Oracle® Beehive

Application Developer's Guide

Release 1 (1.2 and 1.3)

E10484-04

August 2008

ORACLE®

Oracle Beehive Application Developer's Guide, Release 1 (1.2 and 1.3)

E10484-04

Primary Author: Raymond Gallardo

Contributing Author: Jamie Rancourt

Contents

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, with good usability, to the disabled community. 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.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

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

1

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 BeehiveOracle BeehiveOracle 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.

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

2

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.

3

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 A, "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).

Refer to "Oracle Beehive Basic Web Service Samples" found in the Oracle Beehive Web site on Oracle Technology Network for examples of how to search with filters.

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

Refer to "Oracle Beehive Basic Web Service Samples" found in the Oracle Beehive Web site on Oracle Technology Network for an example of GeneralArtifactService that demonstrates how to search with FindArtifacts.

A

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 A–1 AddressBookService

Method	Return Type
GetAddressBooks(tns:WSEntity uID, tns:WSEntity[] abookList, tns:WSFilter abFilter)	tns:WSAddressBook[]
UpdateAddressBook(tns:WSEntity uID, tns:WSEntity parentAB, tns:WSAddressBook abook)	tns:WSAddressBook
DeleteAddressBook(tns:WSEntity abID)	tns:WSResultStatus
GetContact(tns:WSEntity[] cID, tns:WSFilter contactFilter)	tns:WSContact[]
GetAllContacts(tns:WSEntity uID, tns:WSEntity abook, xsd:string contactType, tns:WSFilter contactFilter)	tns:WSContact[]
UpdateContact(tns:WSEntity uID, tns:WSEntity abook, tns:WSContact contact)	tns:WSContact
DeleteContacts(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 A–2 CalendarService

Method	Return Type
GetCalendar(tns:WSEntity uID, tns:WSEntity[] calIDList, tns:WSFilter calFilter)	tns:WSCalendar[]
UpdateCalendar(tns:WSEntity uID, tns:WSCalendar calendar)	tns:WSCalendar

Table A–2 (Cont.) CalendarService

Method	Return Type
DeleteCalendar(tns:WSEntity calID)	tns:WSResultStatus
GetEvents(tns:WSEntity uID, tns:WSEntity calID, xsd:dateTime startTime, xsd:dateTime endTime, tns:WSFilter eventFilter)	tns:WSCalendarEvent[]
GetAllEvents(tns:WSEntity uID, xsd:dateTime startTime, xsd:dateTime endTime, tns:WSFilter eventFilter)	tns:WSCalendarEvent[]
UpdateEvent(tns:WSEntity uID, tns:WSEntity calID, tns:WSCalendarEvent calEvent)	tns:WSCalendarEvent
DeleteEvents(tns:WSEntity[] eventIDList)	tns:WSResultStatus[]
GetRecurringEventSeries(tns:WSEntity uID, tns:WSEntity calID, tns:WSEntity eventSeriesID, tns:WSFilter eventSeriesFilter)	tns:WSEventSeries[]
DeleteRecurringEventSeries(tns:WSEntity eventSeriesID)	tns:WSResultStatus
GetInvitations(tns:WSEntity uID, tns:WSEntity calID, tns:WSEntity eventID, xsd:dateTime startTime, xsd:dateTime endTime, tns:WSFilter inviteFilter)	tns:WSCalendarInvitation[]
GetAllInvitations(tns:WSEntity uID, xsd:dateTime startTime, xsd:dateTime endTime, tns:WSFilter inviteFilter)	tns:WSCalendarInvitation[]
UpdateInvitation(tns:WSCalendarInvitation ci)	tns:WSCalendarInvitation
GetIsBusy(tns:WSEntity[] uIDList, xsd:dateTime time)	xsd:boolean[]
GetFreeBusy(tns:WSEntity[] uIDList, xsd:dateTime startTime, xsd:dateTime endTime)	tns:WSFreeBusyInterval[]
GetReminders(tns:WSEntity uID, tns:WSEntity artifactID, tns:WSFilter remFilter)	tns:WSReminder[]
UpdateReminder(tns:WSEntity uID, tns:WSReminder rem)	tns:WSReminder
DeleteReminder(tns:WSReminder remID)	tns:WSResultStatus
DeleteTasks(tns:WSEntity[] taskIdList)	tns:WSResultStatus
DeleteTaskLists(tns:WSEntity[] taskListIds)	tns:WSResultStatus
GetTaskLists(tns:WSEntity uID, tns:WSEntity[] taskListIDs)	tns:WSTaskList[]
GetTasks(tns:WSEntity uID, tns:WSEntity taskListID, tns:WSEntity[] taskIDs)	tns:WSTask
UpdateTask(tns:WSTask task)	tns:WSTask
UpdateTaskList(tns:WSEntity uID, tns:WSTaskList taskList)	tns:WSTaskList
UpdateRecurringEventSeries(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 A–3 ConferenceService

Method	Return Type
DeleteConferences(tns:WSEntity[] confIDList)	tns:WSResultStatus[]

Table A–3 (Cont.) ConferenceService

Method	Return Type
DeleteTemplate(tns:WSEntity cTemplateID)	tns:WSResultStatus[]
GetConferences(tns:WSEntity uID, xsd:dateTime startTime, xsd:dateTime endTime, tns:WSEntity[] confIDList, tns:WSFilter confFilter)	tns:WSConference[]
GetEndedSessions(tns:WSEntity[] uID, tns:WSEntity[] confID)	tns:WSConferenceSession[]
GetLogEntries(tns:WSEntity uID, tns:WSEntity confID, tns:WSParticipant p)	tns:WSConferenceLogEntry[]
GetRunningSession(tns:WSEntity uID, tns:WSEntity confID, tns:WSFilter confFilter)	tns:WSConferenceSession
GetTemplate(tns:WSEntity uID, tns:WSEntity confID, tns:WSFilter templateFilter)	tns:WSConferenceTemplate
UpdateConference(tns:WSEntity uID, tns:WSConference conf)	tns:WSConference
UpdateConferenceSession(tns:WSEntity uID, tns:WSConferenceSession cs)	tns:WSConferenceSession
UpdateTemplate(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 A–4 DeviceService

Method	Return Type
DeleteDevices(tns:WSEntity[] deviceIDList)	tns:WSResultStatus[]
GetDevicePresence(tns:WSEntity uID, tns:WSEntity devID, tns:WSFilter devFilter)	tns:WSPresence
GetDevices(tns:WSEntity uID, tns:WSEntity[] deviceIDList, tns:WSFilter devFilter)	tns:WSDevice[]
UpdateDevice(tns:WSEntity uID, tns:WSDevice dev)	tns:WSDevice

