ORACLE CRM On Demand

Oracle Web Services On Demand Guide

Version 7.0 (Oracle CRM On Demand Release 19)
Rev. B

December 2011

Contents

1 What’s New in This Release

What’s New in Oracle Web Services On Demand Guide, Version 7.0 (Oracle CRM On Demand Release 19) Rev. B

Table 1 lists changes described in this version of the documentation to support Version 7.0 (Oracle CRM On Demand Release 19) Rev. B of the software.

Table 1. What’s New in Oracle Web Services On Demand Guide, Version 7.0 (Oracle CRM On Demand Release 19) Rev. B

Topic	Description
“Web Services v2.0 API Calls” on page 102 “Service API Calls” on page 139	New topics. For the Web Services v2.0 and Service API methods, sample SOAP requests and SOAP responses are provided.
Various topics including: “SetPassword” on page 185	Modified topics. Various enhancements have been made to improve documentation quality and clarity.

Table 2 lists changes described in this version of the documentation to support Version 7.0 (Oracle CRM On Demand Release 19) Rev. A of the software.

Table 2. What’s New in Oracle Web Services On Demand Guide, Version 7.0 (Oracle CRM On Demand Release 19) Rev. A

Topic	Description
“Using the pagesize, startrownum, and recordcountneeded Arguments” on page 116	Modified topic. A description has been added of the expected behavior when a Web Services v2.0 QueryPage request is made with recordcountneeded set to `true` and the search specification contains calculated fields.
“GetMapping” on page 161	Modified topic. For the GetMapping method, information has been added about fields for which a display name with a value of hidden is returned.

Table 2. What's New in Oracle Web Services On Demand Guide, Version 7.0 (Oracle CRM On Demand Release 19) Rev. A

Topic	Description
"SetSessionTimeZone" on page 187	Modified topic. The information about the CurrentServerTime output argument of the SetSessionTimeZone method has been clarified.
"Core Parent Objects (Web Services v1.0)" on page 254 "Life Sciences Edition Parent Objects (Web Services v1.0)" on page 368 "Financial Services Edition Parent Objects (Web Services v1.0)" on page 371 "Automotive Edition Parent Objects (Web Services v1.0)" on page 381	Modified topics. For objects accessible through the Web Services v1.0 API, the information has been updated to show for which objects the IntegrationId and ExternalSystemId fields can be used as a user key.

Table 3 lists changes described in this version of the documentation to support Version 7.0 (Oracle CRM On Demand Release 19) of the software.

Table 3. What’s New in Oracle Web Services On Demand Guide, Version 7.0 (Oracle CRM On Demand Release 19)

Topic	Description
“Mapping Primary Address Fields Using Web Services” on page 35	New topic. In Release 19, a new field is accessible through Web services allowing you to dynamically map the primary address field in your application to the primary address field used by Oracle CRM On Demand, based on the specified country value. This field is specific to Web services and is not accessible through the Oracle CRM On Demand user interface.
“Web Service Client Name Identification” on page 41	New topic. To allow company administrators to determine the origin of Web service transactions being performed for their company, a ClientName value can now be passed with each SOAP request for both stateless and stateful Web service operations. This feature applies to both the Web Services v1.0 and v2.0 APIs.
“About Service Allotments” on page 45 “Best Practices for Adhering to Web Service Allotments” on page 76	New topics. A number of service allotments have been introduced: ■ **Web Services Operations Allotment**. Determines the number of Web services operations that a company can perform within a 24-hour period. ■ **Web Services Concurrent Request Allotment**. Determines the maximum number of concurrent Web service requests that a company can process.
“Oracle CRM On Demand Authentication Mechanisms” on page 54	New topic. Two new mechanisms have been added for authentication of stateless Web service requests with Oracle CRM On Demand. You now have the option of providing a SAML v1.0 Assertion or a Single Sign-On (SSO) token value in the SOAP header to be used to authenticate a Web service request. The addition of these new login mechanisms allows developers to use stateless Web service requests without having to provide a user name and password in the request for authentication.

Table 3. What's New in Oracle Web Services On Demand Guide, Version 7.0 (Oracle CRM On Demand Release 19)

Topic	Description
"Querying Lists" on page 124	New topic. For the QueryPage method of the Web Services v2.0 API, there is now an optional NamedSearchSpec argument that is used to query any named list in Oracle CRM On Demand. Such a query returns the set of records that match that list's filter criteria, without having to duplicate the filter criteria within the Web services client application.
"AuthAuthorizationUser" on page 140	New topic. The method AuthAuthorizationUser has been added to the Services API. This method validates a user's credentials and returns the list of privileges for that user.
"GetLists" on page 158	New topic. The method GetLists has been added to the Services API. This method returns the set of lists that a user has access to for a specified object.
"Administrative Services API Calls" on page 201 "CurrencyRead" on page 211 "IndustryRead" on page 228 "CurrentOrganizationRead" on page 213 "PageLayoutRelatedInformationRead" on page 232 "RoleRead" on page 238 "SalesAssessmentTemplateRead" on page 241	New topics. The Administrative Services APIs, used to automate the administration of company configurations, have been extended. For a number of Web services, methods have been added as shown in this list: ■ Assessment Scripts - the SalesAssessmentTemplateRead, and SalesAssessmentTemplateReadAll methods ■ Company Profile - CurrentOrganizationRead method ■ Currency Picklist - CurrencyRead, CurrencyReadAll, and CurrencyUpsert methods ■ Industry - IndustryRead and IndustryReadAll methods ■ Page Layout Related Information - PageLayoutRelatedInformationRead methods ■ Role Management - RoleRead method

Table 3. What's New in Oracle Web Services On Demand Guide, Version 7.0 (Oracle CRM On Demand Release 19)

Topic	Description
For example: "PageLayoutFieldRead" on page 229 "PageLayoutFieldReadAll" on page 230	Modified topics. For a number of the Administrative Services API methods, only customized configuration information was returned for previous releases. The <IncludeAll> element can now be used to specify that out-of the-box as well as customized configuration data is returned. You can use <IncludeAll> on Read and ReadAll methods for the following services: ■ Custom Record Type ■ Field Management ■ Page Layout Field ■ Page Layout Related Information ■ Picklist
"PageLayoutFieldRead" on page 229 "PageLayoutRelatedInformationRead" on page 232	Modified topics. For the PageLayoutRead, PageLayoutReadAll, PageLayoutRelatedInformationRead, and PageLayoutRelatedInformationReadAll methods, you can now use the <CurrentUserOnly> argument to specify that configuration data for the current user only is returned.
"Access to the Administrative Services APIs for Non-administrative Users" on page 204	New topic. A number of the Administrative Services APIs now allow read-only access for non-administrative users.
"CustomWebTabUpsert" on page 221	Modified topic. The methods for the Custom Web Tab Web service have been updated to support the new fields added to the Oracle CRM On Demand UI for custom Web tabs.
"BulkOpImportCreateRequest" on page 244	Modified topic. Information about the CSVColumnHeaders and EnableImportAudit arguments of the BulkOpImportCreateRequest method has been added. The EnableImportAudit argument option allows you to enable or disable the auditing of record creation and updating during import. Suppressing auditing while importing can improve the import performance for some objects.

Table 3. What's New in Oracle Web Services On Demand Guide, Version 7.0 (Oracle CRM On Demand Release 19)

Topic	Description
"Exchange Rate" on page 417	New topic. A new Web service, Exchange Rate, has been added to allow administrators to create new currency exchange rates, update, or delete existing exchange rates, and query the exchange rate history for their company. This feature makes it easier to keep exchange rate data synchronized between Oracle CRM On Demand and other applications and data sources.
"Allotment Usage" on page 403	New topic. A new Web service, Allotment Usage, has been added to allow querying of the service allotment usage data for a company.
"Service Allotment" on page 427	New topic. A new Web service, Service Allotment, has been added to allow querying of the service allotment data for a company.

Table 3. What's New in Oracle Web Services On Demand Guide, Version 7.0 (Oracle CRM On Demand Release 19)

Topic	Description
"Account Contact" on page 545 "Account Team" on page 548 "Activity Assessment" on page 552 "Activity Assessment Value" on page 552 "Competitor" on page 557 "Contact Role" on page 560 "Custom Object 1 through 3" on page 563 "Opportunity" on page 571 "Opportunity Child" on page 572 "Opportunity Team" on page 572	New topics. For Web Services v2.0 only, a number of new child objects are now accessible: ■ Account: ■ Competitor ■ Contact Role ■ Contact ■ Custom Object 01 through 03 ■ Partner ■ Team ■ Activity: ■ Assessment ■ Assessment Values ■ Custom Object 01 through 03 ■ Contact: ■ Opportunity ■ Custom Object 01 through 03 ■ Lead: ■ Custom Object 01 through 03 ■ Opportunity: ■ Competitor ■ Contact ■ Contact Role ■ Custom Object 01 through 03 ■ Opportunity Child ■ Partner ■ Team
Appendix B, "Mapping of Display Names to Service API Object Names"	Modified Appendix. Details of some new objects have been added to the table showing the mapping of display names to the Service API object names.

2 Overview of Web Services On Demand

This chapter provides an overview of Oracle CRM On Demand's support for Web services. It contains the following topics:

About Web Services

The term *Web services* describes a standardized way of integrating Web-based applications over the Web. Web services allow businesses to communicate with each other and with other clients, without intimate knowledge of each other's IT systems. Web services share business logic, data, and processes through a Web services application programming interface (API). Application developers can then add the Web services to a software application (such as a Web page or executable program) to offer specific functionality to users.

Web Services Core Technologies

The Web services core technologies are a set of standards-based technologies that include:

- **Extensible Markup Language (XML).** The standard markup language that allows the definition of message structures and facilitates the passing of data between software applications.
- **Web Services Description Language (WSDL).** The XML-formatted language that is used to describe a Web service. A WSDL file defines the available methods, message structures, and network addresses required for using a specific Web service.
- **Simple Object Access Protocol (SOAP).** The XML-based protocol that is used to send Web services request and response messages. Web services messages are sent between the customer implementation of Web services and the SOAP handler on the Oracle Web Server.

For more information on Web services technologies, see:

http://www.w3.org/2002/ws.

Oracle CRM On Demand Web Services Toolkit

The Web Services Toolkit provides access to an application programming interface (API) that companies can use to build programs to integrate with Oracle CRM On Demand. The Toolkit includes a set of WSDL files that describes the interface to the Oracle CRM On Demand objects. This provides a programmatic interface for accessing your company's Oracle CRM On Demand information. A customer application can use the WSDL files through standard Web services development tools, such as those provided by the Oracle SOA Suite.

The API for this release of Oracle CRM On Demand is backward-compatible with previous releases.

Figure 1 shows how the Web Services Toolkit interacts with the Oracle CRM On Demand database. The customer uses the Web Services Toolkit (WSDL files) to define the objects and methods that are contained in the Oracle CRM On Demand Hosted Service. The customer application communicates with Oracle CRM On Demand over the Internet using the secure HTTPS protocol. It invokes the Web services implementation contained in the Oracle CRM On Demand Hosted Service.

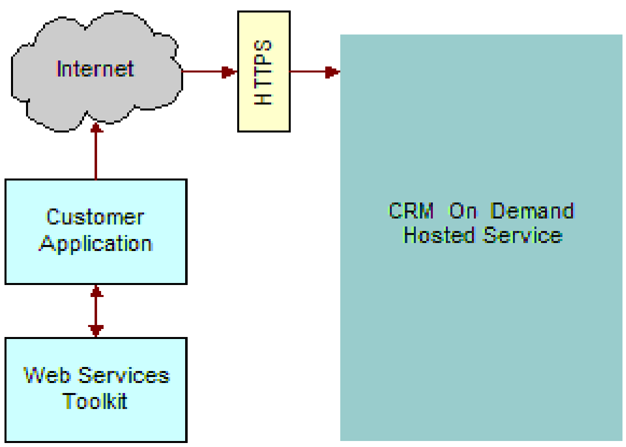

Figure 1. How Web Services Communicate with Oracle CRM On Demand

Oracle CRM On Demand is designed to be backward-compatible with previous releases. WSDL files from previous releases will continue to work with newer releases of Oracle CRM On Demand, and there is no need for customers to modify their code when upgrading to a new release of Oracle CRM On Demand.

Oracle CRM On Demand Web Services and Integration with Oracle CRM On Demand

The Web Services On Demand API allows companies to build programs to integrate with Oracle CRM On Demand. Some common examples of client integrations include the following:

- **Integrations of CRM and back-office applications.** You can retrieve real-time sales, marketing, and service information from Oracle CRM On Demand and use it in financial and other back-office applications. For example, you can retrieve information about recently closed opportunities through the Web services interface and insert this information into an order entry system that has a Web services user interface. In addition, you can store information from back-office applications in Oracle CRM On Demand for instant access by users, visible in custom fields on any Oracle CRM On Demand page.
- **Web-based portal applications.** You can create customized Web-based applications using Active Server Pages (ASPs), Java Server Pages (JSPs), or similar Web technology that accesses Oracle CRM On Demand through the Web services interface. For example, an Oracle CRM On Demand customer can deploy a customized Web form on its corporate Web site, allowing visitors to enter requests for more information. The application creates new lead records in Oracle CRM On Demand for these requests through the Web services interface. Another Web page can allow visitors to browse through solutions to common problems stored in Oracle CRM On Demand and retrieved in real time through the Web services interface.
- **Custom add-on modules.** Customers can also extend Oracle CRM On Demand functionality. For example, a company can create a custom add-on module to streamline its unique quote creation process, or a company can create additional utilities to perform mass data cleanup operations. These modules access data in Oracle CRM On Demand directly through the Web services interface. Oracle CRM On Demand administrators and users can run these modules while concurrently accessing the Oracle CRM On Demand user interface.

Web Services Security

The Oracle CRM On Demand Web Services Integration framework includes the following security features:

- The mustUnderstand attribute of Simple Object Access Protocol (SOAP) 1.1 is supported. This allows a client to specify that the target server must be capable of processing all parameters in the SOAP request header, otherwise the requests must be rejected.
- SOAP message validation is performed, for example, to check for badly formed SOAP requests or for SOAP header elements that are not namespace-qualified.
- Support is provided for the WS-I Basic Security Profile Version 1.0. For more information, see "Support for the WS-I Basic Security Profile Version 1.0" on page 24.
- All communications are encrypted with Secure Sockets Layer (SSL) for security (minimum 128-bit).
- Access is session-based, requiring authorization with a valid Oracle CRM On Demand user name and password.
- Inactive sessions are reused or closed automatically after a period of inactivity.
- The same data visibility and access capabilities that apply to users in the Oracle CRM On Demand hosted service are applied to users connected through the Web services interface. Data visibility and access are restricted by the role that your company assigns. Permissions are checked for every data access.

- A full audit trail of Web services activity is available through Oracle CRM On Demand's Administration pages. These pages display both current and historical usage statistics.
- A number of other proprietary solutions protect Oracle CRM On Demand against malicious use of the Web services interface. These solutions are constantly reviewed and improved as new technologies and techniques become available.

A session with a standard HTTPS request is created to establish a connection with Oracle CRM On Demand through the Web services interface. A client can create a new session with the login operation and close it with the logoff operation. When a session is created, an encrypted session identifier is provided to the client. which for stateful Web services requests, must be included in all subsequent requests during that session. For more information, see "About Establishing and Managing the Web Services Session" on page 53.

Support for the WS-I Basic Security Profile Version 1.0

Support is provided for the WS-I Basic Security Profile Version 1.0, which describes the set of parameters used to authenticate a Web services transaction.

Oracle CRM On Demand has implemented support for the Username and PasswordType parameters, which are part of the UserNameToken standards. This allows a username and password to be passed with a SOAP request, which removes the necessity for a separate login operation. For more information, see "Using Stateless Web Service Requests" on page 54.

Passwords can be specified as type PasswordText only, which mean that the password is in clear text format.

WSSE Namespace Support

The SOAP header of messages received by Oracle CRM On Demand are validated to ensure they are namespace-qualified. Oracle CRM On Demand supports the following namespace values when specifying the WSSE namespace in a SOAP request:

- **Draft Namespaces**:
  - wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
  - wsse="http://schemas.xmlsoap.org/ws/2002/07/secext"
- **Version 1.0 Namespace**: wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"

  The WSSE Version 1.0 namespace must be specified to perform a stateless transaction. (In addition, the Web Services R16 Compatibility Mode check box must be cleared in the Company Profile page and the Username and PasswordText tokens must be provided in the request.)

For more information about stateless transactions and the use of the WSSE namespace, see Chapter 4, "Establishing and Managing the Web Services Session."

Web Services Reliability

All server components of Oracle CRM On Demand, including those responsible for the Web services interface, incorporate load balancing and other high-availability mechanisms. These mechanisms prevent the service from being interrupted by server or network infrastructure failure.

Web Services and the Oracle CRM On Demand Objects

Oracle CRM On Demand Web services allow applications to integrate with Oracle CRM On Demand. They provide the ability to find and invoke the core Oracle On Demand Web Services across the Web from any client application language. This ability makes the process of using Oracle CRM On Demand Web Services easy for those who want to use them.

The Oracle CRM On Demand services provide a basis for customers to perform integration with Oracle CRM On Demand based on SOAP technology.

All major Oracle CRM On Demand business objects are accessible in the Web services, with the names of the Web services matching the default names of the business objects. Chapter 7, "Oracle CRM On Demand Objects Accessible Through Web Services" details the Oracle CRM On Demand parent and child objects that are accessible through Oracle CRM On Demand Web Services.

Web Service APIs

Starting with Web Services On Demand Version 4.0 (CRM On Demand Release 16) , objects are accessible through two APIs:

- **Web Services v1.0.** Used to interact with Custom Objects 01-03, as well as preconfigured objects.
- **Web Services v2.0.** Used to interact with all Oracle CRM On Demand Custom Objects, as well as preconfigured objects.

Before Web Services On Demand Version 4.0, only the Web Services v1.0 was available. In addition, the following APIs are provided:

- **Service APIs**. Used to perform management tasks and retrieve integration events through Web services.
- **Administrative Services APIs**. Used to access company metadata through Web services.

For the Web Services v1.0 API, operations work on the parent objects and all child components are synchronized with the parent. The Web Services v2.0 API, however, works on a node basis, where parent and child components are treated as separate nodes.

The Web Services v2.0 API provides an Execute method for performing multiple operations on separate nodes, and the Web Services v2.0 QueryPage method offers additional options (through the searchspec, namedsearchspec, sortorder, and sortsequence arguments) for issuing queries compared to the Web Services v1.0 QueryPage method.

