NOT NULL	RAW(22)
SITE_ID	NOT NULL	NUMBER(5)
ENTERPRISE_ID	NOT NULL	NUMBER(5)
OWNER_EID		RAW(22)
MSG_TYPE		VARCHAR2(4)
MSG_EID		RAW(22)
CHANNEL	NOT NULL	VARCHAR2(128 CHAR)
STATUS	NOT NULL	VARCHAR2(128 CHAR)

## Mobility Service Views

This section describes the following views:

- [oma\\_ds\\_data\\_operations](#)
- [oma\\_ds\\_data\\_stores](#)
- [oma\\_ds\\_sessions](#)
- [oma\\_ds\\_user\\_devices](#)
- [oma\\_syncml\\_logs](#)

### `oma_ds_data_operations`

This view provides information about the types of data that users may synchronize.

#### `oma_ds_data_operations_rv`

Name	Null?	Type
-----	-----	-----
STORE_ID	NOT NULL	RAW(22)
USER_ID		VARCHAR2(512)
DEVICE_ID	NOT NULL	VARCHAR2(512)
SERVER_STORE_URI		VARCHAR2(128)
TGUID		NUMBER
LUID		VARCHAR2(1024)
OPERATION_TYPE		VARCHAR2(1)
OPERATION_ORIGIN		VARCHAR2(1)
OPERATION_TIME		TIMESTAMP(6)

### `oma_ds_data_stores`

This view provides information about the data itself that users synchronize.

#### `oma_ds_data_stores_rv`

Name	Null?	Type
-----	-----	-----
STORE_ID	NOT NULL	RAW(22)

DEVICE_ID	NOT NULL	VARCHAR2 (512)
USER_ID		VARCHAR2 (512)
SERVER_STORE_URI		VARCHAR2 (128)
CLIENT_STORE_URI		VARCHAR2 (128)
SERVER_LAST_ANCHOR		VARCHAR2 (128)
CLIENT_LAST_ANCHOR		VARCHAR2 (128)
LAST_SYNC_TIME		TIMESTAMP (6)
MAN		VARCHAR2 (128)
MODEL		VARCHAR2 (128)

## oma\_ds\_sessions

This view provides information about when users synchronize their data.

### oma\_ds\_sessions\_rv

Name	Null?	Type
SESSION_ID	NOT NULL	NUMBER
CLIENT_SESSION_ID		VARCHAR2 (128)
DEVICE_ID		VARCHAR2 (512)
USER_ID		VARCHAR2 (512)
SESSION_STATUS		VARCHAR2 (30)
SESSION_START_TIME		TIMESTAMP (6)
SESSION_END_TIME		TIMESTAMP (6)
SESSION_LAST_ACCESS_TIME		TIMESTAMP (6)
SESSION_PROCESS_TIME		NUMBER
SESSION_AUTH_TYPE		VARCHAR2 (30)
SESSION_AUTHENTICATED		VARCHAR2 (1)
SYNC_PHASE		VARCHAR2 (512)
SYNC_TYPE		VARCHAR2 (512)
UPLOAD_BYTES		NUMBER
UPLOAD_ADD_COUNT		NUMBER
UPLOAD_DELETE_COUNT		NUMBER
UPLOAD_REPLACE_COUNT		NUMBER
DOWNLOAD_BYTES		NUMBER
DOWNLOAD_ADD_COUNT		NUMBER
DOWNLOAD_DELETE_COUNT		NUMBER
DOWNLOAD_REPLACE_COUNT		NUMBER
SESSION_MESSAGE		CLOB

## oma\_ds\_user\_devices

This view provides information about the devices users use to synchronize their data.