DiscussionForumService Methods

Table A–5 DiscussionForumService

Method	Return Type
DeleteDiscussionForums(tns:WSEntity[] forumIDList)	tns:WSResultStatus[]
DeleteMessages(tns:WSEntity[] dmIDList)	tns:WSResultStatus[]
DeleteTopic(tns:WSEntity topicID)	tns:WSResultStatus
GetDiscussionForums(tns:WSEntity wID, tns:WSFilter dfFilter)	tns:WSForum[]
GetLastPost(tns:WSEntity wID, tns:WSEntity fID, tns:WSEntity topicID, tns:WSEntity dmFilter)	tns:WSDiscussionMessage

Table A–5 (Cont.) DiscussionForumService

Method	Return Type
GetMessages(tns:WSEntity wID, tns:WSEntity fID, tns:WSTopic topic, tns:WSEntity[] dmIDList, tns:WSEntity dmFilter)	tns:WSDiscussionMessage[]
GetTopics(tns:WSEntity wID, tns:WSEntity forumID, tns:WSFilter forumFilter)	tns:WSTopic[]
PostMessage(tns:WSEntity wID, tns:WSEntity fID, tns:WSTopic topic, tns:WSDiscussionMessage dm)	tns:WSResultStatus
UpdateDiscussionForum(tns:WSEntity parentForum, tns:WSForum forum)	tns:WSForum
UpdateTopic(tns:WSEntity wID, tns:WSEntity fID, tns:WSTopic topic)	tns:WSTopic

DocumentService Methods

Provides methods for managing documents in personal or team workspaces.

Table A–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:WSDocument
DeleteDocuments(tns:WSEntity[] docIDList)	tns:WSResultStatus[]

FolderService Methods

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

Table A–7 FolderService

Method	Return Type
GetFolders(tns:WSEntity uID, tns:WSEntity[] fIDList, tns:WSFilter folderFilter)	tns:WSFolder[]
GetSubFolders(tns:WSEntity containerID, tns:WSFilter folderFilter)	tns:WSFolder[]
UpdateFolder(tns:WSEntity uID, tns:WSFolder folder)	tns:WSFolder
DeleteFolders(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 A–8 GeneralArtifactService

Method	Return Type
CopyArtifact(tns:WSEntity newParent, tns:WSArtifact artifact)	tns:WSArtifact
MoveArtifact(tns:WSEntity newParent, tns:WSArtifact artifact)	tns:WSArtifact
GetTags(tns:WSEntity uID, tns:WSFilter tagFilter)	tns:WSTag[]
UpdateTag(tns:WSEntity uID, tns:WSTag tag)	tns:WSTag
GetBonds(tns:WSEntity entityID, tns:WSFilter bondFilter)	tns:WSBond[]
UpdateBond(tns:WSBond bond)	tns:WSBond
DeleteBonds(tns:WSEntity[] entityList)	tns:WSResultStatus[]
GetLinks(tns:WSEntity entityID, tns:WSFilter linkFilter)	tns:WSLink[]
UpdateLink(tns:WSLink link)	tns:WSLink
DeleteLinks(tns:WSEntity[] linkList)	tns:WSResultStatus[]
GetExternalArtifacts(tns:WSEntity[] externalArtifactIDList)	tns:WSExternalArtifact[]
UpdateExternalArtifact(tns:WSEntity parent, tns:WSExternalArtifact externalArtifact)	tns:WSExternalArtifact
DeleteExternalArtifacts(tns:WSEntity[] externalArtifactIDList)	tns:WSResultStatus[]
GetLocks(tns:WSEntity entity)	tns:WSLock[]
UpdateLock(tns:WSLock lock)	tns:WSLock
DeleteLocks(tns:WSEntity[] lockList)	tns:WSResultStatus[]
DeleteTags(tns:WSEntity[] tagIDList)	tns:WSResultStatus[]
UpdateSubscription(tns:WSEntity eID, tns:WSSubscription sub)	tns:WSSubscription
DeleteSubscriptions(tns:WSEntity[] subIDList)	tns:WSResultStatus[]
GetNotifications(tns:WSEntity uID, tns:WSEntity[] notIDList, tns:WSFilter notFilter)	tns:WSNotification[]
SendNotification(tns:WSEntity uID, tns:WSNotification not)	tns:WSResultStatus
DeleteNotifications(tns:WSEntity[] notIDList)	tns:WSResultStatus[]
GetSubscription(tns:WSEntity uID, tns:WSEntity notID, tns:WSFilter subFilter)	tns:WSSubscription
GetSubscriptionList(tns:WSEntity eID, tns:WSFilter subFilter)	tns:WSSubscription[]
GetSubscriptionTemplates(xsd:WSSubscriptionType subType)	tns:WSSubscriptionTemplate[]
FindArtifacts(xsd:string searchString, tns:WSFilter filter, xsd:int maxCount, xsd:int start, xsd:int end)	tns:WSSearchResult[]
FindTags(xsd:string[] tagNameList)	tns:WSTag[]
FindAllTags(tns:WSEntity uID, tns:WSEntity artifactType)	tns:WSTag[]

Table A–8 (Cont.) GeneralArtifactService

Method	Return Type
FindCategories(xsd:string[] categoryNameList)	tns:WSCategory[]
FindAllCategories(tns:WSEntity uID, tns:WSEntity artifactType)	tns:WSCategory[]
FindRelatedArtifacts(tns:WSEntity artifactID, xsd:string artifactType)	tns:WSArtifact[]
FindAllArtifactsWithTags(tns:WSEntity uID, xsd:string artifactType, xsd:string tagName)	tns:WSArtifact[]
FindArtifactsModifiedSince(tns:WSEntity uID, xsd:dateTime date, xsd:string artifactType)	tns:WSArtifact[]
FindArtifactsWithTags(tns:WSEntity uID, xsd:string artifactType, xsd:string tagName)	tns:WSArtifact[]
FindArtifactsLargerThan(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 A–9 GroupService

Method	Return Type
GetGroup(tns:WSEntity uID, tns:WSFilter groupFilter)	tns:WSGroup[]
DeleteGroup(tns:WSEntity gID)	tns:WSResultStatus
UpdateGroup(tns:WSEntity uID, tns:WSGroup grp)	tns:WSGroup

MembershipService Methods

Provides methods for directly managing the memberships of groups and workspaces without having to load the contents or other attributes of the object.