Table 4 shows the methods available through the Web Services v1.0 and Web Services v2.0 APIs for access to objects.

Table 4. Web Services v1.0 and Web Services v2.0 Methods

Web Services v1.0	Web Services v2.0	Comments
Delete	Delete	Finds records in the Oracle CRM On Demand database that match specified field values, and then deletes them. Deleted records are visible in the Deleted Items area of the Oracle CRM On Demand UI and can be queried using the DeletedItemQueryPage method.
DeleteChild	Not applicable	Deletes child records from the Oracle CRM On Demand database, or removes the association between the child and the parent object.
Not applicable	Execute	Executes multiple update, insert, and delete operations on separate records in the Oracle CRM On Demand database within the same Web services request.
Insert	Insert	Inserts new records into the Oracle CRM On Demand database.
InsertChild	Not applicable	Inserts new child records into the Oracle CRM On Demand database.
InsertOrUpdate	Not applicable	Updates existing records or inserts a new record if one did not exist.
QueryPage	QueryPage	Executes a query against a specified list of records, and returns a subset of the records that match the search criteria set by the method arguments.
Update	Update	Updates records with a new value.
UpdateChild	Not applicable	Updates child records with a new value.

Table 5 shows differences between Web Services v1.0 and Web Services v2.0.

Table 5. Web Services v1.0 and Web Services v2.0 Differences

Web Services v1.0	Web Services v2.0
Supports an upsert operation through InsertOrUpdate call	Does not support an upsert operation
Pagination parameters are supported only at the parent level	Pagination parameters are supported at both the parent and child level

Table 5. Web Services v1.0 and Web Services v2.0 Differences

Web Services v1.0	Web Services v2.0
Returns all child records even if the condition is true for one child. For example, the QueryPage call returns all partner children from an account even if the condition is true for only one partner child	Outputs only the specific child whose condition was met. For example, QueryPage returns only the specific partner child from the account for which the condition was true
UseChildAnd argument of QueryPage call is available for using OR/AND logic between parent and child	The UseChildAnd argument is not available. Instead, by default, all parent records matching the parent criteria and only children matching the child criteria are returned.
Operators cannot be used to construct complex queries across multiple fields	The SearchSpec argument of QueryPage can be used to construct complex queries across multiple fields in a request. For example, the OR operator can be used to find all records that match the specified condition for [Field A] OR the specified condition for [Field B].
Sort order is not customizable	Sortorder and sortsequence arguments are available to customize the sorting order of the records
Update call removes child objects not specified in the request	An Execute call with “operation=update” at the parent level removes the unspecified children in the request
InsertChild call is used to insert the children for existing parent objects	For Web Services v2.0: ■ Insert call can be used to insert both parent records and child records. ■ If a child node is specified in the request, the Insert call inserts the child and associates it with the existing parent record. ■ If a child node is missing, the Insert call inserts only the new parent record.
UpdateChild call is used to update child records	For Web Services v2.0: ■ Update call can be used to update parent records and child records ■ If a child node is specified in the request, the Update call updates the child in the existing parent record ■ If the child node is missing, the Insert call updates only the existing parent record

Table 5. Web Services v1.0 and Web Services v2.0 Differences

Web Services v1.0	Web Services v2.0
DeleteChild call is used to delete the child records	For Web Services v2.0: ■ Delete call can be used to delete both parent records and child records ■ If the child node specified in the request is available, the Delete call deletes the child in the existing parent record, and leaves the parent record undeleted. ■ If the child node is missing, the Delete call deletes the existing parent record.
InsertChild, UpdateChild, and DeleteChild methods are used to perform operations on child records	In an Execute request, a specific node within the request can be skipped using the "operation=skipnode" attribute. This can be used to simulate InsertChild, UpdateChild or DeleteChild by skipping the parent node and only performing the specified actions on the child records.
LOVLanguageMode argument is not available	The LOVLanguageMode argument is an input argument for all of the Web Services v2.0 calls. It determines whether the processing for picklist fields occurs using language independent codes (LIC) or language dependent codes (LDC).
ViewMode argument is not available	The ViewMode argument, which specifies the level of access to records specified in the method call, is available for all of the Web Services v2.0 calls

There are some differences between the format of the WSDL files for Web Services v1.0 and Web Services v2.0:

- In the Web Services v2.0 API, strong data typing is supported. Therefore, in the Web Services v2.0 WSDL files, fields are represented by a range of xsd: data types, while in Web Services v1.0 WSDL files, all fields have the xsd:string data type. For more information, see "Field Types Supported by Oracle CRM On Demand" on page 30.
- In Web Services v2.0, messages do not include the business service name, and have the format:

  [*Objectname*][*Method*]_[Input/Output]

  For example:

  `AccountInsert_Input, ContactQueryPage_Output`

  as opposed to the following for Web Services v1.0:

  `AccountWS_AccountInsert_Input, ContactWS_ContactQueryPage_Output`

- The target namespace of the WSDL for Web Services v2.0 is:

`urn:crmondemand/ws/ecbs/`*`objectname/`*

compared to the following for Web Services v1.0:

`urn:crmondemand/ws/`*`objectname/`*

About Parent-Child Relationships

Many of the Oracle CRM On Demand objects interact with each other through parent-child relationships. A parent object refers to the main or base object of interest and the child object refers to objects that are related to the parent in some way—for example, if the child is contained in the parent, or if the child has records that refer to the parent.

These parent-child relationships can be one-to-many or many-to-many. For example, a lead can be associated with a particular account, but an account can have many leads associated with it. In this case, you can think of the relationship between the account and its leads as a one-to-many parent-child relationship.

Other relationships can be many-to-many, meaning that many children are associated with many parents. For example, a contact can be associated with several opportunities, or an opportunity can have several contacts associated with it. In this case, you can think of the relationship between contacts and their opportunities as a many-to-many parent-child relationship. The parent-child relationship between contacts and opportunities can be treated with either the opportunity as the parent with contacts as children, or with the contact as the parent and the opportunities as children.

Web Services On Demand and Custom Fields

Oracle CRM On Demand allows company administrators to create custom fields that capture information specific to the company's needs. Web Services On Demand allows customers to interact with the data stored in these custom fields. Each custom field has an associated integration tag that is used by Web services and Web links to reference data in custom fields. This feature allows administrators to change the display name of a field without making modifications to the existing Web services integration.

Fields are labeled as iField_Name in the Custom WSDL files and Custom FieldType# in the Generic WSDL files.

Custom Fields can be referenced using two different integration tags:

1. The Custom WSDL file uses the format:

   `fieldtypeDisplay_Name`

   For example, a custom Boolean field with the display name `Account Selected` would have the default custom integration tag `bAccount_Selected`.

2. The Generic WSDL file uses the format:

   `fieldtype##`

   For example, a custom Boolean field would have the generic integration tag `CustomBoolean0`.

To view or modify integration tag information for a record type

1. Navigate to the Field Setup Administration page for the required record type.

   For example: Admin, Application Customization, Account, Account Field Setup, Rename Fields.

2. Click Advanced.

   The integration tag information is displayed for you to view or modify.

You can download custom WSDL files in which the XML tags for the custom fields are based on the integration tags.

To download a WSDL file that is specific to your company's customization

1. Navigate to the Web Services Administration page.
2. Select Web Services v1.0 or Web Services v2.0, and click Go.
3. Select the required record type, and click Download Custom WSDL.

   A record type's WSDL that is specific to your company's customization is downloaded.

For more information about downloading WSDL files, see the online help for Oracle CRM On Demand.

Field Types Supported by Oracle CRM On Demand

The field types supported depend on whether the Web Services v1.0 or Web Services v2.0 API is used, as described in the following topics.

Web Services v1.0

For the Web Services v1.0 API, all fields in Web services On Demand are transmitted and received as strings. It is the client's responsibility to cast these to and from the required data type in any application. The proper type can usually be determined from the name, purpose, or application of the field. There is no dynamic method for determining field types. You can derive clues about a field's type from its name as follows:

- A name ending in the suffix Id is usually a key field, such as a primary key, foreign key, or user key Id. It can usually be treated as a unique text string.
- Fields with names containing Date or Time, such as LastUpdated, DueDate, StartTime, or EndTime might be date fields.
- Telephone number fields can be treated as numeric phone numbers or as plain text. When performing queries on phone number type fields the following formats must be used in Query operations:
  - U.S. Format: +1 872 9269923
  - France: +33 01 40359564

  - Japan: +81 3 54579623
- Other numeric fields, such as currency, size, revenue, or probability can be treated as integer, floating point, or text fields depending on the client application.
- Boolean fields have the value Y for true or N for false.
- Most other fields can be treated as ordinary text.

NOTE: If you attempt to query a field of type Date with syntax like <CloseDate>>'01/01/2004 00:00:00'</CloseDate> you get an error, because the time parameter 00:00:00 is only valid for fields of type Date/Time and not for fields of type Date.

Web Services v2.0

The Web Services v2.0 API supports strong data types for fields, so fields are represented by appropriate XSD data types. Table 6 shows the list of supported XSD data types.

Table 6. Data Type Mapping in the Web Services v2.0 API

Data Type	Mapped XSD Data Type
BOOL	xsd:boolean
CURRENCY	xsd:decimal
NUMBER	xsd:decimal
DATE	xsd:date
DATETIME	xsd:dateTime
UTCDATETIME	xsd:dateTime
ID	xsd:string
NOTE	xsd:string
PHONE	xsd:string
TEXT	xsd:string
INTEGER	xsd:int
TIME	xsd:time
Others	xsd:string

If an incorrect data type is provided in a Web services request, the field is updated to NULL or a default value for that specific data type, as shown in Table 7.

Table 7. Updating of Fields When Incorrect Data Types are Provided in the Web Services v2.0 API

XSD Data Type	Default Value/Null
xsd:boolean	N
xsd:decimal	NULL
xsd:date	NULL
xsd:dateTime	NULL
xsd:string	NULL
xsd:int	0
xsd:time	NULL

For example, Activity has a field named Cost, which takes integer values. If you provide a text value for the field in an update request, the previous value is replaced with a 0.

You can find further details about the definition of XSD data types here:

http://www.w3.org/TR/xmlschema-2/#built-in-datatypes

Special Search Fields

Some field names are prefixed with CI_ to denote that they are special fields that provide better search functionality. These fields do not exist for all objects but are easily identified in the WSDL files as shown in the following excerpt from the Account WSDL file:

```
<xsd:element name="CI_AccountName" maxOccurs="1" minOccurs="0" type="xsd:string"></
xsd:element>

<xsd:element name="CI_Location" maxOccurs="1" minOccurs="0" type="xsd:string"></
xsd:element>
```

Support for Multi-Select Picklists

A multi-select picklist is a picklist from which the user can select multiple values. In Web Services On Demand, multi-select picklists are only accessible for the following record types:

- Account
- Activity
- Contact
- Custom Object 01

- Custom Object 02
- Custom Object 03
- Lead
- Opportunity
- Service Request

For these record types, all standard and custom multi-select picklist fields are accessible. You can add, remove, replace or query selections in parent-level multi-select picklist fields, however child-level multi-select picklist fields are not supported.

Input and output values are language-independent code (LIC) delimited, but the multi-select picklist delimiter is always a semicolon regardless of locale for input and output: <LIC1>;<LIC2>.

Locale-Dependent Access to Oracle CRM On Demand

Oracle CRM On Demand Web Services does not provide any specialized localization interfaces. Oracle CRM On Demand supports full localization, so that the data created through Web services is localized for users. The localized fields in the Web services interfaces follow the formats outlined in the following topics.

Date and Time Fields

Date and time fields for Web services v1.0 are in the following format:

```
MM/DD/YYYY hh:mm:ss
```

For Web services v2.0, the data in SOAP requests conforms to XSD data formats.

The XSD dateTime datatype has the format:

```
yyyy '-' mm '-' dd 'T' hh ':' mm ':' ss ('.' s+)? (zzzzzz)?
```

For example:

```
2009-10-10T12:00:00-05:00
```

represents noon on 10th October 2002, Central Daylight Savings Time, which is equivalent to Eastern Standard Time in the US.

As a further example:

```
2009-10-09T17:00:00Z
```

represents noon on 9th October 2009, UCT, which is equivalent to the GMT time zone.

For the QueryPage method of Web Services v2.0, either the XSD formats (recommended) or the locale-specific formats can be used.

Number and Currency Fields

Number and currency fields in Oracle CRM On Demand are in raw number format. In other words, number and currency fields hold only digits with no currency symbols, decimal separators, or other numeric separators.

NOTE: The “decimal point” may be represented by a different symbol depending on the user's locale.

Validation of Email Fields

When Oracle CRM On Demand validates fields containing email addresses, it identifies the following as invalid:

- Empty string
- String too long
- No characters before the at sign (@) character, for example: @riqhtequip.com
- No at sign (@) character, for example:isampleriqhtequip.com
- No period (.) character, for example: isample@riqhtequipcom
- No domain, for example: isample@
- No domain suffix such as com, for example: isample@riqhtequip
- Multiple at signs (@), for example: isample@@riqhtequip.com
- Consecutive period (.) characters, for example: isample@riqhtequip..com
- Spaces in the string, for example: isa mple@riqhtequip
- Characters other than the following in the local part of an email address:
  - Uppercase and lowercase letters (case insensitive)
  - The digits 0 through 9
  - The characters:
    - Exclamation point (!)
    - Hash symbol (#)
    - Dollar sign ($)
    - Percent (%)
    - Ampersand (&)
    - Single quotation sign (')
    - Asterisk (*)
    - Plus sign (+)
    - Minus sign (-)
    - Slash (/)

- Equal sign (=)
- Question mark (?)
- Caret (^)
- Underscore (_)
- Back single quotation mark (`)
- Left curly brace ({)
- Vertical bar (|)
- Right curly brace (})
- Tilde (~)

- Any special characters in the domain name of an email address. These special characters are the same as those allowed in the local part of the email address, and also the left and right parentheses ().

Mapping Primary Address Fields Using Web Services

In Web services requests, a *PrimaryAddressLine1* field is used to dynamically map the primary address field from an external application to the primary address field in Oracle CRM On Demand. The primary address field in Oracle CRM On Demand can vary depending on the Country value for each address; thus the *PrimaryAddressLine1* field will map to a different field in the address object based on the Country value. Table 8 and Table 9 shows the mapping for the *PrimaryAddressLine1* field depending on the selected Country value.

NOTE: The actual name of the *PrimaryAddressLine1* fields varies with the record type as shown in "Objects Supporting the PrimaryAddressLine1 Field" on page 39.

Table 8. *PrimaryAddressLine1* To Address Field Mapping by Country

Address	Lead	Account (Billing Address)	Account (Shipping Address)	Contact (Billing Address)	Contact (Shipping Address)	Country
Street Address	Street Address	Ship To Street Address	Ship To Street Address	Personal Street Address	Primary Street Address	Group A See Table 9
Street Address 3	Street Address 3	Ship To Street Address 3	Ship To Street Address 3	Personal Street Address 3	Primary Street Address 3	Nauru
Postal Code	Postal Code	Bill To Postal Code	Ship To Postal Code	Personal Postal Code	Primary Postal Code	Group B See Table 9

Table 8. *PrimaryAddressLine1* To Address Field Mapping by Country

Address	Lead	Account (Billing Address)	Account (Shipping Address)	Contact (Billing Address)	Contact (Shipping Address)	Country
County	County	Bill To County	Ship To County	Personal County	Personal County	Group C See Table 9
Province	Province	Bill To Province	Ship To Province	Personal Province	Primary Province	Qatar
City	City	Bill To City	Ship To City	Personal City	Primary City	Papua New Guinea

Table 9. Groups of Countries with Different Address Field Mappings

Group	Countries
A	United States and all other countries apart from those in groups B and C, and those mentioned in Table 8.
B	Hungary, Belarus, Burkina Faso, Congo, Kazakhstan, Kyrgyzstan, Russian Federation, Congo Sudan, Turkmenistan, Ukraine
C	Antigua and Barbuda, Benin, Burundi, Botswana, Cameroon, Central African Republic, Chad, Comoros, Djibouti, Equatorial Guinea, Ethiopia, Gabon, Ghana, Guinea, Ivory Coast, Kenya, Lesotho, Malawi, Mauritania, Namibia, Niger, Niue, Oman, Puerto Rico, Rwanda, Seychelles, Solomon Islands, Swaziland, Tanzania, Togo, Tonga, Tuvalu, Uganda, United Arab Emirates, Vanuatu

Querying for an Address Record using *PrimaryAddressLine1*

When using the *PrimaryAddressLine1* field to query for an address record, the value returned is the value contained in the mapped field for the specified country. For example, when querying for an address with <Country>Canada</Country>, the PrimaryAddressLine1 field is mapped to the Address field:

```
<?xml version="1.0" encoding="UTF-8"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/
2001/XMLSchema">

<SOAP-ENV:Body><ns:AccountQueryPage_Output xmlns:ns="urn:crmondemand/ws/ecbs/account/
">

   <ListOfAccount xmlns="urn:/crmondemand/xml/Account/Data" lastpage="true">

   <Account>
```

```
    <Location>Toronto</Location>
    <AccountName>ACCOUNTTEST1</AccountName>
    <ListOfAddress lastpage="true">
      <Address>
        <Id>1QA2-R7C3O</Id>
        <StreetAddress3></StreetAddress3>
        <Country>Canada</Country>
        <County></County>
        <Description></Description>
        <Province>ON</Province>
        <ZipCode>M2H 3G5</ZipCode>
        <City>Toronto</City>
        <IntegrationId>1QA2-R7C3O</IntegrationId>
        <Address>100 Main Street</Address>
        <StreetAddress2></StreetAddress2>
        <PrimaryAddressLine1>100 Main Street</PrimaryAddressLine1>
      </Address>
  ...
    </ListOfAddress>
  </Account>
  </ListOfAccount>
  </ns:AccountQueryPage_Output>
  </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

whereas, when the <Country> value is Togo, the PrimaryAddressLine1 field maps to the County field:

```
<?xml version="1.0" encoding="UTF-8"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/
2001/XMLSchema">

<SOAP-ENV:Body><ns:AccountQueryPage_Output xmlns:ns="urn:crmondemand/ws/ecbs/account/
">
```

```
<ListOfAccount xmlns="urn:/crmondemand/xml/Account/Data" lastpage="true">
<Account>
  <Location>Togo</Location>
  <AccountName>ACCOUNTTEST2</AccountName>
  <ListOfAddress lastpage="true">
     <Address>
        <Id>1QA2-R7IMS</Id>
        <StreetAddress3></StreetAddress3>
        <Country>Togo</Country>
        <County>10222</County>
        <Description></Description>
        <Province></Province>
        <ZipCode></ZipCode>
        <City>Lomé</City>
        <IntegrationId>1QA2-R7IMS</IntegrationId>
        <Address></Address>
        <StreetAddress2></StreetAddress2>
        <PrimaryAddressLine1>10222</PrimaryAddressLine1>
     </Address>
...
  </ListOfAddress>
</Account>
</ListOfAccount>
</ns:AccountQueryPage_Output>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Inserting or Updating an Address Record using *PrimaryAddressLine1*

When inserting or updating an address record using the *PrimaryAddressLine1* field, the value provided in the *PrimaryAddressLine1* field is written to the primary address field based on the Country value provided in the request. If a value is provided for both the *PrimaryAddressLine1* field and the primary address field (for example, County) for the specified country, the value in the *PrimaryAddressLine1* field is respected and the value in the primary address field is ignored.