### oma\_ds\_user\_devices\_rv

Name	Null?	Type
DEVICE_ID	NOT NULL	VARCHAR2 (512)
USER_ID		VARCHAR2 (512)
SESSION_ID		NUMBER
MAN		VARCHAR2 (128)
MODEL		VARCHAR2 (128)
LAST_SYNC_TIME		TIMESTAMP (6)

## oma\_syncml\_logs

This view provides information about SyncML logs:

**oma\_syncml\_logs\_rv**

Name	Null?	Type
REQUEST_ID	NOT NULL	NUMBER
REQUEST_STATUS		VARCHAR2 (30)
DEVICE_ID		VARCHAR2 (512)
USER_ID		VARCHAR2 (512)
SESSION_ID		NUMBER
CLIENT_SESSION_ID		VARCHAR2 (128)
MESSAGE_ID		NUMBER (38)
REQUEST_TIME		TIMESTAMP (6)
REQUEST_META		VARCHAR2 (4000)
RESPONSE_TIME		TIMESTAMP (6)
RESPONSE_META		VARCHAR2 (4000)

## User Directory Service Views

This section describes the following views:

- [uds\\_address\\_books](#)
- [uds\\_contacts](#)
- [uds\\_external\\_persons](#)
- [uds\\_groups](#)
- [uds\\_group\\_contact\\_members](#)
- [uds\\_group\\_members](#)
- [uds\\_sync\\_profile](#)
- [uds\\_users](#)

### uds\_address\_books

This view provides information about address books.

**uds\_address\_books\_rv**

Name	Null?	Type
ENTERPRISE_ID	NOT NULL	NUMBER (5)
SITE_ID	NOT NULL	NUMBER (5)
TYPE	NOT NULL	VARCHAR2 (4 CHAR)
EID	NOT NULL	RAW (22)
CREATED_ON	NOT NULL	TIMESTAMP (6)
MODIFIED_ON		TIMESTAMP (6)
VISIBILITY		VARCHAR2 (1 CHAR)
PARENT_SITE_ID		NUMBER (5)
PARENT_TYPE		VARCHAR2 (4 CHAR)
PARENT_EID	NOT NULL	RAW (22)

### uds\_contacts

This view provides information about contacts.

**uds\_contacts\_rv**

Name	Null?	Type
-----	-----	-----

ENTERPRISE_ID	NUMBER(5)
SITE_ID	NUMBER(5)
TYPE	VARCHAR2(4 CHAR)
EID	RAW(22)
CREATED_ON	TIMESTAMP(6)
MODIFIED_ON	TIMESTAMP(6)
VISIBILITY	VARCHAR2(1 CHAR)
PARENT_VISIBILITY	CHAR(1 CHAR)
PARENT_SITE_ID	NUMBER(5)
PARENT_TYPE	VARCHAR2(4 CHAR)
PARENT_EID	RAW(22)
DATA_SIZE	NUMBER(38)
PEOPLE_LIST_ENTRY	VARCHAR2(1 CHAR)
BOOKMARK_SITE_ID	NUMBER(5)
BOOKMARK_TYPE	VARCHAR2(4 CHAR)
BOOKMARK_EID	RAW(22)

## uds\_external\_persons

This view provides information about external users.

### uds\_external\_persons\_rv

Name	Null?	Type
-----	-----	-----
ENTERPRISE_ID	NOT NULL	NUMBER(5)
SITE_ID	NOT NULL	NUMBER(5)
PARENT_SITE_ID		NUMBER(5)
EID	NOT NULL	RAW(22)
VISIBILITY		VARCHAR2(1)
CREATED_ON		TIMESTAMP(6)
MODIFIED_ON		TIMESTAMP(6)

## uds\_groups

This view provides information about groups.

### uds\_groups\_rv

Name	Null?	Type
-----	-----	-----
ENTERPRISE_ID	NOT NULL	NUMBER(5)
SITE_ID	NOT NULL	NUMBER(5)
EID	NOT NULL	RAW(22)
TYPE	NOT NULL	VARCHAR2(4)
VISIBILITY		VARCHAR2(1)
CREATED_ON		TIMESTAMP(6)
MODIFIED_ON		TIMESTAMP(6)

## uds\_group\_contact\_members

This view provides information about group contact members.