Table A–10 MembershipService

Method	Return Type
GetUsers(tns:WSEntity[] uIDList, tns:WSFilter userFilter)	tns:WSUser[]
GetMembership(tns:WSEntity uID, tns:WSFilter memberFilter)	tns:WSUser[]
UpdateMembership(tns:WSEntity uID, tns:WSUser user)	tns:WSUser
DeleteMembership(tns:WSEntity uID, tns:WSEntity userID)	tns:WSResultStatus
GetDelegatedPrincipals(tns:WSEntity uID, tns:WSEntity userFilter)	tns:WSUser[]
WhoAmI( )	tns:WSEntity

MessageService Methods

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

Table A–11 MessageService

Method	Return Type
GetContentData(tns:WSEntity emailID, xsd:base64Binary partIdentifier, xsd:int size)	xsd:base64Binary
GetMessageBoxes(tns:WSEntity uID, tns:WSEntity[] mdIDList, tns:WSFilter msgFilter)	tns:WSMessageBox[]
UpdateMessageBox(tns:WSEntity uID, tns:WSEntity parentMBid, tns:WSMessageBox mbox)	tns:WSMessageBox
DeleteMessageBoxes(tns:WSEntity[] mbIDList)	tns:WSResultStatus[]
GetEmailMessages(tns:WSEntity uID, tns:WSEntity msgBoxID, xsd:string msgType, tns:WSEntity[] msgIDList, tns:WSFilter msgFilter)	tns:WSMessage[]
GetAllMessageHeaders(tns:WSEntity uID, tns:WSEntity msgBoxID, xsd:string msgType, tns:WSFilter msgFilter)	tns:WSMessageHeader[]
GetNewMessageHeaders(tns:WSEntity uID, tns:WSEntity msgBoxID, xsd:string msgType, tns:WSFilter msgFilter)	tns:WSMessageHeader[]
GetUnreadMessages(tns:WSEntity uID, tns:WSEntity msgBoxID, xsd:string msgType, tns:WSFilter msgFilter)	tns:WSMessageHeader[]
UpdateMessageHeaders(tns:WSMessageHeader[] msgHdrList)	tns:WSMessageHeader[]
SendMessage(tns:WSEntity uID, tns:WSMessage msg)	tns:WSResultStatus
DeleteEmailMessages(tns:WSEntity[] msgIDList)	tns:WSResultStatus[]
SendInstantMessage(tns:WSInstantMessage instantMsg)	tns:WSResultStatus
GetInstantMessage(xsd:string clientSideID, xsd:string conversationID)	tns:WSInstantMessage[]
SaveDraft(tns:WSEntity uID, tns:WSMessage msg)	tns:WSResultStatus

PreferenceService Methods

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

Table A–12 PreferenceService

Method	Return Type
GetPreferenceList(tns:WSEntity uID, tns:WSFilter prefFilter)	tns:WSPreferenceProfile[]
UpdatePreference(tns:WSEntity prefHolder, tns:WSPreferenceProfile prefProfile)	tns:WSPreferenceProfile
UpdatePreferenceProfile(tns:WSEntity uID, tns:WSPreferenceProfile prefProf)	tns:WSPreferenceProfile
DeletePreferenceProfile(tns:WSEntity[] prefProfIDList)	tns:WSResultStatus[]

PresenceService Methods

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

Table A–13 PresenceService

Method	Return Type
GetPresence(tns:WSEntity[] uIDList, tns:WSFilter presenceFilter)	tns:WSPresence[]
UpdatePresence(tns:WSPresence presence)	tns:WSPresence
SubscribePresence(tns:WSEntity uID)	tns:WSPresence
GetImBuddyList(tns:WSEntity uID)	tns:WSBuddyList

WorkspaceService Methods

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

Table A–14 WorkspaceService

Method	Return Type
GetWorkspaces(tns:WSEntity uID, xsd:string wspType, tns:WSFilter wspFilter)	tns:WSWorkspace[]
GetWorkspaceTemplates(tns:WSEntity uID, tns:WSFilter wspTemplateFilter)	tns:WSWorkspaceTemplate[]
GetTrashItems(tns:WSEntity uID)	tns:WSArtifact[]
UnDeleteItems(tns:WSEntity[] artifactList)	tns:WSResultStatus[]
PurgeTrash(tns:WSEntity[] wIDList)	tns:WSResultStatus[]
DeleteWorkspaces(tns:WSEntity[] wIDList)	tns:WSResultStatus[]
UpdateWorkspace(tns:WSEntity uID, tns:WSWorkspace wksp)	tns:WSWorkspace

AddressBookService

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

GetAddressBooks

Table A–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 A–16 GetAddressBooks Outputs

Type	Description
tns:WSAddressBook[]	If uID is null, it returns an array of address books for the user currently logged in. If uID denotes a user, then an array of address books for the specified user is returned. 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.

Effects None

Exceptions None

UpdateAddressBook

Table A–17 UpdateAddressBook Inputs

Input Name	Type	Description
uID	tns:WSEntity	User or workspace. The logged in user is assumed if it is null.
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 A–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 A–19 DeleteAddressBook Inputs

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

Table A–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 A–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 A–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 A–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 A–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 A–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.
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 A–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 A–27 DeleteContacts Inputs

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

Table A–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.

GetCalendar

Table A–29 GetCalendar 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 A–30 GetCalendar Outputs

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

Effects None.

Exceptions None

UpdateCalendar

Table A–31 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 A–32 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 A–33 DeleteCalendar Inputs

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

Table A–34 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 A–35 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.
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, than 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 A–36 GetEvents Outputs

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

Effects None.

Exceptions None

GetAllEvents

Table A–37 GetAllEvents Inputs

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group or workspace identifier; if null, the logged in user is assumed.
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 startTime is specified, than 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 A–38 GetAllEvents Outputs

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

Effects None.

Exceptions None

UpdateEvent

Table A–39 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 A–40 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 A–41 DeleteEvents Inputs

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

Table A–42 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

GetRecurringEventSeries

Table A–43 GetRecurringEventSeries 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.
eventSeriesID	tns:WSEntity	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 A–44 GetRecurringEventSeries Outputs

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

Effects None.

Exceptions None

DeleteRecurringEventSeries

Table A–45 DeleteRecurringEventSeries Inputs

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

Table A–46 DeleteRecurringEventSeries 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

GetInvitations

Table A–47 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.
eventID	tns:WSEntity	Event identifier. 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 A–48 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

GetAllInvitations

Table A–49 GetAllInvitations Inputs

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group, or workspace identifier; if null, the logged in user is assumed.
startTime	xsd:dateTime	The start of a valid time interval. If null, all invitations are considered.
endTime	xsd:dateTime	The end of a valid time interval. If null, all invitations are considered.

Table A–49 (Cont.) GetAllInvitations Inputs

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

Table A–50 GetAllInvitations Outputs

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

Effects None.

Exceptions None

UpdateInvitation

Table A–51 UpdateInvitation Inputs

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

Table A–52 UpdateInvitation Outputs

Type	Description
tns:WSCalendarInvitation	Updated calendar event

Effects The calendar invitation is updated according to ci.

Exceptions None

GetIsBusy

Table A–53 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 A–54 GetIsBusy Outputs

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

Effects None.

Exceptions None

GetFreeBusy

Table A–55 GetFreeBusy Inputs

Input Name	Type	Description
uIDList	tns:WSEntity[]	List of valid users 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 A–56 GetFreeBusy Outputs

Type	Description
tns:WSFreeBusyInterval[]	List of freebusy information for the given users and time interval.

Effects None.

Exceptions None

GetReminders

Table A–57 GetReminders Inputs

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group or workspace, identifier; if null, the logged in user is assumed.
artifactID	tns:WSEntity	None
remFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

Table A–58 GetReminders Outputs

Type	Description
tns:WSReminder[]	List of reminders for the user/group/workspace.