This is shown in Table 10.

Table 10. Value specified for *PrimaryAddressLine1* and Primary Address Field

Field Name	SOAP Request	Value Written to DB
Country	Togo	Togo
County	BP 128	1 Main Street
Street Address 1	Not applicable	None
PrimaryStreetAddress1	1 Main Street	None

In the case where only a *PrimaryAddressLine1* value is submitted, this value is written to the mapped field in the DB:

Table 11. Value specified for *PrimaryAddressLine1* only

Field Name	SOAP Request	Value Written to DB
Country	Togo	Togo
County	Not applicable	1 Main Street
Street Address 1	Not applicable	None
PrimaryStreetAddress1	1 Main Street	None

Objects Supporting the *PrimaryAddressLine1* Field

The *PrimaryAddressLine1* field is available on a number of objects accessible through the Web Services v2.0 interface as shown in Table 12 and Table 13.

Table 12. Parent Objects on Which the *PrimaryAddressLine1* field is available

Object Name	Fields
Account	BillingPrimaryAddressLine1, ShippingPrimaryAddressLine1
Contact	PrimaryAddressLine1, AlternateAddressLine1
Lead	BillingPrimaryAddressLine1

Table 13. Child Objects on Which the *PrimaryAddressLine1* field is available

Parent Object Name	Child Object Name	Fields
Account	Address	Not applicable
Account	Contact	Not applicable
Contact	Address	PrimaryAddressLine1
Contact	Lead	BillingPrimaryAddressLine1

Web Services Utilization

In the Oracle CRM On Demand application, the Web Services Utilization page provides detailed information on your company's Web services usage, both current and historical.

For each Web services request, Oracle CRM On Demand logs the following information:

- **Session Id.** An identifier representing the session used to process a Web services request.
- **Web Service Name**. The name of the Web service that was executed.
- **Operation**. The operation that was performed.
- **Start Time**. The date and time the request began processing.
- **End Time**. The date and time the request completed processing.
- **Web Service Space**. The namespace for the request that was executed.
- **User Alias**. The alias of the user whose credentials were used to authenticate with.
- **Output Message Size (Bytes)**. The size of the response message in bytes.
- **Entry Type**. Either Login, Logout, or Dispatch.
- **Input Message Size (Bytes)**. The size of the input message in bytes.
- **Web Service Client Name**. The value provided in the <ClientName> parameter in the SOAP request.
- **# of Operations**. The number of operations performed by Oracle CRM On Demand for the request.
- **Error Message**. If the request resulted in an error, it is displayed, otherwise this field remains empty.

For more information about the Web Service Client Name parameter, see "Web Service Client Name Identification" on page 41.

The Web Services Utilization page supports Oracle CRM On Demand list management capabilities, allowing administrators to filter the list of entries and to export the data for further analysis in other applications.

You can also use the UserUsageQueryPage method to retrieve information about Web services utilization. For more information about this method, see "UserUsageQueryPage" on page 197.

See the Oracle CRM On Demand online help for more information on using the Web Services Utilization page.

Web Service Client Name Identification

To allow accurate tracking of requests in the Web Services Utilization page, client applications require a mechanism to identify themselves in each Web service request that is sent to Oracle CRM On Demand. The SOAP header parameter, <ClientName> provides such a mechanism.

The <ClientName> parameter is optional, and is supported for both stateful and stateless web services operations.

NOTE: The <ClientName> parameter is not currently supported for the Administrative Services APIs.

Supported Client Name Characters and Usage

The <ClientName> value passed in the SOAP header is validated by Oracle CRM On Demand. The following characters are supported in the <ClientName> value:

- UnicodeLetterOrDigit characters, that is, the set of Unicode characters identified as either a letter or a digit
- Spaces
- Commas

Any value passed in through the <ClientName> parameter that contains characters other than those specified above is not accepted by Oracle CRM On Demand. The request is still processed however, and the value `Invalid Client Name` is displayed in the Web Services Utilization page. The <ClientName> value is restricted to 100 characters; for any value longer than 100 characters, `Invalid Client Name` is displayed in the Web Services Utilization page.

It is also recommended that the following convention be used when specifying the <ClientName> value:

```
[Developer], [Client Name]
```

For example, an application developed by XYZ Consulting called Account Synchronization Utility can use the following:

```
XYZ Consulting, Account Synchronization Utility
```

This allows the customer to track not only which application has sent a request but also who to contact if the an issue is discovered.

Sending the Client Name in Stateless Web Services Requests

Every stateless Web service request that requires tracking of the client name must include the <ClientName> element in the SOAP header, with the namespace "urn:crmondemand/ws" (or the namespace may be defined at the root level). This is shown in the following example:

```
<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:xsd="http://
www.w3.org/2001/XMLSchema">

<soap:Header>

   <wsse:Security>

      <wsse:UsernameToken>

         wsse:Username>USERNAME</wsse:Username>

         <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>

      </wsse:UsernameToken>

   </wsse:Security>

   <ClientName xmlns="urn:crmondemand/ws">Oracle Corporation, Web Services On Demand
Guide</ClientName>

</soap:Header>

<soap:Body>

   <AccountQueryPage_Input xmlns="urn:crmondemand/ws/ecbs/account/10/2004">

      <ListOfAccount xmlns="urn:/crmondemand/xml/account/">

         <Account>

            <AccountName>LIKE 'a1'</AccountName>

            <Location/>

         </Account>

      </ListOfAccount>

   </AccountQueryPage_Input>

</soap:Body>

</soap:Envelope>
```

A stateless request execution may or may not result in an explicit login operation in Oracle CRM On Demand:

- If a stateless request execution results in explicit login, two entries are created in the Web Services Utilization page. Both the entries for this request, that is, the login and operation execution, show the client name specified in the SOAP request.
- If a stateless request execution does not result in explicit login, a single entry is created in the Web Services Utilization page, and it has the client name specified in the SOAP request.

Sending the Client Name in Stateful Web Services

A stateful Web service request execution involves:

1. **Stateful login**. A one time operation, which covers both login with username and password as well as SSO login.
2. **Stateful request execution**. Multiple request operations using the session ID returned by the login operation.

For a stateful request:

- If the stateful request requires tracking of the client name, it must be specified in the stateful login operation.
- If a client name is specified in a stateful request execution, it is ignored.
- All the stateful requests executed with the session ID returned by the stateful login request are displayed in the Web Services Utilization page with the client name specified in the login operation.

Stateful Login

The login operation can be a HTTP request or a SOAP over HTTP request (R16 compatibility mode).

When the stateful login is a HTTP request, the client name is sent as the HTTP header parameter X-ClientName.

For a login with username and password:

```
GET http://<servername>:<portno>/Services/Integration?command=login

Http Header:

username: <username>

password: <password>

X-ClientName: Oracle Corporation, Web Services On Demand Guide
```

For an SSO login:

```
GET http://<servername>:<portno>/Services/

Integration?command=ssologin&odSsoToken=[Token Value]

X-ClientName: Oracle Corporation, Web Services On Demand Guide
```

Web Services R16 Compatibility Mode

If Web Services R16 Compatibility Mode is enabled, a stateless request is treated as stateful and returns a session ID. For SOAP requests when R16 Compatibility Mode is enabled:

- The client name specified in the SOAP Header is used for the login operation and stateful operation execution
- With the returned session ID, for subsequent requests, if the client name is specified in the SOAP header, it is ignored.

- As for stateful requests, the client name with which login occurs (that is, the first SOAP request in this case) is displayed in the Web Services Utilization page with all requests for the stateful cycle.

```
<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:xsd="http://
www.w3.org/2001/XMLSchema">

<soap:Header>

   <wsse:Security>

      <wsse:UsernameToken>

         wsse:Username>USERNAME</wsse:Username>

         <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>

      </wsse:UsernameToken>

   </wsse:Security>

   <ClientName xmlns="urn:crmondemand/ws">Oracle Corporation, Web Services On Demand
Guide</ClientName>

</soap:Header>

<soap:Body>

   <AccountQueryPage_Input xmlns="urn:crmondemand/ws/ecbs/account/10/2004">

      <ListOfAccount xmlns="urn:/crmondemand/xml/account/">

         <Account>

            <AccountName>LIKE 'a1'</AccountName>

            <Location/>

         </Account>

      </ListOfAccount>

   </AccountQueryPage_Input>

</soap:Body>

</soap:Envelope>
```

About Service Allotments

Service allotments provide insight to customers regarding their usage of Oracle CRM On Demand and also promote equitable use of resources among all customers. Customers who understand their usage of Oracle CRM On Demand can improve user adoption of the application and can also optimize their usage both in the UI and their integrations.

The service allotments for Web service usage include the following:

- **Web Services Operations Allotment**. The number of distinct operations performed by a company over a 24 hour window.
- **Web Services Concurrent Request Allotment**. The maximum number of stateless Web service requests that can be processed at any point in time.

For service allotments, usage from all Web service clients, including those developed by Oracle, as well as those developed by customers and third parties is measured.

In the Oracle CRM On Demand UI, company administrators can view service allotment usage through the links under the Admin, Company Administration, Service Allotment Administration section. By selecting the Service Allotment Administration link, administrators can view details of their allotments, and current and remaining usage. By selecting the Service Allotment Usage History link, administrators can view historical usage for all of their service allotments.

The Web Service Utilization page provides additional details regarding Web service usage. This page can now be accessed either from the Admin homepage or the Company Administration page through a link under the Service Allotment Administration section. Administrators can use this page to see the operations used for each Web service request issued.

NOTE: See the Oracle CRM On Demand online help for more information about service allotment administration.

Determining Current Usage

The Web Services Operations allotment is measured using a 24-hour rolling window. Current usage is displayed in the Oracle CRM On Demand UI or can be retrieved using the Service Allotment Web service (see “Service Allotment” on page 427). Current usage reflects the usage for the current hour plus the previous 23 hours.

For example, at 9:30 A.M., the current usage window extends from 10 A.M on the previous day, until the end of the current hour (10 A.M. today). All operations usage during this period is added together to calculate a company's current usage.

When the current hour elapses, the 24-hour window shifts, releasing any usage from the first hour of the previous window. For example, if a company has used 1000 operations in the current 24-hour window, 100 of which were used during the first hour, when the current hour elapses, the current usage is reduced to 900 operations.

Determining Historical Allotment Usage

Historical allotment usage is displayed in the Oracle CRM On Demand UI in a Related Information applet on the Service Allotment Detail page. You can retrieve this information for analysis or archiving using the following methods:

- The Allotment Usage Web service (see "Allotment Usage" on page 403)
- The Export Assistant
- The List Management Export feature in the Service Allotment Usage History page under Company Administration.

When a Service Allotment Is Reached

If the current usage reaches the service allotment value for a company for the Web Services Operations allotment, further Web service requests are not processed until the 24-hour window shifts and capacity is released. To help avoid this situation, your administrator can configure email alerts to inform one or more users that your company is approaching the service allotment value.

See the Oracle CRM On Demand online help for more information on configuring email alerts for service allotments.

NOTE: If your company requires additional capacity, contact your Oracle CRM On Demand sales representative for information.

For information about best practices, see "Best Practices for Adhering to Web Service Allotments" on page 76.

Calculation of Allotment Usage

The following topics describe how usage is calculated for each allotment.

Web Services Operations Allotment

The Web service operation count is incremented whenever a Web service request is received and executed. A single Web service SOAP request, when processed, may result in one or more Web service operations being executed. For example, Table 14 shows the number of operations resulting for different types of request.

Table 14. Examples of Number of Operations for Different Web Services Requests

Type of Request	Number of Operations
Non-query operations	
Account insert request containing a single Account record (with no child operations)	1

Table 14. Examples of Number of Operations for Different Web Services Requests

Type of Request	Number of Operations
Contact update request containing 10 Contact records (with no child operations)	10
Account update request containing a single Account record with 3 Account Team records	4
Query operations	
Simple query for a set of Accounts	1
Query for a set of Accounts and the associated Contacts for each Account	■ 1 operation to retrieve the set of Accounts ■ 1 operation to retrieve the set of Contacts for each Account

Web Services Concurrent Request Allotment

The Web Services Concurrent Request allotment is a measure of the number of Web service requests (including both stateful and stateless requests) being processed by a company concurrently.

3 Getting Started with Web Services

This chapter provides an overview of how to get started with Oracle CRM On Demand Web Services. It contains the following topics:

Accessing Data in Oracle CRM On Demand

A customer who wants to access data in Oracle CRM On Demand from a Web services-enabled client must perform the following tasks:

1. **Request Web Services Integration.**
2. **Download WSDL files, and Schema files if required.**
3. **Incorporate WSDL files into the development environment.**

Requesting Web Services Integration

On request, an Oracle CRM On Demand Customer Care representative enables the Oracle CRM On Demand Integration capability for your company by limiting access to Web services to particular roles.

Starting with Oracle CRM On Demand Release 17, Customer Care can grant two privileges:

- **Enable Web Services Access.**

  Users whose roles have been granted this privilege can log in to Oracle CRM On Demand through Web services, and issue requests using the Web services API.

  By default, this privilege is enabled for the Administrator role for new companies.

- **Manage Access to Web Services**.

  Users (usually administrators) who have been granted this privilege can grant other roles the Enable Web Services Access privilege.

if your company existed before Release 17, and you have access to Web services, you do not need to take any action unless you wish to restrict access to Web services, in which case you must do the following:

1 Contact Customer Care and request that the Administrator (or equivalent custom role) be granted the Manage Access to Web Services Integration privilege.

2 Configure roles. The Administrator grants or revoke access to Web services on a per role basis.

Enabling Stateless Web Services Support

Support for stateless Web services requests is enabled by Customer Care or the company administrator through the Web Services R16 Compatibility Mode setting on the Company Profile page in Oracle CRM On Demand. This check box controls whether Web services requests are processed as stateless requests or as stateful requests as in Release 16 of Oracle CRM On Demand.

The Web Services R16 Compatibility Mode check box is cleared by default for all new companies, therefore support for stateless Web services is enabled by default for new customers, and disabled by default for companies who were customers before Release 17. It is recommended to clear the check box and thus enable stateless Web Services.

If stateless Web services support is enabled, the namespace specified in requests is used to determine whether or not a request is processed as a stateless request:

- Specifying the WSSE Version 1.0 security namespace results in the request being treated as a stateless request. In this case, session pooling is used to manage user sessions (that is, no session ID is returned in the response). This conforms to WS-I specifications.
- Specifying the WSSE draft namespace results in the request being treated as a stateful request, in which a session ID value is returned in the response (this was the behavior before Release 17).

For more information about these namespaces, see “WSSE Namespace Support” on page 24.

If stateless Web services support is disabled, requests containing the username and password tokens are treated as stateful requests, regardless of the specified namespace value. Responses are returned with a session ID that can be used in subsequent requests to reuse an existing session.

Downloading WSDL files and Schema Files

Your company’s designated Oracle CRM On Demand administrator accesses the Web Services Administration page located under the Admin link in Oracle CRM On Demand to download Web Service Description Language (WSDL) files that can be used to create applications that interact with Oracle CRM On Demand through Web services. The WSDL files are divided into the following types, corresponding to APIs:

- Web Services v1.0 and Web Services v2.0 APIs, which provide access to the data your company has stored within Oracle CRM On Demand
- The Service APIs, which allow administrators to perform actions such as changing passwords or retrieving integration events
- The Administrative Services APIs, which allow access to your company's configuration and can be used to retrieve or change the way you have configured Oracle CRM On Demand programmatically.

You can download WSDL files for the Web Services v1.0 and Web Services v2.0 APIs in either a Custom or Generic version.

- **Downloading Custom WSDL**. For Custom WSDL files, the XML tags for custom fields are based on the Integration tags configured in Field Setup. Custom WSDL files allow you to generate WSDL files that are specific to your company and which use your company's field naming conventions.
- **Downloading Generic WSDL**. For Generic WSDL files, custom fields are based on generic XML tags: CustomNumber0, CustomCurrency0, and so on. Using these placeholders, together with the Mapping Service or Field Management API allows applications to map the generic XML tags to the field names that your company uses.

For Web Services v1.0, and Web Services v2.0 objects for which Integration Events support is offered, you can also download Custom or Generic Schema files, as described in Downloading Schema Files.

To download a WSDL file

1. Navigate to the Web Services Administration page.
2. From the Select Service drop-down list, select Web Services v1.0, Web Services v2.0, Service, or Administrative Services APIs as required.
3. From the WSDL Object drop-down list, select the required record type, for example, Account, or the name of the service API.

   The objects displayed in the WSDL Object drop-down list depend on the record types that are set up for your company.
4. Click either the Download Custom WSDL or Download Generic WSDL button to open a popup window containing the selected WSDL file.
5. Save the WSDL file to your computer.

Downloading Schema Files

The Integration Events Web service (part of the Service API) imports Schema (XSD) files from its WSDL, therefore you must download the required Schema files for the objects included in the WSDL file. You can download Custom or Generic Schema files for all objects for the Web Services v1.0 API, as well as Web Services v2.0 objects for which Integration Events support is offered.

If you create custom fields or rename fields for a record type, you cannot use the Generic XSD files for tracking these fields. Instead, you must download a Custom XSD file using the Download Custom Schema button in the Web Services Administration page.

For more information about the Integration Events Web service, see “Preparing the Integration Event WSDL File and Schema Files” on page 157.

To download a Schema file

1. Navigate to the Web Services Administration page.
2. From the Select Service drop-down list, select Web Services v1.0 or Web Services v2.0.

3 From the WSDL Object drop-down list, select the required record type, for example, Account.

   The objects displayed in the WSDL Object drop-down list depend on the record types that are set up for your company.