### uds\_group\_contact\_members\_rv

Name	Null?	Type
-----	-----	-----
ENTERPRISE_ID		NUMBER(5)
GROUP_SITE_ID		NUMBER(5)
GROUP_TYPE		VARCHAR2(4 CHAR)
GROUP_EID	NOT NULL	RAW(22)



MEMBER_SITE_ID	NUMBER(5)
MEMBER_TYPE	VARCHAR2(4 CHAR)
MEMBER_EID	NOT NULL RAW(22)

## uds\_group\_members

This view provides information about group members.

### uds\_group\_members\_rv

Name	Null?	Type
ENTERPRISE_ID	NOT NULL	NUMBER(5)
GROUP_SITE_ID	NOT NULL	NUMBER(5)
GROUP_TYPE	NOT NULL	VARCHAR2(4)
GROUP_EID	NOT NULL	RAW(22)
MEMBER_SITE_ID	NOT NULL	NUMBER(5)
MEMBER_TYPE	NOT NULL	VARCHAR2(4)
MEMBER_EID	NOT NULL	RAW(22)

## uds\_sync\_profile

This view provides information about LDAP synchronization profiles.

### uds\_sync\_profile\_rv

Name	Null?	Type
PROFILEID	NOT NULL	VARCHAR2(256)
CHANGEID		VARCHAR2(256)
CHANGETIME		DATE

## uds\_users

These views provide information about users.

### uds\_users\_rv

Name	Null?	Type
ENTERPRISE_ID	NOT NULL	NUMBER(5)
SITE_ID	NOT NULL	NUMBER(5)
PARENT_SITE_ID		NUMBER(5)
EID	NOT NULL	RAW(22)
VISIBILITY		VARCHAR2(1)
IS_EXTERNAL_INBOX		VARCHAR2(1)
IS_EXTENDED_ENTERPRISE_USER		VARCHAR2(1)
CREATED_ON		TIMESTAMP(6)
MODIFIED_ON		TIMESTAMP(6)

### uds\_users\_s\_rv

Name	Null?	Type
ENTERPRISE_ID	NOT NULL	NUMBER(5)
SITE_ID		NUMBER(5)
TYPE		CHAR(4)
EID		RAW(22)
LOGIN_ID		VARCHAR2(1000 CHAR)

## Creating Schema with Access Only to Oracle Beehive Views

Suppose you want to create an application that allows non-administrators (such as upper-level management) to view statistical reports on information from Oracle Business Views. In this situation, create a database schema that has access only to Oracle Beehive Views; your application would access Oracle Beehive views with this schema. This will prevent non-administrators accessing or modifying sensitive data in your Oracle Beehive database.

Follow these steps to create a database schema that has access only to Oracle Business Views:

1. Obtain the site ID of your Oracle Beehive deployment. In the following example, the side ID is 21734.

```
beectl list_properties --component _CURRENT_SITE
```

```
-----+-----
Property name      | Property value
-----+-----
Alias              | _CURRENT_SITE
...
...
SiteId             | 21734
...
...
-----+-----
```

2. As a user with SYSDBA privileges, create a new tablespace in the Oracle Beehive Database. This example creates a tablespace named BEE\_RVIEWS\_TBLSPC:

```
sqlplus /nolog
```

```
SQL> connect SYS/<Password of SYS user> as SYSDBA
Connected
SQL> create tablespace BEE_RVIEWS_TBLSPC;
```

3. As a user with SYSDBA privileges, run the `manage_reporting_user.sql` script.

In the following example, BEE\_RVIEWS is the name of the new schema (and in this example, the name of the tablespace you created previously). Welcome1 is the password for the new schema. TEMP is the name of the default temporary tablespace (you may specify another temporary tablespace). 21734 is the site ID of your Oracle Beehive deployment. (Line breaks have been inserted in the following example for clarity.)

```
SQL> <Oracle Beehive home>/beehive/db/framework/
@manage_reporting_user.sql create BEE_RVIEWS Welcome1 BEE_RVIEWS_TBLSPC
TEMP 21734
```