Effects None.

Exceptions None

UpdateReminder

Table A–59 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 A–60 UpdateReminder Outputs

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

Effects No effects

Exceptions None

DeleteReminder

Table A–61 DeleteReminder Inputs

Input Name	Type	Description
remID	tns:WSReminder	Reminder existing in the system

Table A–62 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 list of reminders is deleted, subject to access control.

Exceptions None

DeleteTasks

Table A–63 DeleteTasks Inputs

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

Table A–64 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 A–65 DeleteTaskLists Inputs

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

Table A–66 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 A–67 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 tasklist identifiers for uID. If null, then all tasklists is assumed.

Table A–68 GetTaskLists Outputs

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

Effects None.

Exceptions None

GetTasks

Table A–69 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 tasklist identifier for uID. If null, then default tasklist is assumed.
taskIDs	tns:WSEntity[]	Specific task identifier(s) for uID to retrieve. If null, then all task are retrieved.

Table A–70 GetTasks Outputs

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

Effects None.

Exceptions None

UpdateTask

Table A–71 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 A–72 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 A–73 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	Tasklist to be updated. If the taskLists's taskListInfo's artifactID's id field is null, this will be a newly created task list.

Table A–74 UpdateTaskList Outputs

Type	Description
tns:WSTaskList	Updated (or created) tasklist.

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

Exceptions None

UpdateRecurringEventSeries

Table A–75 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 A–76 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 A–77 DeleteConferences Inputs

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

Table A–78 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 A–79 DeleteTemplate Inputs

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

Table A–80 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 A–81 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 A–82 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 A–83 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 A–84 GetEndedSessions Outputs

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

Effects None

Exceptions None

GetLogEntries

Table A–85 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 A–85 (Cont.) GetLogEntries Inputs

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

Table A–86 GetLogEntries Outputs

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

Effects None

Exceptions None

GetRunningSession

Table A–87 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 A–88 GetRunningSession Outputs

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

Effects None

Exceptions None

GetTemplate

Table A–89 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 A–90 GetTemplate Outputs

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

Effects None

Exceptions None

UpdateConference

Table A–91 UpdateConference Inputs

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group, or workspace; if null, the logged in user is assumed.
conf	tns:WSConference	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 A–92 UpdateConference Outputs

Type	Description
tns:WSConference	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 A–93 UpdateConferenceSession Inputs

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group, or workspace; if null, the logged in user is assumed.
cs	tns:WSConferenceSession	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 A–94 UpdateConferenceSession Outputs

Type	Description
tns:WSConferenceSession	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 A–95 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 A–96 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 A–97 DeleteDevices Inputs

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

Table A–98 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 A–99 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 A–99 (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 A–100 GetDevicePresence Outputs

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

Effects None

Exceptions None

GetDevices

Table A–101 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 A–102 GetDevices Outputs

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

Effects None

Exceptions None

UpdateDevice

Table A–103 UpdateDevice Inputs

Input Name	Type	Description
uID	tns:WSEntity	Valid user. If null, the logged in user is assumed.
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 A–104 UpdateDevice Outputs

Type	Description
tns:WSDevice	Newly created or updated device.

Effects None

Exceptions None

DiscussionForumService

DeleteDiscussionForums

Table A–105 DeleteDiscussionForums Inputs

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

Table A–106 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 A–107 DeleteMessages Inputs

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

Table A–108 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 A–109 DeleteTopic Inputs

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

Table A–110 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 A–111 GetDiscussionForums Inputs

Input Name	Type	Description
wID	tns:WSEntity	Workspace; it cannot be null.
dfFilter	tns:WSFilter	Optional parameter specifying a predicate involving filterable attributes to reduce the size of the returned list.

Table A–112 GetDiscussionForums Outputs

Type	Description
tns:WSForum[]	List of discussion forums in the workspace specified by wID.

Effects None

Exceptions None

GetLastPost

Table A–113 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 A–114 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 A–115 GetMessages Inputs

Input Name	Type	Description
wID	tns:WSEntity	Team workspace. If null, then fID cannot be null.

Table A–115 (Cont.) GetMessages Inputs

Input Name	Type	Description
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 A–116 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 A–117 GetTopics Inputs

Input Name	Type	Description
wID	tns:WSEntity	Team workspace. If null, forumID cannot be null.
forumID	tns:WSEntity	Forum. If null, then wID cannot be null. If specified, wID is ignored.
forumFilter	tns:WSFilter	Optional parameter specifying a predicate involving filterable attributes to reduce the size of the returned list.

Table A–118 GetTopics Outputs

Type	Description
tns:WSTopic[]	If forumID is specified, list of topics in the forum. If forumID is null, list of default announcements in the team workspace.

Effects None

Exceptions None

PostMessage

Table A–119 PostMessage Inputs

Input Name	Type	Description
wID	tns:WSEntity	Team workspace. If null, fID cannot be null.
fID	tns:WSEntity	Forum. If null, wID cannot be null. If specified, wID is ignored.

Table A–119 (Cont.) PostMessage Inputs

Input Name	Type	Description
topic	tns:WSTopic	Existing topic or new topic related to an artifact (ensure that relatedArtifact is set).
dm	tns:WSDiscussionMessage	New discussion message, its entity ID is null.

Table A–120 PostMessage Outputs

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

Effects The message specified by dm is posted to the topic (specified by topic) in the forum (specified by fID). If fID is null, the message is posted as a default announcement in the team workspace. If the topic is new and related to the artifact, it is also created.

Exceptions None

UpdateDiscussionForum

Table A–121 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 A–122 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 A–123 UpdateTopic Inputs

Input Name	Type	Description
wID	tns:WSEntity	Team workspace. If null, fID cannot be null.
fID	tns:WSEntity	Forum. If null, wID cannot be null. If specified, wID is ignored.
topic	tns:WSTopic	New or existing topic.

Table A–124 UpdateTopic Outputs

Type	Description
tns:WSTopic	New or updated topic.

Effects If the topic is new, it is created in the specified forum. If fID is null, then the topic is created as a default announcement in the specified team workspace. If the topic exists, it is updated. In this case, fID and wID are ignored.

Exceptions None

DocumentService

Provides methods for managing documents in personal or team workspaces.

CancelCheckoutDocument

Table A–125 CancelCheckoutDocument Inputs

Input Name	Type	Description
docID	tns:WSEntity	None

Void return type

Effects None.

Exceptions None

CheckinDocument

Table A–126 CheckinDocument Inputs

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

Table A–127 CheckinDocument Outputs

Type	Description
tns:WSVersion	None

Effects None.

Exceptions None

CheckoutDocument

Table A–128 CheckoutDocument Inputs

Input Name	Type	Description
docID	tns:WSEntity	None

Table A–128 (Cont.) CheckoutDocument Inputs

Input Name	Type	Description
checkoutComment	xsd:string	None

Table A–129 CheckoutDocument Outputs

Type	Description
tns:WSVersion	None

Effects None.