4 Click one of the following buttons:

   - Download Custom Schema
   - Download Generic Schema

   A page containing the Schema is displayed. Depending on whether you select Download Custom Schema, or Download Generic Schema, custom fields are displayed differently in the Schema. For Custom Schema, the XML tags for the custom fields are based on the Integration tags from Field Setup. For Generic Schema, the custom fields are based on generic XML tags.

   NOTE: If you selected Web Services v2.0, and Schema file generation is not supported for the selected record type, an error message is displayed.

5 Save the Schema file to your computer.

Incorporating WSDL Files into the Development Environment

To use the downloaded WSDL files, the company incorporates the WSDL files into its Web services development environment—for example, by generating .NET, Java, or C# (C Sharp) proxy classes.

For some information about best practices when generating proxy classes, see “Avoiding Proxy Class Compilation Failure Due to the Java 64KB Limit” on page 70.

Review

4 Establishing and Managing the Web Services Session

This chapter describes the different ways in which an Oracle CRM On Demand Web services sessions can be established and maintained. It contains the following topics:

About Establishing and Managing the Web Services Session

A Web services-enabled client (that is, a client written in any language that interacts with the Web services framework), must establish a secure session with Oracle CRM On Demand. After authentication, the client interacts with the published Oracle CRM On Demand Web Services to perform data retrieval, modification, creation, and deletion operations. Oracle CRM On Demand and the client format requests and resulting data as standard XML/SOAP messages.

There are multiple ways in which client applications can establish authenticated sessions with Oracle CRM On Demand. These sessions can be managed by the client or by the Oracle CRM On Demand server based on the login mechanism used to establish the session.

Oracle CRM On Demand Stateful and Stateless Request Support

Oracle CRM On Demand offers two different mechanisms to manage the allocation of resources to users who are interacting with Oracle CRM On Demand through Web services:

- Stateless requests
- Stateful session management

Stateless Web services transactions are those in which the client application does not need to maintain a session identifier (a JSESSIONID value) to perform multiple requests to Oracle CRM On Demand using the same session. Stateless Web services transactions are ideally suited for situations in which an interactive application is used to integrate with Oracle CRM On Demand.

Oracle CRM On Demand Authentication Mechanisms

Oracle CRM On Demand offers the ability to log in using either a stateful or stateless mechanism when performing a Web service operation. Stateful login can be used with the Web Service v1.0, Web Service v2.0, Service, and Bulk Data APIs, but is not available for the Administrative Services APIs. Stateless login is available on all APIs.

Stateless Web services requests can be authenticated using the following mechanisms:

- UserName and Password provided in SOAP security header (using WSSE Version 1.0 Namespace)
- HTTP Login (with Oracle CRM On Demand Single Sign On (SSO) Token in HTTP Header)
- SAML v1.0 Assertion provided in SOAP security header (using WSSE Version 1.0 Namespace)

NOTE: For Administrative Services APIs only, the following login option is supported for stateless Web services requests.

- UserName and Password provided in SOAP security header (using WSSE Draft Namespace)

The following login options are supported by Oracle CRM On Demand for stateful requests:

- HTTP Login (containing UserName and Password in the HTTP Header)
- UserName and Password provided in SOAP security header (using WSSE Draft Namespace)
- HTTP Login (with Oracle CRM On Demand Single Sign On (SSO) Token in HTTP Header)
- SAML v1.0 Assertion provided in SOAP security header (using WSSE Version 1.0 Namespace)

Using Stateless Web Service Requests

Interactive applications can be developed such that a user's credentials are supplied within a request sent to Oracle CRM On Demand, eliminating the need for an explicit login request. If the request qualifies as a stateless request (see “Enabling Support for Stateless Requests” on page 55), the Oracle CRM On Demand server checks to see whether a session has already been established for that user. If a session is found, it is reused for the new request. If no existing session is found for the user, a new session is established. Each user is limited in the number of concurrent sessions that he or she can establish.

When a request is received by the server, a server ID value is returned in the response. This value does not identify a specific user session, but rather is used to identify the server in the Oracle CRM On Demand environment on which the session has been created. Subsequent requests can be sent that include the server ID value, thus ensuring that the request is routed to the same server as the original request.

NOTE: Providing a server ID in subsequent requests is optional, however it is recommended that this value is included.

There is no logoff operation required for stateless Web service requests. Sessions will eventually be released either due to time-out, or will be reclaimed by session management routines for use by other users.

Enabling Support for Stateless Requests

To establish a session that is managed on the Oracle CRM On Demand server:

- A company must have stateless Web services support enabled, see "Enabling Stateless Web Services Support" on page 50.
- The WS-I login mechanism must be used with the correct namespace and formats as described by the WS-I basic security profile at

  http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html#UsernameToken:

  - The WSSE security namespace (http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd) must be specified in the request.

Requests that meet these criteria qualify as stateless requests.

Maintaining a Stateless Java Session

The optional SOAP header parameter <SessionKeepAlive> allows a client to instruct the server to leave the Java session open after a stateless request has been processed. By specifying the optional <SessionKeepAlive> parameter with a value of `true` in the SOAP header of a stateless request, the Java session that is established following authentication will not be logged off by Oracle CRM On Demand after the request has been processed, and OM session pooling will continue to be used. In addition, the JSESSIONID value that is returned in the SOAP response can be used to authenticate future requests allowing them to re-use the same Java session, without having to re-authenticate. This is especially useful when SAML assertions are used for authentication, as it eliminates multiple round-trips to the Identity Provider when multiple requests are being processed for the same user.

Unlike stateful Web services sessions, there is no logoff command. Java sessions will be logged off by the Oracle CRM On Demand server following a period of inactivity. Any attempt to use a JSESSIONID value for a Java session that has expired, will result in an error being returned to the client for that request. Stateless requests must provide either a valid JSESSIONID value or valid user credentials (SSO Token, SAML Assertion or UserName and Password) to authenticate. Any request lacking either a valid JSESSIONID value or valid user credentials will be rejected.

NOTE: A request that contains both a JSESSIONID value as well as user credentials will be re-authenticated.

Stateless Login Mechanisms

The following topics describe the various mechanisms for logging in and making integration requests when making stateless Web services requests.

Logging in with UserName and Password in the SOAP Security Header

The ability to supply a user's credentials is due to support for the UserNameToken profile of the WS-I Basic Security Profile Version 1.0. In this case, the SOAP header contains the element <wsse:UsernameToken>, which has child elements containing a username and password:

```
<soap:Header>

    <wsse:Security soap:mustUnderstand="1">

      <wsse:UsernameToken>

        <wsse:Username>admin@test.com</wsse:Username>

<wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">adminOnDemand</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

</soap:Header>
```

NOTE: URL encoding of login credentials is not supported when they are provided in the SOAP security headers.

For the Administrative Services API, a similar login mechanism is used, but with a WSSE draft namespace instead of the WSSE Version 1.0 Namespace.

Logging in Using Single Sign-On

The Single Sign-On (SSO) feature of CRM On Demand allows companies to integrate the hosted Oracle CRM On Demand service with other systems that have the ability to manage user credentials and authentication.

If your company has been set up to use SSO for Oracle CRM On Demand, the following steps are used to log in and retrieve the session ID.

1. The Web service client makes a request with the following command specifying the SSO Company Identifier.

   ```
   https://secure-ausomx[POD].crmondemand.com/Services/
   Integration?command=ssoitsurl&ssoid=<SSO_Company_Identifier>
   ```

2. The server returns the SSO ITS URL in the "X-SsoItsUrl" HTTP header of the response
3. The Web service makes a request with the ITS URL and retrieves a session ID.

Any client using Inbound SSO must be able to perform the following actions:

- Follow redirects
- Accept cookies

Furthermore, the identity provider may respond to requests with a prompt for authentication credentials, for example, username and password. The client application must be able to recognize this request and respond appropriately.

NOTE: The behavior of the identity provider is beyond the control of Oracle CRM On Demand. Customers are responsible for ensuring that their client applications are compatible with their chosen identity provider.

For detailed information about Single Sign-On, see the White Paper available from Customer Care.

For code samples for single sign-on see: https://codesamples.samplecode.oracle.com/servlets/Scarab/action/ExecuteQuery?query=crm_on_demand

The outbound SSO feature allows users who have signed into Oracle CRM On Demand using SSO to pass the SSO credentials from Oracle CRM On Demand to third-party sites such as corporate Web pages or intranets. This allows users to embed or access third-party sites from within Oracle CRM On Demand.

Outbound SSO in Oracle CRM On Demand uses a proprietary method to generate a hashed message authentication code (HMAC) token that is passed to the third-party site. This third-party site makes a request back to Oracle CRM On Demand with the token. Oracle CRM On Demand then validates the token and provides a username back to the third-party site, or authenticates the token and provides a session ID to the user.

Outbound SSO Methods

Two methods are available as part of outbound SSO:

1. **SSO Token Validation**. The following steps are used to validate an SSO token:
   a. The third-party application makes a request with the following command specifying the SSO token:
   b. https://server/Services/SSOTokenValidate?odSsoToken = "ssotoken value"
   c. The server returns the username in the response.
2. **Login using SSO Token**. The following steps are used to obtain a session ID using the SSO token:
   a. The third party application makes a request with the following command specifying the SSO token:
   b. https://server/Services/Integration?command=ssologin&odSsoToken="ssotoken value"
   c. The server returns the session ID in the response, which is used for access to data within Oracle CRM On Demand.

For detailed information about outbound SSO, see the Customer Care Portal - Web services resource library.

Logging in Using SAML Assertion

The following is an example of a SOAP security header containing a SAML v1.0 Assertion:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>

<soap:Envelope

  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"

  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

  xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd"

  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
  xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
<soap:Header>
 <wsse:Security>
   <saml:Assertion
      AssertionID="id-ejZZo3FyrQcenpRKN9d4MTNHFOA-"
      IssueInstant="2011-01-05T19:30:49Z"
      Issuer="http://sdchs20i117.us.oracle.com:7499/fed/idp"
      MajorVersion="1"
      MinorVersion="1">
     <saml:Conditions NotBefore="2011-01-05T19:20:49Z"
NotOnOrAfter="2011-01-05T19:45:49Z">
       <saml:AudienceRestrictionCondition>
         <saml:Audience>http://sdchs20i117.us.oracle.com:7499/fed/sp/samlv11sso
         </saml:Audience>
       </saml:AudienceRestrictionCondition>
     </saml:Conditions>
     <saml:AuthenticationStatement
       AuthenticationInstant="2011-01-05T19:30:49Z"
       AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
       <saml:Subject>
         <saml:NameIdentifier
Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName">ADMIN@KAI.COM
         </saml:NameIdentifier>
           <saml:SubjectConfirmation>
             <saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer
           </saml:ConfirmationMethod>
         </saml:SubjectConfirmation>
       </saml:Subject>
     </saml:AuthenticationStatement>
     <dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
```

```
        <dsig:SignedInfo>
          <dsig:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
          <dsig:SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
          <dsig:Reference URI="#id-ejZZo3FyrQcenpRKN9d4MTNHFOA-">
            <dsig:Transforms>
              <dsig:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
              <dsig:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
            </dsig:Transforms>
            <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <dsig:DigestValue>cwrWFitulBs6yoASwCcAOgQr8yw=</dsig:DigestValue>
          </dsig:Reference>
        </dsig:SignedInfo>
       <dsig:SignatureValue>HJHgg8XGs8miAyQnKdgBw5Af8KI0Vsukt4abjw7TgYRu18oiMx+ugBjWBk/
UoIujMdcP2gEBQvcdJ7jJFtYsy7tcuNk/JAoevyarvkF2U9RkdoHAqoTY8FuwOELKhIqiQZgkOA0Eip/
fyxyS9jBrCegBoDo64U1InCxcMyrUUa8=</dsig:SignatureValue>
      </dsig:Signature>
  </saml:Assertion>
  <wsse:SecurityTokenReference>
     <wsse:KeyIdentifier
       ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.0#SAMLAssertionID">id-ejZZo3FyrQcenpRKN9d4MTNHFOA-</wsse:KeyIdentifier>
  </wsse:SecurityTokenReference>
 </wsse:Security>
</soap:Header>
<soap:Body>
  <AuthWS_AuthAuthorizationUser_Input xmlns="urn:crmondemand/ws/auth/"/>
</soap:Body>
</soap:Envelope>
```

Using Stateful Web Services Requests

In instances in which the application manages the creation and release of user sessions, the following apply:

- Clients must make login and logoff calls in their code to manage the session.
- The login step returns an HTTP cookie that contains the session identifier that must be used for making additional requests.
- A session remains active until the user explicitly logs out or until the session times out.

Web services session management is HTTP-based and uses a session ID (also known as a JSESSIONID), which is contained in HTTP Session cookies, to uniquely identify a session established with Oracle CRM On Demand.

Oracle CRM On Demand Web Services enable session management by first creating a session using the login call, which is then referenced in any subsequent SOAP operations.

In an Oracle SOAP session, after a session ID has been created in a login request, it can be referenced in one of these ways:

- The session ID can be attached as a parameter to the URL request line. When a session ID is present in the URL line, it is identified by the string “jsessionid” in lowercase, followed by the exact session ID, which is coded using URL syntax.
- The session ID can be part of the cookie header line. When a session ID is referenced as a cookie, a cookie header line must appear in the SOAP request with the name JSESSIONID=. In this case, the session ID appears in uppercase, and the value of the cookie is exactly the same as the session ID received from the login request.

  NOTE: This is the recommended approach to referencing the session ID, for several reasons. The Java servlet specification advises the use of cookies and not the URL wherever possible. Most development environments and programming languages are efficient in using cookies rather than adding arguments to the URL. Furthermore, it is much simpler to implement cookies because the cookie container from the Oracle CRM On Demand login response can be copied onto further requests being sent to Oracle CRM On Demand.

When a login request is made, the session ID is returned as a cookie in the response to the request. The client is responsible for extracting this session ID and using it throughout the session. If the session times out for any reason, the error returned reports that the session is not valid and the client must then request a new session. In this case, no explicit logoff operation is required.

To issue a logoff request to terminate a session, the request must identify the session using the JSESSIONID value.

All requests must use Secure Sockets Layer (SSL) over HTTP (HTTPS).

Stateful Login Mechanisms

The following topics describe the various mechanisms for logging in and making integration requests when making stateful Web services requests.

Logging in Using HTTP GET

An HTTPS request can be used to instantiate an Oracle CRM On Demand Web services session and obtain a valid session ID. A client invokes login by sending an HTTP GET request to a URL like the following:

```
https://secure-ausomx[ENV].crmondemand.com/Services/Integration?command=login
```

where [ENV] is the three-letter identifier for your company's environment. If you do not know what this value is for your company, refer to the URL you use to access the Oracle CRM On Demand UI.

NOTE: The login parameter value is case sensitive.

Login Input

There are different mechanisms for login depending on whether the login header contains URL encoding with the UTF-8 encoding system. This is necessary when login credentials contain multi-byte characters.

The input to login is provided in the URL parameters and the HTTP headers, as follows:

- Two URL parameters:
  - command, which has the value login
  - isEncoded, which is used if the HTTP headers are URL encoded using UTF-8. This parameter must have the value Y or y if encoding is required. The default value is N or n.
- Two HTTP headers, UserName and Password, must be set with the appropriate values for your system. For example:
  - UserName: johndoe@email.com
  - Password: mypass

  HTPP headers can be in clear text, or can be URL encoded.

Login Output

The login command returns the following items:

- A session cookie, JSESSIONID. The client must use this cookie when submitting subsequent requests, including logoff requests.
- A status code of 200, if the session does not encounter any errors. This indicates that the request succeeded.

For code samples for login, see: https://codesamples.samplecode.oracle.com/servlets/Scarab/action/ExecuteQuery?query=crm_on_demand

It is also possible to log in at the same time as making an integration request; for more information, see "Integration Requests for the Web Services Session" on page 63.

Logging in with UserName and Password in the SOAP Security Header

A similar login mechanism to that used for stateless request is used, but with a WSSE draft namespace instead of the WSSE Version 1.0 Namespace. For more information, see “Logging in with UserName and Password in the SOAP Security Header” on page 62.

Logging in Using Single Sign-On

This is the same as for stateless Web service requests, see “Logging in Using Single Sign-On” on page 62.

Logging in Using SAML Assertion

This is the same as for stateless Web service requests, see “Logging in Using SAML Assertion” on page 62.

Logging Out of the Web Services Session

A client logs out by sending an HTTP POST or HTTP GET request to a URL. For example:

```
https://secure-ausomx[ENV].crmondemand.com/Services/Integration?command=logoff;
```

NOTE: The parameter value logoff is case sensitive.

Logoff Input

The preferred method is where the JSESSIONID returned to the client during login must be included with the request as a cookie with the same name. There are no other URL parameters or HTTP headers, and there is no HTTP body.

Logoff Output

A status code of 200 is returned if the request is received by the Oracle CRM On Demand server. This does not guarantee that the logoff request has been processed, and there may be a delay until the session resources are released for use by a new Web services session.

Alternative Logoff

The JSESSIONID can be included in the URL instead of a cookie if the user wants. The following URL is what the user would use, where *XXXX* is the JSESSIONID.