Exceptions None

GetDocuments

Table A–130 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 A–131 GetDocuments Outputs

Type	Description
tns:WSDocument[]	List of documents retrieved

Effects None.

Exceptions None

GetDocumentsInFolder

Table A–132 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 A–133 GetDocumentsInFolder Outputs

Type	Description
tns:WSDocument[]	List of documents retrieved

Effects None.

Exceptions None

GetContent

Table A–134 GetContent Inputs

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

Table A–135 GetContent Outputs

Type	Description
tns:WSContent[]	Content of documents retrieved

Effects None.

Exceptions None

GetContentOfDocs

Table A–136 GetContentOfDocs Inputs

Input Name	Type	Description
contentIDList	tns:WSEntity[]	None

Table A–137 GetContentOfDocs Outputs

Type	Description
tns:WSContent[]	None

Effects None.

Exceptions None

UpdateDocument

Table A–138 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

Table A–139 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 A–140 DeleteDocuments Inputs

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

Table A–141 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 A–142 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 A–143 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 A–144 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 A–145 GetSubFolders Outputs

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

Effects None.

Exceptions None

UpdateFolder

Table A–146 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 A–147 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 A–148 DeleteFolders Inputs

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

Table A–149 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 A–150 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 A–151 CopyArtifact Outputs

Type	Description
tns:WSArtifact	New copy of artifact

Effects Copies artifact to parent specified by newParent.

Exceptions None

MoveArtifact

Table A–152 MoveArtifact Inputs

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

Table A–153 MoveArtifact Outputs

Type	Description
tns:WSArtifact	Artifact that was moved

Effects Moves artifact to parent specified by newParent.

Exceptions None

GetTags

Table A–154 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 A–155 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 A–156 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 A–157 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 A–158 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 A–159 GetBonds Outputs

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

Effects None.

Exceptions None

UpdateBond

Table A–160 UpdateBond Inputs

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

Table A–161 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 A–162 DeleteBonds Inputs

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

Table A–163 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

GetLinks

Table A–164 GetLinks Inputs

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

Table A–165 GetLinks Outputs

Type	Description
tns:WSLink[]	Links associated with entityID, subject to linkFilter.

Effects None.

Exceptions None

UpdateLink

Table A–166 UpdateLink Inputs

Input Name	Type	Description
link	tns:WSLink	Valid link

Table A–167 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 A–168 DeleteLinks Inputs

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

Table A–169 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 A–170 GetExternalArtifacts Inputs

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

Table A–171 GetExternalArtifacts Outputs

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

Effects None

Exceptions None

UpdateExternalArtifact

Table A–172 UpdateExternalArtifact Inputs

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

Table A–173 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 A–174 DeleteExternalArtifacts Inputs

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

Table A–175 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

GetLocks

Table A–176 GetLocks Inputs

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

Table A–177 GetLocks Outputs

Type	Description
tns:WSLock[]	List of locks on the entity specified in the input.

Effects None

Exceptions None

UpdateLock

Table A–178 UpdateLock Inputs

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

Table A–179 UpdateLock Outputs

Type	Description
tns:WSLock	Updated or created lock.

Effects None

Exceptions None

DeleteLocks

Table A–180 DeleteLocks Inputs

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

Table A–181 DeleteLocks Outputs

Type	Description
tns:WSResultStatus[]	None

Effects None

Exceptions None

DeleteTags

Table A–182 DeleteTags Inputs

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

Table A–183 DeleteTags Outputs

Type	Description
tns:WSResultStatus[]	None

Effects None

Exceptions None

UpdateSubscription

Table A–184 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 A–185 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 A–186 DeleteSubscriptions Inputs

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

Table A–187 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

GetNotifications

Table A–188 GetNotifications Inputs

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group or workspace identifier; if null the logged in user is assumed.
notIDList	tns:WSEntity[]	Set of notification identifiers. If specified, uID is ignored.
notFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

Table A–189 GetNotifications Outputs

Type	Description
tns:WSNotification[]	List of notifications in notIDList, or for the user/group/workspace.

Effects None

Exceptions None

SendNotification

Table A–190 SendNotification Inputs

Input Name	Type	Description
uID	tns:WSEntity	Valid user, group or workspace identifier; if null the logged in user is assumed.
not	tns:WSNotification	New notification

Table A–191 SendNotification 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 notification to the specified user.

Exceptions None

DeleteNotifications

Table A–192 DeleteNotifications Inputs

Input Name	Type	Description
notIDList	tns:WSEntity[]	List of IDs of notifications to be deleted

Table A–193 DeleteNotifications 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 notifications are deleted from the system, subject to access control and privileges.

Exceptions None

GetSubscription

Table A–194 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 A–195 GetSubscription Outputs

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

Effects None

Exceptions None

GetSubscriptionList

Table A–196 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 A–197 GetSubscriptionList Outputs

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

Effects None

Exceptions None

GetSubscriptionTemplates

Table A–198 GetSubscriptionTemplates Inputs

Input Name	Type	Description
subType	xsd:WSSubscriptionType	None

Table A–199 GetSubscriptionTemplates Outputs

Type	Description
tns:WSSubscriptionTemplate[]	None

Effects None

Exceptions None

FindArtifacts

Table A–200 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 A–201 FindArtifacts Outputs

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

Effects None.

Exceptions None

FindTags

Table A–202 FindTags Inputs

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

Table A–203 FindTags Outputs

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

Effects None.

Exceptions None

FindAllTags

Table A–204 FindAllTags Inputs

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

Table A–205 FindAllTags Outputs

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

Effects None.

Exceptions None

FindCategories

Table A–206 FindCategories Inputs

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

Table A–207 FindCategories Outputs

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

Effects None.

Exceptions None

FindAllCategories

Table A–208 FindAllCategories Inputs

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

Table A–209 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 A–210 FindRelatedArtifacts Inputs

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

Table A–211 FindRelatedArtifacts Outputs

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

Effects None.

Exceptions None

FindAllArtifactsWithTags

Table A–212 FindAllArtifactsWithTags Inputs

Input Name	Type	Description
uID	tns:WSEntity	None
artifactType	xsd:string	None
tagName	xsd:string	None

Table A–213 FindAllArtifactsWithTags Outputs

Type	Description
tns:WSArtifact[]	None

Effects No effects

Exceptions None

FindArtifactsModifiedSince

Table A–214 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 A–215 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 A–216 FindArtifactsWithTags Inputs

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

Table A–217 FindArtifactsWithTags Outputs

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

Effects None.

Exceptions None

FindArtifactsLargerThan

Table A–218 FindArtifactsLargerThan Inputs

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

Table A–219 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 A–220 GetGroup Inputs

Input Name	Type	Description
uID	tns:WSEntity	Group or workspace identifier. This parameter cannot be null.
groupFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

Table A–221 GetGroup Outputs

Type	Description
tns:WSGroup[]	If uID is a group, the information about that group is returned as a singleton array. If uID is a workspace, all groups defined in that workspace are returned.