```
https://secure-ausomx[ENV].crmondemand.com/Services/
Integration;jsessionid=XXXX?command=logoff
```

Integration Requests for the Web Services Session

An integration request is an HTTPS request to invoke a Web service to perform data creation, retrieval, update, and deletion operations. An integration request is made by an HTTP POST command to a URL like the following:

https://secure-ausomx[ENV].crmondemand.com/Services/Integration/*object*

where *object* is the name of the relevant Oracle CRM On Demand object (record type). This Oracle CRM On Demand object is determined from the contents of the SOAP request.

Integration request input. The JSESSIONID returned to the client during login must be included with the request. The request must contain the JSESSIONID either as a cookie or as a URL parameter, as follows:

https://secure-ausomx[ENV].crmondemand.com/Services/Integration/*object*;
jsessionid=xyZ12489w3482413

The Web service input is provided as a SOAP command in the body of the HTTP POST request.

Integration request output. The properties returned by the HTTP server populate the response headers and the response body. Table 15 shows the top-level properties that specify key properties of the HTTP response.

Table 15. Properties of the HTTP Response

Property	Comments
HttpStatus	Status code returned in the response. If no value is provided, the response is given the value 200 (indicating success).
Content-Type	Content type returned in the response. If no value is provided, the response is given the value text/xml.

Comparison of Stateless Versus Stateful Web Services Transactions

Stateless Web services transactions:

- Support OM session pooling
- Perform a logoff (invalidation of the Java session) after processing each request unless the <SessionKeepAlive> parameter is set to true
- Return a JSESSIONID value to help maintain server affinity (that is, direct subsequent requests to the same Java server)

Stateless Web services transactions for Administrative Services APIs:

- Similar to Stateless Web Services but a full login is performed for each request.

Stateful Web Services transactions:

- Do not utilize OM session pooling
- Return a valid JSESSIONID value which is used for authenticating subsequent requests
- An OM session is not shared and can store details about previous requests, this is helpful when performing pagination over a query result set
- The Concurrent Session Limit (Company) limiter controls the number of sessions that can be established for a company.

5 Best Practices for Designing Client Applications

This chapter provides best practice recommendations that allow you to design client applications that interface optimally with Oracle CRM On Demand using Web Services On Demand. It contains the following topics:

Best Practices for Integration Design

This topic describes best practices for optimizing the design of client applications.

Sending of Web Services Requests to Oracle CRM On Demand

Oracle CRM On Demand processes Web services requests in a synchronous manner, therefore client applications using a single session must send requests in a synchronous manner. If the client application needs to send messages asynchronously, multiple sessions must be used.

Flushing of Caches

In Oracle CRM On Demand, there are internal caches that store metadata information such as field customization data, access or privilege settings, book information and so on.

For stateless Web services only, users can flush the caches to reload any metadata information that has changed recently. You use the MetadataChangeSummaryQueryPage method to determine whether there have been any metadata changes; see "MetadataChangeSummaryQueryPage" on page 175.

In the case of session-based Web services (stateful Web Services), the same flushing of caches would only occur on logging in again.

Best Practices for Flushing of Caches

The best practices for flushing of the caches are as follows:

- You are advised not to flush caches unless really necessary, because it affects throughput.
- If you detect stale metadata, you use a flush cache flag to reload the caches.

You can use the MetadataChangeSummaryQueryPage method to check whether the metadata has changed.

Usage

To flush the caches, you include the SOAP header element <FlushCache> in requests. The element can contain the values true or 1 to indicate that caches are to be flushed. Any other value is considered as false.

The following shows a SOAP message containing the <FlushCache> element:

```
<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">

<soap:Header>

<wsse:Security>

<wsse:UsernameToken>

   <wsse:Username>user@ondemand.com</wsse:Username>

   <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">password</wsse:Password>

</wsse:UsernameToken>

</wsse:Security>

<FlushCache xmlns=”urn:crmondemand/ws”>true</FlushCache>

</soap:Header>

<soap:Body>

... Input request or payload here ......

</soap:Body>

</soap:Envelope>
```

Using Stateless Web Services

Stateless Web services use server resources more efficiently and can reduce the cost of implementation as customers do not need to implement session pooling algorithms in their client applications. The best practices for using stateless Web services are described in the following sections.

The best practices for stateless Web services are described in the following sections.

NOTE: The best practices for stateful Web services operations must also be applied to the use of stateless Web services; see "Session Management and Pooling" on page 77.

Using Both Stateful and Stateless Web Services

Customers may decide to implement both stateful and stateless Web services originating either within the same client application, or they may decide to implement one application that is stateless and another that is stateful. While Oracle CRM On Demand supports the use of both login mechanisms, it is recommended that stateful and stateless Web services calls are not mixed.

For example, if you follow this sequence:

1. Perform a stateful web service login and store the JSESSIONID value
2. Issue a stateless Web service request while providing the JSESSIONID from the stateful request.

This sequence results in the stateless Web service call using the session associated with the JSESSIONID and will therefore not be a stateless request. The JSESSIONID will not be invalidated after processing the request, and the user must perform a log off for the session because it was created through an implicit login.

When issuing a stateless request, a client can keep the resulting Java session alive using the <SessionKeepAlive> parameter. This results in a session ID being returned to the client application. To maintain server affinity, the session ID returned by the initial request must be returned in subsequent requests. For many development platforms this is the default behavior, that is, the cookie is returned with the next request.

If the <SessionKeepAlive> parameter is not set to true, the session ID does not identify a specific session as it does for stateful requests, however it does identify the specific server on which the session resides. Including this session ID for subsequent requests ensures that the request is routed to the correct server (not doing so may result in the subsequent request being routed to another server and the establishment of a new session on that server).

When the <SessionKeepAlive> parameter is set to true, the session ID value functions similarly to stateful Web services and allows a subsequent request to be handled by the session associated to the session ID value. Note that there is no logoff command for stateless Web service even if the <SessionKeepAlive> parameter is set to true; Oracle CRM On Demand, server-side session management logs off sessions as required to ensure equitable allocation of resources.

Avoid Multiple Concurrent Requests

Avoid issuing multiple concurrent requests for the same application unless absolutely required. If you do send multiple concurrent requests, and if you receive a RIP_WAIT error, or server unavailable error, the server might be busy due to the number of concurrent requests it is handling. If this happens, do one of the following:

- If possible, try the request again later, as the load on the server may decrease.

  Most load conditions are rare and temporary. You may never see one during development.

- Retry with an exponential backoff. The client can be implemented such that retries are issued automatically using an exponential retry rate (that is, retries occur after 100ms, 200ms, 400ms, 1600ms and so on).

Use Sort Criteria when Using QueryPage

If the QueryPage method is used as a stateless transaction, each request for an additional page of data returns any records that have been added or updated since the initial query. Any records that have been deleted since the first request will no longer appear in the result set.

It is recommended to use sort criteria to reduce the possibility of returning the same record when paging through results using the QueryPage method:

- Use sort criteria on the Id field, which helps in most simple cases.
- Use sort criteria on a field that is being filtered to help improve performance.
- Use a stateful QueryPage Web service request, if it is required to paginate through a snapshot of data.

NOTE: It is however not recommended to use multiple sort criteria in a QueryPage request as it diminishes the performance of the query.

Setting and Querying Blank Values

When updating or querying for blank values the best practice is to specify isNull instead of leaving the value blank.

As an example, the AccountName and Location fields form a user key for updating or querying Account records. Location is not a required field, therefore a null or blank value can be set for this field. The best practice is therefore to specify isNull for Location instead of blank.

Working with Opportunity Product Revenue Records

When implementing a client application that inserts or both inserts and updates Product Revenue records associated with an Opportunity, it is important to ensure that the Revenue record is associated not only with the Opportunity but also with a Product record. This is because Opportunity Revenue records that do not have an associated Product will not appear within the Oracle CRM On Demand UI. These records will appear in Forecast and Opportunity revenue roll-up but are not editable using the UI or Web services.

Error Handling and Logging

Error handling and logging are essential when developing a client application. The client application must provide for:

- Logging of detailed information about the error observed.

- Logging of the body and header information of all SOAP requests and responses. For the resolution of some errors, the actual SOAP request can be extremely useful in identifying the root cause of a problem.
- A call stack, which can be extremely important when analyzing problems and can provide useful hints that may reveal contributing factors to the problem.
- Entry points wrapped in log messages. The ability to identify entry and exit of Web service calls is important when analyzing issues.
- If a Web service request returns an error, the ability to analyze the result, stop immediately, or continue depending on the severity of the issue reported.
- End points that are not hard-coded.
- Dynamic server name and protocol configuration

Handling Outages and Failures

The client application must contain a mechanism to recognize when the Oracle CRM On Demand application is not available, and be able to persist in a dormant state. This mechanism can either be achieved manually or programmatically; for example:

- A process can become dormant if it receives a HTTP 404 error message and retry after several minutes.
- A process can alert an administrator and shut down after *x* failed attempts.

A situation may arise where it is unknown if an operation has succeeded or not. In this situation, if the client application can detect duplicate errors, you can retry an insert operation with Oracle CRM On Demand user keys allowing you to uniquely identify records. You can identify lost updates by examining modification dates on records.

Best Practices for Generating Web Services Proxy Classes in Java Environments

This topic provides information on commonly encountered issues while generating Web services proxy classes in the Java integrated development environments (IDE) Oracle JDeveloper and Axis. Solutions and workarounds are provided to successfully generate the Web Services v2.0 API proxy classes in these Java IDEs.

Oracle JDeveloper Workaround

For WSDLS that have a large number of fields, proxy generation succeeds for all WSDL files, but compilation may fail due to the doSerialize() or doDeserialize() method being more than 64KB in size.

To work around the failure

1 See "Avoiding Proxy Class Compilation Failure Due to the Java 64KB Limit" on page 70

2 Increase the Java heap size in the jdev.conf file.

Axis Workaround

Proxy generation can fail for large WSDL files, and throw the following exception:

```
java.lang.OutOfMemoryError: Java heap space
```

To work around the failure

1 Remove unused child objects from WSDL files

2 Increase the Java heap size in the wsdl2java.bat/wsdl2java.sh file.

To remove child objects from Web Services v2.0 WSDL files

1 In the <Types> section in the WSDL, find the <xsd:schema> element with the target namespace "urn:/crmondemand/xml/*ParentName*/Query".

   a Traverse to the <xsd:complexType> element with the name "*ParentName*Query".

   b Remove the <xsd:element> element with the name "ListOf*ChildName*Query".

   c Remove the <xsd:complexType> element with the name "ListOf*ChildName*Query".

   d Remove the <xsd:complexType> element with the name "*ChildName*Query".

2 Find the <xsd:schema> element with the target namespace "urn:/crmondemand/xml/*ParentName*/Data".

   a Traverse to the <xsd:complexType> element with the name "*ParentName*Data".

   b Remove the <xsd:element> element with the name "ListOf*ChildName*Data".

   c Remove the <xsd:complexType> element with the name "ListOf*ChildName*Data".

   d Remove the <xsd:complexType> element with the name "*ChildName*Data".

Avoiding Proxy Class Compilation Failure Due to the Java 64KB Limit

The Java language enforces a size limit on member functions, which cannot exceed the size of 64KB (see http://java.sun.com/docs/books/jvms/second_edition/html/ClassFile.doc.html#9279).

As part of the process of generating proxy classes from the Oracle CRM On Demand WSDL files, the serializer and deserializer methods that are generated can exceed 64KB in size. Therefore, a process like the following can result in a compilation error:

1 Generate proxy classes using Oracle JDeveloper 11g or any other Java integrated development environment (IDE).

2 Compile the proxy classes.

An out of memory error message is thrown due to the 64KB limit on member functions. For example, the error message for Oracle JDeveloper 11g is:

```
code segment of method doDeserialize(oracle.j2ee.ws.common.streaming.XMLReader,
oracle.j2ee.ws.common.encoding.SOAPDeserializationContext) too large
```

This occurs for any Java compiler depending on the number of fields that are generated in the doSerialize or doDeserialize methods, as the 64KB limit is a Java VM limitation.

There are two possible workarounds:

- Split the doSerialize method.

  You can split the doDeserialize () and doSerialize () methods into multiple smaller methods to bypass the 64KB limit. The sample code snippets in Figure 2 and Figure 3 illustrate the splitting of the doSerialize method.

  TIP: This is the recommended approach, as it is more consistent and is a more generic solution that can be implemented across all client integrations.

- Delete unused fields generated in the serialize methods.

  You can trim the doSerialize method by deleting unused fields, which is specific to each client integration.

Figure 2. A doSerialize() Method Before Splitting:

```
public void doSerialize(java.lang.Object obj, XMLWriter writer,    SOAPSerializationContext context) throws
Exception

{

// @GeneratedBlockBegin (value={"oracle.j2ee.ws"} );

              activity.proxy.types.crmondemand.xml.activity.data.ActivityData instance =

        (activity.proxy.types.crmondemand.xml.activity.data.ActivityData)obj;


              if (instance.getModifiedDate() != null) {

                  myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializer.setNullable( false );

myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializer.serialize(instance.getModifiedDate(),

          ns2_ModifiedDate_QNAME, null, writer, context);

              }

              if (instance.getCreatedDate() != null) {

                  myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializer.setNullable( false );

myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializer.serialize(instance.getCreatedDate(),

                  ns2_CreatedDate_QNAME, null, writer, context);

              }

              if (instance.getModifiedById() != null) {

                  myns3_string__java_lang_String_String_Serializer.setNullable( false );
```

```
                 myns3_string__java_lang_String_String_Serializer.serialize(instance.getModifiedById(),
                 ns2_ModifiedById_QNAME, null, writer, context);
             }
             if (instance.getCreatedById() != null) {
                 myns3_string__java_lang_String_String_Serializer.setNullable( false );
                 myns3_string__java_lang_String_String_Serializer.serialize(instance.getCreatedById(),
                 ns2_CreatedById_QNAME, null, writer, context);
             }
             if (instance.getModId() != null) {
                 myns3__int__java_lang_Integer_Int_Serializer.setNullable( false );
                 myns3__int__java_lang_Integer_Int_Serializer.serialize(instance.getModId(),
                 ns2_ModId_QNAME, null, writer, context);
             }
             ...
}
```

Figure 3. A doSerialize() Method After Splitting

```
public void doSerialize(java.lang.Object obj, XMLWriter writer, SOAPSerializationContext context) throws
Exception
    {
   // @GeneratedBlockBegin (value={"oracle.j2ee.ws"} );
        activity.proxy.types.crmondemand.xml.activity.data.ActivityData instance =
(activity.proxy.types.crmondemand.xml.activity.data.ActivityData)obj;


   instance = doSerialize1(instance, writer, context);
   instance = doSerialize2(instance, writer, context);
   ...
    }


public activity.proxy.types.crmondemand.xml.activity.data.ActivityData
doSerialize1(activity.proxy.types.crmondemand.xml.activity.data.ActivityData instance, XMLWriter writer,
SOAPSerializationContext context) throws Exception
    {
   if (instance.getModifiedDate() != null)
      {
    myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializer.setNullable( false );
    myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializer.serialize(instance.getModifiedDate(),
         ns2_ModifiedDate_QNAME, null, writer, context);
      }
      if (instance.getCreatedDate() != null)
```

```
      {
   myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializer.setNullable( false );
   myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializer.serialize(instance.getCreatedDate(),
        ns2_CreatedDate_QNAME, null, writer, context);
      }
      ...
return instance;
    }
public activity.proxy.types.crmondemand.xml.activity.data.ActivityData
doSerialize2(activity.proxy.types.crmondemand.xml.activity.data.ActivityData instance, XMLWriter writer,
SOAPSerializationContext context) throws Exception
    {
   if (instance.getModifiedById() != null)
      {
    myns3_string__java_lang_String_String_Serializer.setNullable( false );
    myns3_string__java_lang_String_String_Serializer.serialize(instance.getModifiedById(),
         ns2_ModifiedById_QNAME, null, writer, context);
      }
      if (instance.getCreatedById() != null) {
    myns3_string__java_lang_String_String_Serializer.setNullable( false );
    myns3_string__java_lang_String_String_Serializer.serialize(instance.getCreatedById(),
         ns2_CreatedById_QNAME, null, writer, context);
      }
      if (instance.getModId() != null) {
    myns3__int__java_lang_Integer_Int_Serializer.setNullable( false );
    myns3__int__java_lang_Integer_Int_Serializer.serialize(instance.getModId(),
         ns2_ModId_QNAME, null, writer, context);
      }
     ...
return instance;
    }
```

Entering Telephone Number Values through Web Services

Telephone number values entered through Web service requests and the Oracle CRM On Demand UI are displayed differently both in the UI and in the response to the QueryPage operation. Table 16 shows how telephone number values are displayed in the UI or within the response to a QueryPage request.

Table 16. Formatting of telephone number values entered through Web services or the UI

Input Mechanism	Formatting in UI	Formatting in Web Service Response
Input through Web service		
14042621601123	1 (140) 426-2160 #1123	+1 404 2621601 #123
14042621601	1 (140) 426-2160 #1	+1 140 4262160 #1
+14042621601,123	1 (4042621601) 123	+1 4042621601 123
Input through UI		
14042621601123	14042621601123	+1 4042621601123
14042621601	1 (404) 262-1601	+1 404 2621601
+14042621601,123	1 (404) 262-1601 #123	+1 404 2621601 #123

Telephone number values differ from other values in that the format of the value impacts the accuracy of the value. When querying for a telephone number value, it is necessary to return the value with formatting.

Recommended Formats for Telephone Numbers

When determining the value to be input through Web Services you must take into account how the telephone number will appear in the UI and in the Web service response.

The following formats are recommended for telephone numbers entered through the UI or Web services, as they are formatted the same way in the UI and in Web service responses:

1 (404) 262-1601 123

1 (404) 262-1601 #123

1 (140) 426-2160 #1

1 (140) 426-2160

The following is an example of a request that uses one of the above formats to insert a Contact with telephone number fields.