Effects None

Exceptions None

DeleteGroup

Table A–222 DeleteGroup Inputs

Input Name	Type	Description
gID	tns:WSEntity	Group identifier.

Table A–223 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 A–224 UpdateGroup Inputs

Input Name	Type	Description
uID	tns:WSEntity	Group or workspace identifier. If it is null then grp denotes a top-level group, otherwise it must indicate the identifier of the parent (group or workspace) of grp.
grp	tns:WSGroup	Detailed information about a new or existing group. If the group is new, then the entity ID in grp is null. If the group already exists, grp must contain the entity ID of the group. If this parameter is specified, then uID is not a required parameter unless there is a parent-child relationship between uID and grp.

Table A–225 UpdateGroup Outputs

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

Effects If the entity ID in grp is null, then a new group is created with the information provided in grp. The new group is a top level group if uID is null, otherwise it is created as a subgroup of the group or workspace denoted by uID.

If the entity ID in grp is not null, then that existing group is updated according to the group information provided in grp. If uID is null, then grp is a top level group, otherwise the uID must denote the parent group or workspace of grp.

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

MembershipService

Provides methods for directly managing the memberships of groups and workspaces without having to load the contents or other attributes of the object.

GetUsers

Table A–226 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 A–227 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

GetMembership

Table A–228 GetMembership Inputs

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

Table A–229 GetMembership Outputs

Type	Description
tns:WSUser[]	If the input is null, returns user info for the user currently logged in. If the input is a user ID, returns user info for the specified user. If the input is a group ID, returns an array of all the users in the group. If input is a workspace ID, returns an array of all users in the workspace.

Effects None.

Exceptions None

UpdateMembership

Table A–230 UpdateMembership Inputs

Input Name	Type	Description
uID	tns:WSEntity	Existing group or workspace, or null if the user is being updated.
user	tns:WSUser	New or updated information about the user/member.

Table A–231 UpdateMembership Outputs

Type	Description
tns:WSUser	The updated information for the user. For example, if the user has been successfully added to a group, the return parameter reflects the new information.

Effects If uID is null, then the user information is updated according to the parameter user.

If uID specifies a group, then the user denoted by the second parameter is added to the group if the member does not already exist in it or replaces the existing member information for the user in the group, such as a new role.

If uID specifies a workspace, then the user denoted by the second parameter is added to the workspace if the member does not already exist in it, or replaces the existing member information for the user in the workspace.

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

DeleteMembership

Table A–232 DeleteMembership Inputs

Input Name	Type	Description
uID	tns:WSEntity	Group or workspace.
userID	tns:WSEntity	Handle to a user/member. The logged in user is assumed if it is null.

Table A–233 DeleteMembership 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 If uID is a group ID, then the user denoted by userID is removed from the group.

If uID is a workspace ID, then the user is removed from the workspace.

Exceptions None

GetDelegatedPrincipals

Table A–234 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 A–235 GetDelegatedPrincipals Outputs

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

Effects None

Exceptions None

WhoAmI

Table A–236 WhoAmI Outputs

Type	Description
tns:WSEntity	Entity information for the logged in User.

Effects None

Exceptions None

MessageService

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

GetContentData

Table A–237 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 A–238 GetContentData Outputs

Type	Description
xsd:base64Binary	None

Effects None.

Exceptions None

GetMessageBoxes

Table A–239 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

Table A–240 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 A–241 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.

Table A–241 (Cont.) UpdateMessageBox Inputs

Input Name	Type	Description
mbox	tns:WSMessage Box	New or existing message box for the user, group, or workspace; if new, entity ID is not set.

Table A–242 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 A–243 DeleteMessageBoxes Inputs

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

Table A–244 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 A–245 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.
msgType	xsd:string	"Email", "FAX", "Voice", "SMS", "INSTANT" or "DISCUSSION". If null, all message types are considered.
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 A–246 GetEmailMessages Outputs

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

Effects None.

Exceptions None

GetAllMessageHeaders

Table A–247 GetAllMessageHeaders 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.
msgType	xsd:string	"Email", "FAX", "Voice", "SMS", "INSTANT" or "DISCUSSION". If null, all message types are considered.
msgFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

Table A–248 GetAllMessageHeaders Outputs

Type	Description
tns:WSMessageHeader[]	List of all message headers for the given type and user/group/workspace

Effects None.

Exceptions None

GetNewMessageHeaders

Table A–249 GetNewMessageHeaders 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.
msgType	xsd:string	"Email", "FAX", "Voice", "SMS", "INSTANT" or "DISCUSSION". If null, all message types are considered.
msgFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

Table A–250 GetNewMessageHeaders Outputs

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

Effects No effects

Exceptions No exceptions thrown

GetUnreadMessages

Table A–251 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.
msgType	xsd:string	"Email", "FAX", "Voice", "SMS", "INSTANT" or "DISCUSSION". If null, all message types are considered.
msgFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

Table A–252 GetUnreadMessages Outputs

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

Effects None.

Exceptions None

UpdateMessageHeaders

Table A–253 UpdateMessageHeaders Inputs

Input Name	Type	Description
msgHdrList	tns:WSMessage Header[]	List of message headers to be updated

Table A–254 UpdateMessageHeaders Outputs

Type	Description
tns:WSMessageHeader[]	List of updated message headers.

Effects None.

Exceptions No exceptions thrown

SendMessage

Table A–255 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 A–256 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 A–257 DeleteEmailMessages Inputs

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

Table A–258 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 A–259 SendInstantMessage Inputs

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

Table A–260 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 A–261 GetInstantMessage Inputs

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

Table A–262 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 A–263 SaveDraft Inputs

Input Name	Type	Description
uID	tns:WSEntity	None
msg	tns:WSMessage	Message to save.

Table A–264 SaveDraft Outputs

Type	Description
tns:WSResultStatus	None

Effects Sends the message from the logged in user.

Exceptions None

PreferenceService

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

GetPreferenceList

Table A–265 GetPreferenceList Inputs

Input Name	Type	Description
uID	tns:WSEntity	User identifier in the system; if it is null, the logged in user is assumed.

Table A–265 (Cont.) GetPreferenceList Inputs

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

Table A–266 GetPreferenceList Outputs

Type	Description
tns:WSPreferenceProfile[]	List of preference profiles for the user, with the active one as the first one.

Effects None.

Exceptions None

UpdatePreference

Table A–267 UpdatePreference Inputs

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

Table A–268 UpdatePreference Outputs

Type	Description
tns:WSPreferenceProfile	New or updated preference profile with the output entity ID set if new.

Effects If prefProfile is a new profile for the user (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 None

UpdatePreferenceProfile

Table A–269 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 A–270 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.

DeletePreferenceProfile

Table A–271 DeletePreferenceProfile Inputs

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

Table A–272 DeletePreferenceProfile 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

PresenceService

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

GetPresence

Table A–273 GetPresence Inputs

Input Name	Type	Description
uIDList	tns:WSEntity[]	List of valid user and/or group identifiers; if null, the logged in user is assumed.
presenceFilter	tns:WSFilter	Parameter that specifies a predicate, which involves filterable attributes to reduce the size of the returned list

Table A–274 GetPresence Outputs

Type	Description
tns:WSPresence[]	List of presence info for the users or groups, depending on the type of uID.

Effects None.

Exceptions None

UpdatePresence

Table A–275 UpdatePresence Inputs

Input Name	Type	Description
presence	tns:WSPresence	Presence information

Table A–276 UpdatePresence Outputs

Type	Description
tns:WSPresence	The updated presence information.

Effects Presence for the logged in user updated according to the presence parameter.

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

SubscribePresence

Table A–277 SubscribePresence Inputs

Input Name	Type	Description
uID	tns:WSEntity	Valid user or group identifier whose presence is to be subscribed. It cannot be null.

Table A–278 SubscribePresence Outputs

Type	Description
tns:WSPresence	Presence information for uID; it might contain pending status.

Effects A subscription to the presence information for the user or group is created for the logged in user.

Exceptions None

GetImBuddyList

Table A–279 GetImBuddyList Inputs

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

Table A–280 GetImBuddyList Outputs

Type	Description
tns:WSBuddyList	The buddy list for the specified user, subject to access control and privileges.

Effects None.

Exceptions None

WorkspaceService

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

GetWorkspaces

Table A–281 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
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.

Table A–282 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 A–283 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 A–284 GetWorkspaceTemplates Outputs

Type	Description
tns:WSWorkspaceTemplate[]	None

Effects None

Exceptions None

GetTrashItems

Table A–285 GetTrashItems Inputs

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

Table A–286 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

UnDeleteItems

Table A–287 UnDeleteItems Inputs

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

Table A–288 UnDeleteItems 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 A–289 PurgeTrash Inputs

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

Table A–290 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 A–291 DeleteWorkspaces Inputs

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

Table A–292 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 A–293 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 A–294 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

Types

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

WSActivityType

Enumeration, may have a value of CONFERENCE, HOLIDAY, MEAL, MEETING, ON_THE_PHONE, OTHER, OUT_OF_OFFICE, STEERING, TRAVEL, VACATION, or UNSET.

WSAddressBook

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

Table A–295 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 A–296 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.

Table A–296 (Cont.) WSArtifact Attributes

Attribute	Type	Description
artifactID	tns:WSEntity	Entity information for the artifact.
size	xsd:int	Size of the artifact
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 A–297 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 A–298 WSAttributeApplication Attributes

Attribute	Type	Description
value	xsd:string	
attributeTemplate	tns:WSAttribute	

WSAttributeName

Enumeration, may have a value of ALL_ATTRIBUTES, PARENT_ID, CATEGORY_ID_LIST, CATEGORY_APPLICATIONS, TAG_ID_LIST, LOCKS, VERSION_TYPE, WORKSPACE_PARENT_ID, WORKSPACE_DISCUSSION_ID_LIST, WORKSPACE_CALENDAR_ID_LIST, WORKSPACE_MEMBER_ID_LIST, WORKSPACE_INBOX_ID, WORKSPACE_ADDRESSBOOK_ID_LIST, WORKSPACE_TASK_ID_LIST, WORKSPACE_SCOPE_INFO, WORKSPACE_REMINDER_LIST, WORKSPACE_LIBRARY_ID_LIST, WORKSPACE_USERS, WORKSPACE_GROUPS, 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_STREAM", DOCUMENT_VERSION, DOCUMENT_VERSION_HISTORY, or DOCUMENT_CHECKEDOUT_BY

WSBond

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

Table A–299 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 A–300 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 A–301 WSCalendar Attributes

Attribute	Type	Description
timeZone	xsd:string	Time zone of the calendar.

Table A–301 (Cont.) WSCalendar Attributes

Attribute	Type	Description
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 A–302 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.
iCalPriority	xsd:int	The priority, if this is an iCalendar event.

Table A–302 (Cont.) WSCalendarEvent Attributes

Attribute	Type	Description
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

WSCalendarEventType

Either MEETING or DAY EVENT

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 A–303 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.

Table A–303 (Cont.) WSCalendarInvitation Attributes

Attribute	Type	Description
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

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 A–304 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:WSAttribute[]	
superCategory	tns:WSArtifact	Artifact information about the category to which this category belongs.
attributeDefinitionList	tns:WSAttribute[]	

Filterable Parameters

All attributes

WSCategoryApplication

Table A–305 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 A–306 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.
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 A–307 WSConference Attributes

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

WSConferenceLogEntry

Table A–308 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 A–309 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 A–310 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 A–311 WSConferenceTemplate Attributes

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

WSContact

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

Table A–312 WSContact Attributes

Attribute	Type	Description
workPhones	xsd:string[]	List of work phone numbers.
im_usernames	xsd:string[]	Instant Message usernames.
mailingAddress	xsd:string	Mailing address of this contact.

Table A–312 (Cont.) WSContact Attributes

Attribute	Type	Description
userID	tns:WSEntity	If this contact is an internal user, entity information for that user.
userType	xsd:string	INTERNAL or EXTERNAL
type	xsd:string	One of USER, GROUP, PERSON, or RESOURCE
homePhone	xsd:string	Home phone number for this contact.
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.
isBuddy	xsd:boolean	True if this contact is on the logged in user's buddy list.

Filterable Parameters

All attributes except userID, groupID, and isBuddy

WSContent

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

Table A–313 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	

Table A–313 (Cont.) WSContent Attributes

Attribute	Type	Description
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	

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 A–314 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 A–315 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 A–316 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.
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.

WSEnterprise

Top-level scope for all entities.

Table A–317 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 A–318 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

WSEventSeries

A set of recurring calendar events.

Table A–319 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.

Table A–319 (Cont.) WSEventSeries Attributes

Attribute	Type	Description
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 A–320 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 A–321 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.

Table A–321 (Cont.) WSFilter Attributes

Attribute	Type	Description
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 A–322 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 A–323 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 A–324 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 A–325 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 A–326 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

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 A–327 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 A–328 WSLocale Attributes

Attribute	Type	Description
language	xsd:string	Language of locale.
script	xsd:string	Script of locale.
region	xsd:string	Region of locale.
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 A–329 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 A–330 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 A–331 WSMessage Attributes

Attribute	Type	Description
messageHeader	tns:WSMessageHeader	

Table A–331 (Cont.) WSMessage Attributes

Attribute	Type	Description
receiverIDList	tns:WSParticipant[]	List of those to whom the message is sent.
inReplyToMessageID	xsd:string[]	If the message is a reply to an earlier message, this will refer to that earlier message.
replyToID	tns:WSEntity[]	ID that represents to whom to send a reply.
bccReceiverIDList	tns:WSParticipant[]	List of those to whom the message is BCC'd.
senderID	tns:WSParticipant	Who sent the message.
messageType	tns:WSMessageType	One of EMAIL, VOICE, SMS, INSTANT, DISCUSSION, or FAX.
ccReceiverIDList	tns:WSParticipant[]	List of those to whom the message is CC'ed.
msgInfo	tns:WSArtifact	Artifact information about this message.
flags	xsd:string[]	
subject	xsd:string	Subject of the message.
spawnedMessageIDList	tns:WSEntity[]	Deprecated. Unused.
body	tns:WSContent	Body of message.

Filterable Parameters

All attributes

WSMessageAnnotation

Table A–332 WSMessageAnnotation Attributes

Attribute	Type	Description
name	xsd:string	Name of message annotation.
attributeList	tns:WSProperty[]	Attributes associated with message annotation.

WSMessageBox

Table A–333 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.

Table A–333 (Cont.) WSMessageBox Attributes

Attribute	Type	Description
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

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 A–334 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.

WSNotification

Type of message that is used to alert the user upon occurrence of some event

Table A–335 WSNotification Attributes

Attribute	Type	Description
messageHeader	tns:WSMessageHeader	Message header of notification.
notificationInfo	tns:WSArtifact	Additional information about this notification.
body	tns:WSContent	Body of notification.
attachmentIDList	tns:WSEntity[]	List of attachments attached to notification.

Filterable Parameters

All attributes

WSOrganization

An organization is a scope under an enterprise. Organizations can be nested to form hierarchies.

Table A–336 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 A–337 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.

Table A–337 (Cont.) WSParticipant Attributes

Attribute	Type	Description
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 A–338 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

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 A–339 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

Table A–339 (Cont.) WSPredicate Attributes

Attribute	Type	Description
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 A–340 WSPreference Attributes

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 A–341 WSPreferenceProfile Attributes

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 A–342 WSPreferenceProperty Attributes

Attribute	Type	Description
value	xsd:string	
valueType	xsd:string	
values	xsd:string[]	
overWritable	xsd:boolean	
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 A–343 WSPresence Attributes

Attribute	Type	Description
userID	tns:WSEntity	ID of user.
presenceStatus	xsd:string[]	Presence status of user.
notes	xsd:string[]	Miscellaneous notes about user's presence.
presentityType	xsd:string	Type of entity whose presence is being encapsulated by this object. It could be of type User or Group
lastUpdated	xsd:dateTime	Date and time when the presence was last updated.
presenceImage	xsd:image	Image representing presence.

Filterable Parameters

All attributes

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 A–344 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 A–345 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 A–346 WSProperty Attributes

Attribute	Type	Description
value	xsd:string	Value of the property
valueType	tns:WSPropertyValueType	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

WSPropertyValueType

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 A–347 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 A–348 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

WSResource

An allocatable entity limited in capacity for performing an action or being acted upon (resources are usually not sharable at one time).

Table A–349 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 A–350 WSResultStatus Attributes

Attribute	Type	Description
errorMsg	xsd:string	Error message (if there is any)
statusCode	xsd:string	0 indicates success

WSRole

A role is a named set of permissions

Table A–351 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

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 A–352 WSRule Attributes

Attribute	Type	Description
conjunctive	xsd:boolean	
rule	xsd:string	
ruleDefinition	tns:WSEntity	
description	xsd:string	

Table A–352 (Cont.) WSRule Attributes

Attribute	Type	Description
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 A–353 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 A–354 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 A–355 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

Table A–355 (Cont.) WSSearchResultItem Attributes

Attribute	Type	Description
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 A–356 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 A–357 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 A–358 WSSubscription Attributes

Attribute	Type	Description
conjunctive	xsd:boolean	
attributeValues	tns:WSProperty[]	
subscriptionTemplate	tns:WSEntity	

Table A–358 (Cont.) WSSubscription Attributes

Attribute	Type	Description
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

WSSubscriptionTemplate

Template used to create subscriptions.

Table A–359 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 A–360 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 A–361 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 A–362 WSTaskAssignment Attributes

Attribute	Type	Description
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.
taskDueTime	xsd:dateTime	Due time of the task.
assigneeStartTime	xsd:dateTime	Start time of the task specified by the assignee.
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.

Table A–362 (Cont.) WSTaskAssignment Attributes

Attribute	Type	Description
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	xsd:string[]	
assigneeParticipantStatus	xsd:string	ACCEPTED, COMPLETED, DECLINED, NEEDS_ ACTION, TENTATIVE, WAITING_ON_OTHER
recurringEventSeries	tns:WSEventSeries	Event series associated with this task assignment.
assignee	tns:WSParticipant	Participant assigned to the task specified by this task assignment.
assigneeTimeSpent	xsd:dateTime	Amount of time spent by the assignee on the task.
source	xsd:string	ID of the TaskAssigment source.
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 A–363 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

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 A–364 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 A–365 WSUser Attributes

Attribute	Type	Description
userWorkspaces	tns:WSEntity[]	Workspaces to which the user belongs
userType	xsd:string	Immutable. Type of user, such as "internal" or "external".
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
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

WSVersion

Defines attributes of a versionable artifact, which represents a specific version of an artifact.

Table A–366 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.

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 A–367 WSWorkspace Attributes

Attribute	Type	Description
parentID	tns:WSEntity	Identifier of the parent community
discussionForumIDList	tns:WSEntity[]	List of discussion forums associated with workspace.
calendarIDList	tns:WSEntity[]	List of calendars associated with workspace.
inboxID	tns:WSEntity	Workspace inbox.
workspaceType	xsd:string	Type of the workspace, either PERSONAL or TEAM
creationDate	xsd:dateTime	Date and time workspace was created.
addressbookIDList	tns:WSEntity[]	List of address books associated with workspace.
categoryList	tns:WSEntity[]	List of categories associated with workspace.
tasklistList	tns:WSEntity[]	List of tasklists associated with workspace.
memberIDList	tns:WSEntity[]	List of members in the workspace (updated through the membership service)
creator	tns:WSEntity	Creator of workspace.
tagIDList	tns:WSEntity[]	List of tags associated with workspace.
versioningType	xsd:string	Versioning type.
scopeInfo	tns:WSScope	Scope information of workspace.
isPublished	xsd:boolean	True if workspace has been published.

Table A–367 (Cont.) WSWorkspace Attributes

Attribute	Type	Description
lastModifiedDate	xsd:dateTime	Last date and time workspace was modified.
emailAddress	xsd:string	E-mail address associated with workspace.
reminderList	tns:WSEntity[]	List of reminders associated with workspace.
locks	tns:WSLock[]	List of reminders associated with workspace.
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.
libraryIDList	tns:WSEntity[]	Folder for artifacts

Filterable Parameters

All attributes

WSWorkspaceTemplate

Defines a workspace

Table A–368 WSWorkspaceTemplate Attributes

Attribute	Type	Description
workspaceTemplateType	xsd:string	
contactInfo	xsd:string	
workspaceTemplateInfo	tns:WSArtifact	
authorCreationTime	xsd:dateTime	
comyrightInfo	xsd:string	
templateID	xsd:string	
author	xsd:string	
transportableFormat	xsd:base64Binary	