```
<ListOfContact>
```

```
   <Contact>
      <ContactFirstName>Contact</ContactFirstName>
      <ContactLastName>Name</ContactLastName>
      <WorkPhone>1 (404) 262-1601 #123</WorkPhone>
      <PHONE_000>1 (404) 262-1601 #123</PHONE_000>
   </Contact>
</ListOfContact>
```

The following is the query response:

```
<ListOfContact xmlns="urn:/crmondemand/xml/Contact/Data" lastpage="true">
   <Contact>
      <ContactFirstName>qq2</ContactFirstName>
      <ContactLastName>ww2</ContactLastName>
      <WorkPhone>1 (404) 262-1601 #123</WorkPhone>
      <PHONE_000>1 (404) 262-1601 #123</PHONE_000>
   </Contact>
</ListOfContact>
```

Other Considerations for Telephone Numbers

If you use the + character and parentheses () characters together in a Web services request, the number is displayed differently in the UI and Web service response. For example, the input:

```
+1 (404) 262-1601 #123
```

is displayed as the following in the Web service response:

```
+1 404 2621601 #123
```

and as the following in the UI:

```
1 (404) 262-1601 #123
```

It is not possible to modify the telephone number in the UI to display a + prefix for a telephone number, or to remove the parentheses () from a telephone number using Web services. For example, the input:

```
+1 404 262-1601 #123
```

is displayed as the following in the Web service response:

```
1 404 2621601 #123
```

and as the following in the UI:

```
1 (404) 262-1601 #123
```

Telephone numbers submitted through Web services requests must begin with a numeric value. For example, updating a telephone number field with the value `DO NOT CALL` would result in an error being returned to the client. However, specifying the value `1DONOTCALL` would succeed.

Best Practices for Adhering to Web Service Allotments

This topic lists several best practices to help ensure that you do not exceed the various Web services allotment values.

For the Web Services Operations allotment:

- Avoid unnecessary Web services requests.
  - Avoid issuing the same query multiple times, if you do not expect the set of records returned to change; cache the data locally to avoid repeated requests for the same information.
  - Issue subsequent page requests when necessary, do not retrieve the entire record set if it is not necessary.
- Ensure high data quality to minimize errors (this is especially important for batch requests):
  - Ensure field values are of the correct type
  - Include values for required fields in all requests
  - Do not use invalid Id values when associating records using Web services.
- Use the Oracle Data Loader On Demand tool or the Import Assistant for loading large amounts of data into Oracle CRM On Demand.
- Use the Export Assistant for extracting large amounts of data from Oracle CRM On Demand.
- Include the miniumum set of related objects in QueryPage requests.

For the Web Services Concurrent Request allotment:

- Schedule automated or low priority clients to execute during off-peak hours.
- When performing more than one operation based on a user's input, send requests sequentially, not in parallel.
- Use the Execute method in the Web Services v2.0 API to bundle multiple requests on a set of records of the same type in a single Web service request.

Best Practices for Integration Performance

This topic describes best practices for optimizing the performance of client applications.

Batch Processing

With Web Services On Demand, you can perform batch operations that optimize performance by combining multiple requests into one.

Oracle CRM On Demand batch processing has a limit of 20 top-level records for each request and is supported for the following operations:

- Insert
- Delete
- Update
- InsertOrUpdate (Web Services v1.0 only)
- QueryPage

Because batch calls take longer to process than single operations they must only be used in instances where longer response time would not impact the user experience. However, for such interactive applications, if Oracle CRM On Demand needs to process multiple records of the same type, batch operations increase the performance.

If a single record in a batch causes an error, the entire batch is not processed. For example, a batch of 20 Account inserts where one record contains an error will require all records to be re-inserted.

NOTE: When performing batch operations using Web services, workflow actions are triggered as each operation in the batch is performed. If an error is encountered in the submitted batch of Web service operations, workflow actions that have been triggered before the error is encountered (including integration event generation) cannot be rolled back.

A batch error could result from a data error or other error (for example, network outage, session expiry, and so on). If the error is not data-related, it is recommended that the user logs in again and tries the Web service call again. If the error is data-related, the batch can be split into smaller batches so that the records that do not cause errors can be processed.

Session Management and Pooling

It is recommended that you use stateless Web services as opposed to stateful Web services whenever possible, as described in "Using Stateless Web Services" on page 66. This topic however discusses the best practices for the situations when you use stateful Web services.

For stateful Web services requests, Web Services On Demand uses a session-based security mechanism for which each operation is synchronous.

It is recommended that a user:

- Always closes sessions if the application process is not likely to be used multiple times within the session idle time-out period (10 minutes by default).
- Always keeps sessions open and reuses them when the application process is likely to be used multiple times within the session idle time-out period. It is important to reuse sessions that are not in use, as frequent logins add overhead to your process and slow it down.

Client applications must not reuse sessions that are in use, in other words, they must not submit several simultaneous requests using the same session.

Client applications must not send multiple requests simultaneously using the same session ID, rather, the client must wait for a response before sending a new request using the same session ID.

The client time-out on a single Web service call must be set to at least 10 minutes, so that the client does not time out when a request is still pending.

For information about Web services sessions, see "About Establishing and Managing the Web Services Session" on page 53.

Session Pooling

Session pooling is another option for increasing the performance of your application further. Session pooling involves maintaining a list of active sessions on the client application. The client application must ensure that each session is active and valid (it must have a valid session ID) before using it in a request. The client application might determine whether the session is active based on the success of the login operation and the time that has passed since the session was used. If all active sessions are in use for pending Web service requests, add a new session to the pool.

You can use session pooling to improve performance in both a single-threaded or multi-threaded application. In a single-threaded application, session pooling can avoid the unnecessary overhead of re-logging into Oracle CRM On Demand for each request. In a multi-threaded application session, you can use session pooling to run multiple requests at the same time.

API Calls

Whenever possible, it is recommended that queries be as specific as possible to reduce the number of records in the result set. You must restrict the fields returned by queries to only the fields that are required by your process.

Queries that involve related child objects (that is child objects that are top-level objects), or complex queries that involve criteria from both parent and related child objects, may perform better if they are separated into multiple requests.

The following are also recommended:

- For Web Services v1.0 calls, use the child methods DeleteChild, InsertChild, and UpdateChild for child delete, insert, and update operations.
- Whenever possible, store your company's unique identifiers in the external system ID field on objects.

Performance of the Update and Insert Methods Versus the InsertOrUpdate Method

If you use Web Services v1.0, designing your application to use the Insert and Update methods may result in an increase in throughput compared to using the InsertOrUpdate method. This is due to the additional business logic and SQL statements executed during the InsertOrUpdate operation to determine whether the submitted records match any existing records in the Oracle CRM On Demand database. The performance and throughput improvement may not be apparent at lower volumes, but high volume applications could benefit from the direct update and insert operation, which does not require the additional business logic to determine whether the records already exist.

Working with Attachments

For a number of record types, you can include attachments with Web services requests. If you add attachments, remember that:

- Requests with large attachments perform more slowly than those with smaller attachments or no attachments. The maximum attachment size supported is 9MB.
- Requests with many attachments perform more slowly than those with a single attachment or no attachments.

If you are adding the same attachment to multiple records, you can take advantage of a reuse facility for attachments, see "Attaching a File to Multiple Records" on page 589. In this way, you do not include the content for each record in the request, you use the ContentId on one record and reference that ContentId from other records. For more information about using attachments in Web Services On Demand, see Appendix A, "Using Attachments With Web Services On Demand".

Querying Login History and User Usage

When you use the LoginHistoryQueryPage and UserUsageQueryPage methods, the queries must be as specific as possible, that is, you must not query for all records with every request. For example, you can narrow the search results by:

- Querying for records owned by a specific UserID
- Querying for records covering a period of time

For more information, see "LoginHistoryQueryPage" on page 169 and "UserUsageQueryPage" on page 197.

Using the QueryPage Method

The following are best practices for using the QueryPage method and similar methods like MetadataChangeSummaryQueryPage and SalesProcessQueryPage:

- When formulating a query, use indexed fields, which are highlighted in green in the New List page in the Oracle CRM On Demand UI. Indexed fields are optimized for fast retrieval. As an example, use the indexed field ModifiedDateExt rather than the ModifiedDate field in QueryPage operations to provide better performance.

- Specify the best operators to make queries faster. For example, use the equality (=) operator instead of the * wildcard. Other operators may provide some functional flexibility, but can severely impair performance. Therefore, you must only use other operators when absolutely required.
- If filtering on a non-equality operator and not combining with any other filter criteria, change the sort order so that you are sorting on the same field that you are filtering.
- Specify only fields that you are intending to use. Adding all fields or specifying fields that are not required impacts the response time or throughput.
- If filtering on a custom field, make sure that the custom field is an indexed custom field. It may be required to migrate data from the existing field to the indexed custom field. For more information, see "Using Indexed Custom Fields" on page 80.
- Constrain filter criteria to return the least number of records possible (that is, queries must be as specific as possible).
- If you must return many sorted records, make sure that you sort on an indexed field.
- If you are using manager visibility (ViewMode="Manager" or ViewMode="EmployeeManager"), the query must contain an equality operator on an indexed field.
- Avoid unnecessary use of attachments by accessing attachments through the Oracle CRM On Demand UI whenever possible.
- Use the Web Services v2.0 API when querying for associated records, as Web Services v2.0 supports filtering and paging at the child level.
- Use targeted searches and smaller page sizes for QueryPage operations to return less data in each request.

  NOTE: This can result in higher Web Service Operations Allotment usage if the user needs to page through multiple sets of records to find the right record.

For more information about the QueryPage method, see "QueryPage" on page 91, or "QueryPage" on page 115.

Using Indexed Custom Fields

To optimize performance, you can use custom fields that have been indexed for specific record types. Indexed fields are special fields that improve the response time during the search process or sorting on a particular list. Indexed custom fields are preconfigured in the Oracle CRM On Demand database. You can change the labels on the indexed custom fields, but you cannot change the integration tags.

NOTE: As an option, you can choose to migrate your data from nonindexed to indexed custom fields to increase the performance of the Web services queries that your users execute. To migrate the existing data to the available record types, use Oracle CRM On Demand's export and import functionality or Web services. For more information about exporting and importing data, see the online help for Oracle CRM On Demand.

Indexed custom fields are prefixed with Indexed as a default.

Best Practices for Integration Management

This topic describes best practices for system management when using client applications that make Web services calls.

Moving Customers Between Pods

To reduce the impact on customers when they move between pods (Oracle CRM On Demand instances), it is important that the server URL values for pods are parameterized, so that they can be changed easily.

Web services clients must be implemented in such a way that moving a customer to a new pod does not require any code changes within the Web services client. A best practice is to use an .ini file to store the server URL, so that changing the server value in the .ini file results in the Web services request being routed to the new Oracle CRM On Demand instance.

Web links, Web tabs, and custom Web applets that refer to Oracle CRM On Demand must be parameterized so that they can easily be redirected. A best practice is to pass the server URL or POD value (that is, the 3 letter pod identifier) so that it can be parsed from the URL, and the page being called can continue to interact with Oracle CRM On Demand through Web services.

Handling Outages

Outages can be due to:

- **Scheduled Maintenance Downtime**. From time to time, Oracle CRM On Demand will have scheduled downtime when it is shut down to perform regular maintenance and upgrades. It is important for your client applications to be able to identify and respond correctly to this scenario.
- **Application Failures**. If there is a failure within Oracle CRM On Demand, it is important for the client application to respond appropriately. Performing proper error handling and logging is extremely important because it will not only help you resolve issues on your own but, if necessary, help you engage with Oracle CRM On Demand Customer Support and provide them with critical information. For more information, see "Handling Outages and Failures" on page 69.

Maintaining SSL Certificates

Oracle CRM On Demand is accessible only through HTTPS and Oracle servers using Secure Sockets Layer (SSL) certificates issued by common certificate authorities (CA) such as Verisign. Approximately yearly these SSL certificates are renewed, so it is important that any clients (including browsers and integration clients) are configured to trust the root CA certificates and not the specific server certificates.

Occasionally the certificate authority may issue new root certificates or start issuing server SSL certificates to Oracle that use a different root or intermediate certificate. It is the responsibility of customers to ensure that their clients (browser or integration platform) are kept up to date with the latest root certificates from Verisign or others. See, for example, the Verisign Web site for details of how to update your client.

6 Web Services On Demand API Calls

This chapter contains the following topics:

Web Services On Demand API Calls

This methods that the Oracle CRM On Demand Web services can call are:

- The methods of the Web Services v1.0 API that are called on Oracle CRM On Demand objects to insert, update, delete, and find data within a specified Oracle CRM On Demand instance.
- The methods of the Web Services v2.0 API that are called on Oracle CRM On Demand objects to insert, update, delete, and find data within a specified Oracle CRM On Demand instance.
- The service API methods that are used to perform queries and other operations on non-data records through Web services.
- The methods of the Administrative Services API that are used to automate the administration of a company's configurations.
- The methods of the Data Loader API that are used to perform bulk data import operations.

For each of these methods, usage, arguments taken, and return values are detailed.

TIP: Sample XML code for the various APIs may be available in the Oracle Sample Code public repository on the Oracle Technology Network. You must have an account to access the samples. See, for example: https://codesamples.samplecode.oracle.com/servlets/tracking/action/ExecuteQuery?query=crm_on_demand.

Web Services v1.0 API Calls

The Web Services v1.0 methods are listed in Table 17. These methods can be called on most Oracle CRM On Demand objects (record types), for example, Account, Contact, Opportunity, and so on, including Custom Objects 01 through 03, but not Custom Objects from 04 onwards.

NOTE: The actual method names consist of the object name prefix and Delete, Insert, and so on, for example, AccountDelete and AccountInsert are methods of the Account service.

Table 17. Web Services v1.0 Methods

Method	Comments
Delete	Finds records in the Oracle CRM On Demand database that match specified field values, and then deletes them (in other words, puts them into the Deleted Items area).
DeleteChild	Deletes child records from the Oracle CRM On Demand database, or removes the association between the child and the parent object.
Insert	Inserts new records into the Oracle CRM On Demand database.
InsertChild	Inserts new child records into the Oracle CRM On Demand database.
InsertOrUpdate	Updates existing records or inserts a new record if one did not exist.
QueryPage	Executes a query against a specified list of records, and returns a subset of the records that match the search criteria set by the method arguments.
Update	Updates records with a new value.
UpdateChild	Updates child records with a new value.

For each object, the methods are defined in the WSDL file for that object. Many of the methods described in this chapter can be called on all of the objects.

Before processing requests for the Insert and InsertOrUpdate methods, Oracle CRM On Demand checks whether the Record allotment for the company has been reached. If the allotment has been reached, the request is not processed.

Depending on whether an object is a parent or child object, Web services methods can act in different ways on the object in question. These differences are described in the following topics.

User Keys

The objects (both child and parent level) provided in the input arguments for the Web Services v1.0 methods must provide data in at least one of the user keys for the given object. You use the user key information to uniquely identify records. If no user key values are provided, or if there is a conflict with the user keys of an existing record, the method fails, and a SOAP error is thrown by the API.

The Echo Argument

Oracle CRM On Demand Web services using the methods Insert, Update, InsertAndUpdate, Delete, InsertChild, UpdateChild, and DeleteChild can specify an Echo input argument. The Echo string is used only for Integration events and is not required. Echo is case-sensitive and controls whether data sent to Oracle CRM On Demand through integration Web services are recorded as transactions. The default value is On. When the Echo value is On or missing, the transaction is recorded. When the Echo value is Off, the transaction is not recorded.

NOTE: For Java users, the Echo string is required for all input methods. The echo string can be set to Off.

Delete

Removes records of a specified record type from the Oracle CRM On Demand database.

Usage

You use the Delete method to remove one or more records of a particular object from an Oracle CRM On Demand instance.

The deleted records appear in the Deleted Items folder and can be restored through the Oracle CRM On Demand UI.

NOTE: To conform with Oracle CRM On Demand’s business logic, be careful about the order in which objects are deleted. You cannot delete some objects unless some action is performed on its child objects. For example, you cannot delete an account unless you re-associate all its service requests with a different account. For information about the behavior of the Delete method on child objects, see Table 18.

Table 18 illustrates the behavior of the Delete method on child objects that are related to the parent object being deleted. For more information about deleting records, see the online help for Oracle CRM On Demand.

NOTE: If you update an object, and the child is not in the input, that child is deleted from Oracle CRM On Demand. For more information, see Table 29.

Table 18. Behavior of Delete Method on Child Objects

Parent Object	Child	Action When Parent Is Deleted
Account	Activity	Delete
	Asset	Delete
	Competitor	None
	Contact	None
	Lead	Delete
	Note	Delete
	Opportunity	Delete
	Partner	None
	ServiceRequest	None
	Team	Delete
Activity	Attachment	Delete
Campaign	Activity	Delete
	Contact	None
	Lead	Delete
	Note	Delete
	Opportunity	None
Contact	Account	None
	Activity	Delete
	Asset	None
	Campaign	None
	Interests	Delete
	Lead	Delete
	Note	Delete
	Opportunity	None
	ServiceRequest	None

Table 18. Behavior of Delete Method on Child Objects

Parent Object	Child	Action When Parent Is Deleted
Household	HouseholdTeam	None
Lead	Activity	Delete
MedEd	Invitees	Delete
Opportunity	Activity	Delete
	Competitor	None
	Contact	None
	Lead	Delete
	Note	Delete
	OpportunityTeam	None
	Partner	None
ServiceRequest	Activity	Delete
	AuditTrail	None
	Note	None
	Solution	Not Specified
Solution	ServiceRequest	None

Arguments

Table 19 describes the arguments taken by the Delete method.

Table 19. Arguments Taken by the Delete Method

Name	Description	Required	Default	I/O
ListOf*(Object)*. For example, ListOfAccount	The list of object instances to be deleted.	Yes	Not applicable	Input/ Output
Echo	Controls whether data sent to Oracle CRM On Demand through integration Web services are recorded as transactions.	No	On	Input

Return Value of the Call

The status key for each of the deleted objects.

Related Topic

Update.

DeleteChild

Removes child records from the Oracle CRM On Demand database.

Usage

You use the DeleteChild method to remove one or more child records of a particular object from an Oracle CRM On Demand instance, or remove the association between the child and parent object. Table 18 on page 86 is also applicable for the DeleteChild method.

The deletion of child records or removal of association follows the same pattern as for deletion in the UI of the Oracle CRM On Demand application. For example, if you use AccountDeleteChild on a Contact child record, the association is removed, but the Contact is not deleted. On the other hand, if you use AccountDeleteChild on a Team child record, that record is deleted. However, the integration events generated in the UI and from Web services requests differ for child objects of Account, Contact, and Opportunity. For more information about these differences in integration events and about deleting records, see the information about workflow rules in the online help for Oracle CRM On Demand.

CAUTION: The parent object may be deleted by the DeleteChild method in some cases when a child object is not specified when executing the different DeleteChild methods. Nodes with at least one child are called internal nodes and nodes without children are called leaf nodes. DeleteChild operates on leaf nodes, so that if the request specifies a parent that has no children, the parent is deleted.You can avoid this situation by calling the Update method on the parent with an empty container for the children.

Arguments

Table 20 describes the arguments taken by the DeleteChild method.

Table 20. Arguments Taken by the DeleteChild Method

Name	Description	Required	Default	I/O
ListOf*(Object)*. For example, ListOfAccount	The list of child object instances to be deleted. Each child object has an associated parent object.	Yes	Not applicable	Input/ Output
Echo	Controls whether data sent to Oracle CRM On Demand through integration Web services are recorded as transactions.	No	On	Input

Return Value of the Call

The status key for each of the deleted child records.

Insert

Inserts new records in the Oracle CRM On Demand database.

Usage

You use the Insert method to create one or more records of a particular object in an Oracle CRM On Demand instance.

When inserting a batch of records, the batch is treated as a single transaction. If one record fails to insert during a batch insertion, the entire operation is rolled back and no records are inserted.

Table 21 outlines how the Insert method acts on parent and child objects to create or update an object instance.

Table 21. Effect of Insert on Parent and Child Objects

Method	New Parent	New Child	Existing Parent	Existing Child Record
Insert	New parent instance	New child instance	Error	Associate to new parent instance

Arguments

Table 22 describes the arguments taken by the Insert method.

Table 22. Arguments Taken by the Insert Method

Name	Description	Required	Default	I/O
ListOf*(Object)*. For example, ListOfAccount	The list of object instances to be inserted.	Yes	Not applicable	Input/ Output
Echo	Controls whether data sent to Oracle CRM On Demand through integration Web services are recorded as transactions.	No	On	Input

Return Value of the Call

The status key for each of the Oracle CRM On Demand objects.

Related Topic

Update.

InsertChild

Inserts new child records in the Oracle CRM On Demand database.

Usage

You use the InsertChild method to create one or more child records of a particular object in an Oracle CRM On Demand instance.

Arguments

Table 23 describes the arguments taken by the InsertChild method.

Table 23. Arguments Taken by the InsertChild Method

Name	Description	Required	Default	I/O
ListOf*(Object)*. For example, ListOfAccount	The list of child object instances to be inserted. Each child object has an associated parent object	Yes	Not applicable	Input/ Output
Echo	Controls whether data sent to Oracle CRM On Demand through integration Web services are recorded as transactions.	No	On	Input

Return Value of the Call

The status key for each of the inserted child records.

InsertOrUpdate

Updates existing records or inserts a new record if one did not exist for an instance of the object.

Usage

You use the InsertOrUpdate method to update one or more records of a particular object in an Oracle CRM On Demand instance. Use the user key specified for the parent level objects in the input argument to determine whether to insert each of the parent records, or to update an existing parent record.

Table 24 outlines how the InsertOrUpdate method acts on parent and child objects to create or update an object instance.

Table 24. Effect of InsetOrUpdate on Parent and Child Objects

Method	New Parent	New Child	Existing Parent	Existing Child Record
InsertOrUpdate	New parent	New child	Update parent	Update child

Arguments

Table 25 describes the arguments taken by the InsertOrUpdate method.

Table 25. Arguments Taken by the InsertOrUpdate Method

Name	Description	Required	Default	I/O
ListOf*(Object).* For example, ListOfAccount	The object instances to be inserted or updated.	Yes	Not available	Input/ Output
Echo	Controls whether data sent to Oracle CRM On Demand through integration Web services are recorded as transactions.	No	On	Input

Return Value of the Call

The status key for each of the inserted or updated objects.

Related Topic

Update, Insert

QueryPage

Executes a query against the set of records for an object, and returns the subset of the records that match the search criteria set by the method arguments.

Query by Template

To improve performance, the QueryPage result contains only those fields and objects that are included in the QueryPage request. To retrieve the values of fields that are not a part of the search criteria, the field must be included in the search request with a blank value.

Query by Children

For all Oracle CRM On Demand object methods (except Attachment), it is possible to query using one operation within a parent-child relationship. This type of query is called Query By Children. The query can be assembled using parent attributes as well as child attributes. You can query for all children of a particular parent or set of parents, all parents of a particular child or set of children, or for both parents and children of a particular set.

NOTE: When performing a query using Query By Children, certain fields on a child object may not be filterable.

Not all child objects are filterable in Web Services v1.0 (for example, the Partner child of Opportunity).

CAUTION: This type of query performs slowly and must only be used when necessary.

As an example of how you can search for parent records using criteria on a child:

```
<UseChildAnd>true</UseChildAnd>

<ListOfContact>

   <Contact>

      <ContactFirstName>='John'</ContactFirstName>

      <ContactLastName/>

      <ListOfAddress>

         <Address>

            <Country>='Canada'</Country>

         <Address>

      </ListOfAddress>

   </Contact>

</ListOfContact>
```

This query would return all contacts where the first name is John and at least one of John's addresses is for Canada. The child criteria are appended to the parent using an AND or OR operator based on the value of the argument UseChildAnd. When UseChildAnd is true, the child criteria are "ANDed", otherwise they are "ORed".

Using the PageSize and StartRowNum Arguments

The PageSize argument, which has a maximum value of 100, is used to specify the maximum number of records to be returned in a QueryPage response.

The pagesize and startrownum arguments are specified in requests as follows:

```
<StartRowNum>0</StartRowNum>

<PageSize>20</PageSize>
```

```
<ListOfAccount>
```

Depending on the value of PageSize, records are returned as follows:

- If the number of records in the record set is less than the PageSize value, the full record set is returned, and the LastPage argument is set to true.
- If the number of records in the record set exceeds the PageSize value, only the number of records specified by the PageSize parameter is returned, and the LastPage value is false.
- For a query whose record set exceeds the PageSize value, incrementing the StartRowNum argument by PageSize returns the next PageSize number of records. For example, to return a record set with 997 records you can use a PageSize of 100 as follows:

PageSize	StartRowNum	Records Returned	LastPage
100	0	0-99	false
100	100	100-199	false
100	200	200-299	false
100	300	300-399	false
100	400	400-499	false
100	500	500-599	false
100	600	600-699	false
100	700	700-799	false
100	800	800-899	false
100	900	900-997	true

- If the size of the record set is greater than PageSize, and this is a subsequent query where there are less than PageSize number of records remaining to be returned, all of the remaining records are returned and the LastPage attribute has a value of true.

Even though the QueryPage method returns a limited number of records, the results are cached, and can then be retrieved by calling the QueryPage method with a value of PageSize+1 for the StartRowNum argument. To retrieve all records in the result set, StartRowNum must be incremented by PageSize for each subsequent QueryPage request until a QueryPage response is returned with LastPage equal to true.

About Oracle CRM On Demand Query Syntax

The query syntax supports only a small subset of binary and unary operators. No Siebel Query Language constructs or functions are supported. The query syntax is summarized in Table 26.

Table 26. Query Syntax for QueryPage

Syntax Type	Notes
`expression`	
{*Operator*} {*Value*}	*Operator* can be binary or unary. The {*Value*} need only be specified for binary operators. Every expression must start with an operator to avoid ambiguity. There is no default operator.
`(`*`expression`*`)` *`conjunction`* `(`*`expression`*`)`	A conjugated expression must be enclosed in parentheses to avoid ambiguity. However, nonconjugated expressions must not be enclosed in parentheses.
`conjunction`	
`OR`	None
`AND`	None
`unary operator`	None
IS NULL	Used to find a match for a value that has no value
`binary operator`	
`=`	None
`~=`	Denotes a case-insensitive exact search (no wildcards used)
`<`	Must be specified as < to ensure well-formed XML.
`<=`	Must be specified as <=
`>`	None
`>=`	None
`<>`	Must be specified as <>
`LIKE`	Wildcard characters are treated as such only in the context of the operator LIKE.
`~LIKE`	Denotes a case-insensitive wildcard search

Table 26. Query Syntax for QueryPage

Syntax Type	Notes
value	
'*literal*'	Literal data is always enclosed in single quotes. To use a single quote within a literal, place another single quote immediately beside that quote. In this way, the query recognizes the quote as a literal and not as an operator. For example, the string ab'c is specified as ab''c. To use the special characters such as asterisk (*), question mark (?), and backslash (\) in queries, preceded them with the \ character. For example, to use the ? wildcard operator in a query, precede it with the \ character as follows: \?

CAUTION: Queries that are case-insensitive or with leading wildcard characters will perform slowly and must only be used when necessary.

Querying Oracle CRM On Demand Data Using Web Services

The QueryPage functions require a list of object instances as input to perform a query. This input argument is called ListOf(*Object*). For example, the ContactQueryPage method requires the ListOfContact argument. Each ListOf(*Object*) argument requires at least one instance of the Object to specify a valid query.

To query an object by a certain field, specify the expression that corresponds to the desired result. The examples in this topic cover the corner cases of quote and wildcard escaping. Assume that a table in the Oracle CRM On Demand database contains the following values for a particular column that is being queried:

```
?abc
abcd
'abc'
= 'abc'
abc?d
abc*d
aBc*D
abcd
abc*d
abc\d
abc\*d
abc\\*d
abc\d
abc\*'d
abc\?"d
```

```
abc\*"d
abc\*'"d
(NULL value)
```

Table 27 specifies the returned record sets for various values of each field value that maps to the preceding list.

Table 27. Returned Record Sets

Field Value	Returned Record Set	Comments
abc	Not applicable	An unquoted value without an explicit operator is invalid input.
'abc'	Not applicable	A quoted value without an explicit operator is invalid input.
= '''abc	''''abc'	None
= "'abc'"	Not applicable	Double quotes are not allowed by the Oracle CRM On Demand Validator. Consequently, this example returns an error message.
= 'abc	'abc	None
= '= ''abc	'''= 'abc'	None
= '= 'abc''	Not applicable	The caller is responsible for correctly formatting quotes in Query* methods. This example does not have correctly formatted quotes, so it results in an error.
= 'abc?d	'abc?d	None
= 'abc\?d	'abc?d	None
LIKE 'abc\?d	'abc?d	None
LIKE 'abc?d	'abc?d abc*d abc\d	None
~LIKE 'abc?d	'abc?d aBc*D abc*d abc\d	None
= 'abc*d'	abc*d	Any wildcard character that has not been formatted with quotes is treated as if it were formatted with quotes.
= 'abc*d	'abc*d	None
= 'abc*d	'abc*d	None

Table 27. Returned Record Sets

Field Value	Returned Record Set	Comments
LIKE 'abc*d	'abc\d abc*d abc*d abc\d abc*'d abc\?"d abc*"d abc*'"d	None
= 'abc*"d	'abc*"d	None
= 'abc\\?"d	'abc\?"d	None
= 'abc\\\?"d	'abc*"d	None
LIKE 'abc\\?"d	'abc\?"d abc*"d	None
LIKE 'abc\\\?"d	'abc\?"d	None
LIKE 'abc*"d	'abc\?"d abc*"d abc*'"d	None
LIKE 'abc*"d	'abc*"d	None
= 'abc*''d	'abc*'d	None
~ LIKE 'abc*\d	'aBc*D abc*d abcd abc*d	None
LIKE 'abc*\d	'abc*d abcd abc*d	None
(empty field)	Not applicable	An empty field value does not influence the search specification in Query by Template. **NOTE:** A query in which all fields are empty fields returns all records.
IS NULL	(empty field) ((> 'abc*') AND (< 'abcd'))	None
OR (~= 'abc*d')	abc*d aBc*D abc*d	None

Table 27. Returned Record Sets

Field Value	Returned Record Set	Comments
NOT LIKE 'abc?d'	Not applicable	The Oracle CRM On Demand Query Validator does not support the NOT operator, so this query returns an error.
> 'abc' BUT < 'abcd'	Not applicable	BUT is not a valid conjunction. Consequently, this query returns an error.

Querying Multiple Fields

To specify a query on multiple fields, expressions must be provided for each field comprising the search specification. When multiple fields in an object instance have expressions, the QueryPage method result is the intersection of all the expressions, or in other words, all of the expressions are combined using the AND operator. This is outlined in "Example 1: Combining Expressions Using the AND Operator."

Example 1: Combining Expressions Using the AND Operator

The Web service client requires the first name, last name, and job title of all the contacts in Oracle CRM On Demand that have a job title equal to CEO and a last name equal to Doe. The XML representation of the ListOfContact object that must be sent in the ContactQueryPage call is as follows:

```
<ListOfContact>
    <Contact>
        <JobTitle>=‘CEO’</JobTitle>
        <ContactLastName>='Doe'</ContactLastName>
        <ContactFirstName />
    </Contact>
</ListOfContact>
```

Multiple Expressions on a Single Field

If you want to apply multiple expressions to a single field, you can combine each expression using either the AND or the OR operator. The result is either the intersection or the union of the object instances respectively.

NOTE: For multiple expressions on a single field, each expression must be enclosed in brackets.

Example 2: Combining Multiple Expressions Using the AND Operator

The Web service client requires the first name, last name, and job title of all the contacts that have been updated between July 28, 2009 6:30am and July 28, 2009 6:45 am.

Send the following XML representation of the ListOfContact object in the ContactQueryPage call:

```
<ListOfContact>
    <Contact>
        <JobTitle />
        <ContactLastName />
        <ContactFirstName />
        <LastUpdated> (>='07/28/2009 06:30:00') AND (<='07/28/2009 06:45:00')</
        LastUpdated >
    </Contact>
</ListOfContact>
```

Example 3: Combining Multiple Expressions Using the OR Operator

The Web service client requires the first name, last name, and job title of all the contacts in Oracle CRM On Demand that have a last name equal to Doe or Brown.

Send the following XML representation of the ListOfContact object in the ContactQueryPage call:

```
<ListOfContact>
    <Contact>
        <JobTitle />
        <ContactLastName>(='Doe') OR (=’Brown’)</ContactLastName>
        <ContactFirstName />
    </Contact>
</ListOfContact>
```

Specifying Books in Queries

For queries on record types that support books, you can use the BookId or BookName arguments to constrain the query to only the records in a particular book. For more information about the use of books, see the online help for Oracle CRM On Demand.

Arguments

Table 28 describes the arguments taken by the QueryPage method.

Table 28. Arguments Taken by the QueryPage Method

Name	Description	Required	Default	I/O
ListOf*(Object)*. For example, ListOfAccount	The list of object instances queried (input), and after query execution, the list of object instances returned (output).	Yes	Not applicable	Input/ Output
PageSize	The maximum number of records displayed on a page following a query.	No	10	Input
StartRowNum	Indicates the row from which the QueryPage method starts to return records. Use the StartRowNum argument to return a set of records for any given method. For example, if you want to return records 1-100, you set StartRowNum to 0. Then, if you want to return records 101-200, you set StartRowNum to 100, and run the query again. You continue doing this until the last page is returned. In this way, you can return all records for a particular query.	No	0	Input
UseChildAnd	If this argument is set to true, the query result set returns the set of records that satisfy both parent and child search criteria. (That is, the query set returned is the AND combination of parent and child queries.) If this argument is set to false (or not set at all), the query result set returns the set of records that satisfy either the parent or the child search criteria. (That is, the query set returned is the OR combination of parent and child queries.)	No	False	Input
BookId	The book Id.	No	Not applicable	Input
BookName	The book name. This argument is ignored if a value for BookId is supplied.	No	Not applicable	Input
IncludeSubBooks	Whether subbooks are to be included.	No	False	Input

Return Value of the Call

An object or list of objects of the type on which the method was called.

- **LastPage.** A Boolean value that indicates whether or not the last value in the query set has been returned.

Update

Updates records with a new value.

Usage

You use the Update method to update one or more records of a particular object in an Oracle CRM On Demand instance.

NOTE: If the company administrator customizes a record type to add a required field, Oracle CRM On Demand does not check for the required field when existing records are updated. When you update the record without the required field through a Web services request, or merge it with a record that does not have the required field, the record is updated or merged without error. This is the intended behavior; when a field is made required, it is the responsibility of the administrator to update all existing records to populate the required field. On inserting new records however, Oracle CRM On Demand checks for the required field.

Table 29 outlines how the Update method acts on parent and child objects to update an object instance.

Table 29. Effect of Update on Parent and Child Objects

Method	New Parent	New Child	Existing Parent	Existing Child Record	Existing Child Record But Not in Input
Update	Error	New child	Update parent	Update child	Child is removed

Arguments

Table 30 describes the arguments taken by the Update method.

Table 30. Arguments Taken by the Update Method

Name	Description	Required	Default	I/O
ListOf*(Object)*. For example, ListOfAccount	The object instance to be updated.	Yes	Not applicable	Input/ Output
Echo	Controls whether data sent to Oracle CRM On Demand through integration Web services are recorded as transactions.	No	On	Input

Return Value of the Call

The status key for each of the updated records.

UpdateChild

Updates a child record with a given value in the Oracle CRM On Demand database.

Usage

You use the UpdateChild method to update one or more child records of a particular object in an Oracle CRM On Demand instance.

Arguments

Table 31 describes the arguments taken by the UpdateChild method.

Table 31. Arguments Taken by the UpdateChild Method

Name	Description	Required	Default	I/O
ListOf*(Object)*. For example, ListOfAccount	The list of child object instances to be updated. Each child object has an associated parent object.	Yes	Not applicable	Input/ Output
Echo	Controls whether data sent to Oracle CRM On Demand through integration Web services are recorded as transactions.	No	On	Input

Return Value of the Call

The status key for each of the updated child records.

Web Services v2.0 API Calls

The Web Services v2.0 API methods are listed in Table 32. These methods can be called on all Oracle CRM On Demand objects (record types), for example, Account, Contact, Opportunity, and so on, including all Custom Objects.

The objects provided in the input arguments for the methods must provide data in at least one of the user keys for the given object. You use the user key information to uniquely identify records. If no user key values are provided, or if there is a conflict with the user keys of an existing record, the method fails, and a SOAP error is thrown by the API.

If a request supplies an ID value in a reference field, an association is created between two records through, for example, the insert operation. Joined in fields from a referenced object are read-only.

NOTE: The actual method names consist of the object name prefix and Delete, Insert, and so on, for example, AccountDelete and AccountInsert are methods of the Account service.

Table 32. Web Services v2.0 API Methods

Method	Comments
Delete	Finds records in the Oracle CRM On Demand database that match specified field values, and then deletes them (in other words, puts them into the Deleted Items area).
Execute	Executes multiple update, insert, and delete operations on separate records in the Oracle CRM On Demand database.
Insert	Inserts new records into the Oracle CRM On Demand database.
QueryPage	Executes a query against a specified list of records, and returns a subset of the records that match the search criteria set by the method arguments.
Update	Updates records with a new value.

Before processing requests for the Insert method, Oracle CRM On Demand checks whether the Record allotment for the company has been reached. If the allotment has been reached, the request is not processed.

For each object, the methods are defined in the WSDL file for that object. Many of the methods described in this chapter can be called on all of the objects.

The Echo Argument

Oracle CRM On Demand Web services using the methods Insert, Update, InsertOrUpdate, Delete, InsertChild, UpdateChild, and DeleteChild can specify an Echo input argument; for example:

```
<Echo>Off</Echo>
```

The Echo value is case-sensitive and is used to indicate whether a Web services transaction is to trigger the creation of an integration event (assuming that a workflow and integration event action have been configured to capture that particular transaction type). The default value of Echo is `On`, therefore when this parameter is provided in the request with a value of `On`, or is not included in the request, the integration event action is triggered and the transaction is recorded. When the Echo parameter is included with a value of `Off`, the transaction is not recorded.

NOTE: For Java users, the Echo string is required for all input methods. The echo string can be set to Off.

The ViewMode Argument

The ViewMode argument is an input argument for all of the Web Services v2.0 API methods. It applies to parent objects only, and specifies the level of access to parent records identified in the method call. ViewMode can take one of the following values:

- **Manager.** Provides access to records for which the current user or a subordinate of the current user owns the records, or is part of the team that owns the records.
- **EmployeeManager.** Provides access to records for which the current user or a subordinate of the current user owns the records. This value is similar to Manager, but teams are not considered.
- **Sales Rep.** Provides access to records for which the current user is part of the team that owns the records.
- **Personal.** Provides access to records owned by the user.
- **Organization.** Provides access to records within the current user's organization.
- **Broadest.** Provides access to the maximum set of records allowed for the user. This is the default value.
- **AllBooks.** For objects that support book, provides access to all books. This value is similar to Broadest, and works like All+ with the Book Selector in the Oracle CRM On Demand UI.
- **Context.** For objects that support books, provides access to the default book. For access to a specific book, the QueryPage method must use the BookId and BookName arguments.

The order of preference from most restrictive to least restrictive is:

- Personal
- Sales Rep
- Organization

Performance may be improved when a restricted set of records is accessed.

The LOVLanguageMode Argument

The LOVLanguageMode argument is an input argument for all of the Web Services v2.0 API methods. It determines whether the processing for picklist fields occurs using language independent codes (LIC) or language dependent codes (LDC). The argument is applicable only to simple picklist fields (which have an Edit Picklist link in the Field Setup page in the Oracle CRM On Demand application).

The LOVLanguageMode argument can take one of two values: LIC or LDC. LIC is the default value.

For methods other than QueryPage:

- If LIC is specified, then Web Services On Demand expects LIC values and converts them to LDV (based on the user’s current setting) and performs the processing.
- If LDC is specified, no translation is performed and values are expected to be in the user’s language.

For the QueryPage method:

- If LIC is specified in the query, the response contains picklist fields with language independent values.
- If LDC is specified in the query, the response contains picklist values in the user's language.

Delete

Removes records of a specified record type from the Oracle CRM On Demand database.

Usage

You use the Delete method to remove one or more records of a particular object from an Oracle CRM On Demand instance.

The deleted records appear in the Deleted Items folder and can be restored through the Oracle CRM On Demand UI.

The Delete call can be used to delete both parent record and child records. If the child node specified in the request is available, it deletes the child in the existing parent record, and leaves the parent record undeleted. However, if the child node is missing, it deletes the existing parent record.

Arguments

Table 33 describes the arguments taken by the Delete method.

Table 33. Arguments Taken by the Delete Method

Name	Description	Required	Default	I/O
ListOf*(Object)*. For example, ListOfAccount	The list of object instances to be deleted.	Yes	Not applicable	Input/ Output
LOVLanguageMode	The language mode for picklists, see "The LOVLanguageMode Argument" on page 104.	No	LIC	Input
Echo	Controls whether data sent to Oracle CRM On Demand through integration Web services are recorded as transactions.	No	On	Input
ViewMode	Specifies the level of access to records specified in the method call, see "The ViewMode Argument" on page 104.	No	Broadest	Input

Return Value of the Call

The status key for each of the deleted objects.

Sample SOAP Request - Delete

The following SOAP request contains two contact records, one of which specifies a parent contact record and one of its child account records, while the other only specifies a parent contact record. When a child record is included in the request, the parent record is ignored and only the specified child record is deleted, whereas when no child record is included in the request, the parent record is deleted.

The contact ID is specified as the user key to identify the contact records in the request.

Refer to "User Key Fields on the Contact Object" on page 413 for a list of valid user key field combinations for the contact object.

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%USERNAME%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%PASSWORD%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <ContactDelete_Input xmlns="urn:crmondemand/ws/ecbs/contact/">

      <ListOfContact>

        <Contact>

          <Id>ADSA-96IZCS</Id>

          <ListOfAccount>

            <Account>

              <AccountName>A1</AccountName>

              <AccountLocation>HQ</AccountLocation>

            </Account>

          </ListOfAccount>

        </Contact>

        <Contact>

          <Id>ADSA-96J5ZL</Id>
```

```
        </Contact>

      </ListOfContact>

    </ContactDelete_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response

The following SOAP response contains the returned status keys for the deleted objects:

```
<?xml version="1.0" encoding="utf-16"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <SOAP-ENV:Body>

    <ns:ContactDelete_Output xmlns:ns="urn:crmondemand/ws/ecbs/contact/">

      <ListOfContact xmlns="urn:/crmondemand/xml/Contact/Data">

        <Contact>

          <ModifiedDate>2011-08-10T15:21:43Z</ModifiedDate>

          <CreatedDate>2011-08-09T21:17:05Z</CreatedDate>

          <ModifiedById>HRZ780-I5RUE</ModifiedById>

          <CreatedById>HRZ780-I5RUE</CreatedById>

          <ModId>4</ModId>

          <Id>ADSA-96IZCS</Id>

          <CreatedBy>Joanne Brown, 08/09/2011 16:17:05</CreatedBy>

          <ModifiedBy>Joanne Brown, 08/10/2011 10:21:43</ModifiedBy>

          <ListOfAccount>

            <Account>

              <ModifiedDate>2011-08-10T14:44:27Z</ModifiedDate>

              <CreatedDate>2011-08-10T14:44:27Z</CreatedDate>

              <ModifiedById>HRZ780-I5RUE</ModifiedById>

              <CreatedById>HRZ780-I5RUE</CreatedById>

              <ModId>0</ModId>

              <Id>ADSA-977GCL</Id>

              <CreatedBy>Joanne Brown, 08/10/2011 09:44:27</CreatedBy>

              <ModifiedBy>Joanne Brown, 08/10/2011 09:44:27</ModifiedBy>

            </Account>

          </ListOfAccount>

        </Contact>

        <Contact>
```

```
            <ModifiedDate>2011-08-10T15:22:46Z</ModifiedDate>
            <CreatedDate>2011-08-09T21:13:33Z</CreatedDate>
            <ModifiedById>HRZ780-I5RUE</ModifiedById>
            <CreatedById>HRZ780-I5RUE</CreatedById>
            <ModId>2</ModId>
            <Id>ADSA-96J5ZL</Id>
            <CreatedBy>Joanne Brown, 08/09/2011 16:13:33</CreatedBy>
            <ModifiedBy>Joanne Brown, 08/10/2011 10:22:46</ModifiedBy>
          </Contact>
        </ListOfContact>
      </ns:ContactDelete_Output>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Execute

Executes multiple insert, delete, or update operations on separate records within a single Web services request.

Usage

You use the Execute method to perform different operations on records within the same Web services request for a single object.

The operations are defined by the operation attribute on the object element in the SOAP request, for example:

```
<Account operation='insert”>
```

specifies an insert operation for an Account object.

The operation attribute can have one of the following values:

- **update.** Updates the specified record
- **insert.** Inserts the specified record.
- **delete.** Deletes the matching record.
- **skipnode.** Skips the parent object and processes the child objects.

Every object specified in the call requires one valid operation. Mixed node operations are allowed for the Execute method.

Arguments

Table 34 describes the arguments taken by the Execute method.

Table 34. Arguments Taken by the Execute Method

Name	Description	Required	Default	I/O
ListOf*(Object)*. For example, ListOfAccount	The list of object instances to be inserted, deleted, or updated.	Yes	Not applicable	Input/ Output
LOVLanguageMode	The language mode for picklists, see "The LOVLanguageMode Argument" on page 104.	No	LIC	Input
ViewMode	Specifies the level of access to records specified in the method call, see "The ViewMode Argument" on page 104.	No	Broadest	Input
Echo	Controls whether data sent to Oracle CRM On Demand through integration Web services are recorded as transactions.	No	On	Input

Return Value of the Call

The status key for each of the Oracle CRM On Demand objects.

Sample SOAP Request - Execute

The following SOAP request performs Execute operations for four separate contact records in a single Web service request. The first operation inserts a record, the second updates, the third deletes, and the final operation ignores the parent contact record and performs an update on the activity child record.

The ContactFirstName and ContactLastName is specified as the user key to identify the contact records in the request.

Refer to "User Key Fields on the Contact Object" on page 413 for a list of valid user key field combinations for the contact object.

NOTE: For operations delete or update, if there are multiple records in Oracle CRM On Demand with the same ContactFirstName and ContactLastName, the request fails with a multiple match error. It is recommended to use the ID user key field to uniquely identify records as it avoids multiple match errors and also improves performance.

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>
```

```
    <wsse:Security>
      <wsse:UsernameToken>
        <wsse:Username>%%USERNAME%%</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%PASSWORD%%</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <ContactExecute_Input xmlns="urn:crmondemand/ws/ecbs/contact/">
      <ListOfContact>
        <Contact operation="insert">
          <ContactFirstName>NewCon5</ContactFirstName>
          <ContactLastName>NewCon5</ContactLastName>
        </Contact>
        <Contact operation="update">
          <ContactFirstName>Nathan</ContactFirstName>
          <ContactLastName>Lak</ContactLastName>
          <ContactEmail>n@l.com</ContactEmail>
        </Contact>
        <Contact operation="delete">
          <ContactFirstName>NewCon5</ContactFirstName>
          <ContactLastName>Con5</ContactLastName>
        </Contact>
        <Contact operation="skipnode">
          <ContactFirstName>NewCon6</ContactFirstName>
          <ContactLastName>NewCon6</ContactLastName>
          <ListOfActivity>
            <Activity operation="update">
              <Subject>Subject 1</Subject>
              <Activity>Task</Activity>
              <Priority>1-High</Priority>
            </Activity>
          </ListOfActivity>
        </Contact>
      </ListOfContact>
    </ContactExecute_Input>
```

```
  </soap:Body>
</soap:Envelope>
```

Sample SOAP Response

The following SOAP response contains the returned status keys for the contact objects:


```
<?xml version="1.0" encoding="utf-16"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <SOAP-ENV:Body>

    <ns:ContactExecute_Output xmlns:ns="urn:crmondemand/ws/ecbs/contact/">

      <ListOfContact xmlns="urn:/crmondemand/xml/Contact/Data">

        <Contact>

          <ModifiedDate>2011-08-25T15:28:34Z</ModifiedDate>

          <CreatedDate>2011-08-25T15:28:34Z</CreatedDate>

          <ModifiedById>HRZ780-I5RUE</ModifiedById>

          <CreatedById>HRZ780-I5RUE</CreatedById>

          <ModId>0</ModId>

          <Id>ADSA-9HXB9V</Id>

          <CreatedBy>Joanne Brown, 08/25/2011 10:28:34</CreatedBy>

          <ModifiedBy>Joanne Brown, 08/25/2011 10:28:34</ModifiedBy>

        </Contact>

        <Contact>

          <ModifiedDate>2011-08-25T15:28:36Z</ModifiedDate>

          <CreatedDate>2011-08-04T17:01:41Z</CreatedDate>

          <ModifiedById>HRZ780-I5RUE</ModifiedById>

          <CreatedById>HRZ780-I5RUE</CreatedById>

          <ModId>5</ModId>

          <Id>ADSA-93DZIC</Id>

          <CreatedBy>Joanne Brown, 08/04/2011 12:01:41</CreatedBy>

          <ModifiedBy>Joanne Brown, 08/25/2011 10:28:36</ModifiedBy>

        </Contact>

        <Contact>

          <ModifiedDate>2011-08-12T15:11:17Z</ModifiedDate>

          <CreatedDate>2011-08-12T15:11:17Z</CreatedDate>

          <ModifiedById>HRZ780-I5RUE</ModifiedById>

          <CreatedById>HRZ780-I5RUE</CreatedById>

          <ModId>0</ModId>
```

```
        <Id>ADSA-98V4V7</Id>
        <CreatedBy>Joanne Brown, 08/12/2011 10:11:17</CreatedBy>
        <ModifiedBy>Joanne Brown, 08/12/2011 10:11:17</ModifiedBy>
      </Contact>
    </ListOfContact>
  </ns:ContactExecute_Output>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Insert

Inserts a new record in the Oracle CRM On Demand database.

Usage

You use the Insert method to create one or more records of a particular object in an Oracle CRM On Demand instance.

When inserting a batch of records, the batch is treated as a single transaction. If one record fails to insert during a batch insertion, the entire operation is rolled back and no records are inserted.

The Insert method can be used to insert both parent records and child records. If a child node is specified in the request, the Insert method inserts the child and associates it with the existing parent record. If a child node is missing, the Insert call inserts only the new parent record.

Arguments

Table 35 describes the arguments taken by the Insert method.

Table 35. Arguments Taken by the Insert Method

Name	Description	Required	Default	I/O
ListOf*(Object)*. For example, ListOfAccount	The list of object instances to be inserted.	Yes	Not applicable	Input/ Output
LOVLanguageMode	The language mode for picklists, see “The LOVLanguageMode Argument” on page 104.	No	LIC	Input

Table 35. Arguments Taken by the Insert Method

Name	Description	Required	Default	I/O
Echo	Controls whether data sent to Oracle CRM On Demand through integration Web services are recorded as transactions.	No	On	Input
ViewMode	Specifies the level of access to records specified in the method call, see "The ViewMode Argument" on page 104.	No	Broadest	Input

Return Value of the Call

The status key for each of the Oracle CRM On Demand objects.

Sample SOAP Request - Insert

The following SOAP request contains two contact records, one of which specifies a parent contact record and one of its child Account records, and the other only specifies a parent contact record. When a child record is included in the request, the parent record must exist in the system and the child record specified is inserted and associated to the existing parent record. When no child record is included in the request, the parent record is inserted.

The ContactFirstName and ContactLastName are specified as the user key to identify the contact records in the request.

Refer to "User Key Fields on the Contact Object" on page 413 for a list of valid user key field combinations for the contact object.

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%USERNAME%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%PASSWORD%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <ContactInsert_Input xmlns="urn:crmondemand/ws/ecbs/contact/">

      <Echo>On</Echo>
```

```
      <LOVLanguageMode>LIC</LOVLanguageMode>
      <ListOfContact>
        <Contact>
          <ContactFirstName>NewCon2</ContactFirstName>
          <ContactLastName>Con2</ContactLastName>
          <ExternalSystemId>12345009</ExternalSystemId>
        </Contact>
        <Contact>
          <ContactFirstName>ExistingCon1</ContactFirstName>
          <ContactLastName>ExistingCon1</ContactLastName>
          <ListOfAccount>
            <Account>
              <AccountName>Action Rentals</AccountName>
            </Account>
          </ListOfAccount>
        </Contact>
      </ListOfContact>
    </ContactInsert_Input>
  </soap:Body>
</soap:Envelope>
```

Sample SOAP Response

The following SOAP response contains the returned status keys for the inserted objects:

```
<?xml version="1.0" encoding="utf-16"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns:ContactInsert_Output xmlns:ns="urn:crmondemand/ws/ecbs/contact/">
      <ListOfContact xmlns="urn:/crmondemand/xml/Contact/Data">
        <Contact>
          <ModifiedDate>2011-08-10T14:44:24Z</ModifiedDate>
          <CreatedDate>2011-08-10T14:44:24Z</CreatedDate>
          <ModifiedById>HRZ780-I5RUE</ModifiedById>
          <CreatedById>HRZ780-I5RUE</CreatedById>
          <ModId>0</ModId>
          <Id>ADSA-977GCJ</Id>
          <CreatedBy>Joanne Brown, 08/10/2011 09:44:24</CreatedBy>
```

```
          <ModifiedBy>Joanne Brown, 08/10/2011 09:44:24</ModifiedBy>
        </Contact>
        <Contact>
          <ModifiedDate>2011-08-10T14:41:55Z</ModifiedDate>
          <CreatedDate>2011-08-09T21:17:05Z</CreatedDate>
          <ModifiedById>HRZ780-I5RUE</ModifiedById>
          <CreatedById>HRZ780-I5RUE</CreatedById>
          <ModId>2</ModId>
          <Id>ADSA-96IZCS</Id>
          <CreatedBy>Joanne Brown, 08/09/2011 16:17:05</CreatedBy>
          <ModifiedBy>Joanne Brown, 08/10/2011 09:41:55</ModifiedBy>
          <ListOfAccount>
            <Account>
              <ModifiedDate>2011-08-10T14:44:27Z</ModifiedDate>
              <CreatedDate>2011-08-10T14:44:27Z</CreatedDate>
              <ModifiedById>HRZ780-I5RUE</ModifiedById>
              <CreatedById>HRZ780-I5RUE</CreatedById>
              <ModId>0</ModId>
              <Id>ADSA-977GCL</Id>
              <CreatedBy>Joanne Brown, 08/10/2011 09:44:27</CreatedBy>
              <ModifiedBy>Joanne Brown, 08/10/2011 09:44:27</ModifiedBy>
            </Account>
          </ListOfAccount>
        </Contact>
      </ListOfContact>
    </ns:ContactInsert_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


QueryPage

Executes a query against the set of records for an object, and returns the subset of the records that match the search criteria set by the method arguments.

NOTE: For fields of all data types apart from xsd:string, if the field has a blank or no value, it does not appear in the query response.

Query by Template

To improve performance, the QueryPage result contains only those fields and objects that are included in the QueryPage request. To retrieve the values of fields that are not a part of the search criteria, the field must be included in the search request with a blank value.

Using the pagesize, startrownum, and recordcountneeded Arguments

The pagesize argument, which has a maximum value of 100, is used to specify the maximum number of records to be returned in a QueryPage response. The pagesize and startrownum arguments are specified as attributes of the ListOf*(Object)* element in requests, for example:

```
<ListOfAccount pagesize="20" startrownum="0" recordcountneeded="true">
```

Depending on the value of pagesize, records are returned as follows:

- If the number of records in the record set is less than the pagesize value, the full record set is returned, and the lastpage attribute is set to true.
- If the number of records in the record set exceeds the pagesize value, only the number of records specified by the pagesize parameter is returned, and the lastpage attribute is false.
- For a query whose record set exceeds the pagesize value, setting the startrownum attribute to pagesize+1 (setting startrownum to pagesize returns the next pagesize number of records) returns the next pagesize number of records.
- If the size of the record set is greater than pagesize, and this is a subsequent query where there are less than pagesize number of records remaining to be returned, all of the remaining records are returned and lastpage attribute has a value of true.

Even though the QueryPage method returns a limited number of records, it keeps the data in the cache, which you can then retrieve by calling the QueryPage method again with a new value for the startrownum argument.

If you set the recordcountneeded argument to `true`, the record count is returned, except when the search specification includes calculated fields:

- If the QueryPage request contains filter criteria on a calculated field, or a calculated field is part of the value of the searchspec attribute, and there are records that match the given filter criteria, an error is thrown.
- If the QueryPage request contains filter criteria on a calculated field, or a calculated field is part of the value of the searchspec attribute, and there are no records that match the filter criteria (`recordcount = "0"` returned), no error is thrown.

NOTE: There is a performance impact if recordcountneeded is set to `true`, especially if the search specification contains calculated fields.

Using Search Specifications

You can use the searchspec argument of QueryPage to return only those records matching specified search criteria for an object. You can also specify the sort order and sort sequence for returned records using the sortorder and sortsequence arguments respectively. For more information, see "Specifying the Sort Order and Sort Sequence" on page 123.

The searchspec argument is specified as an attribute of an *Object* element in requests, for example:

```
<Contact searchspec="[ContactFirstName] = 'Jo*'">
```

which would return only contact records for which the ContactFirstName field value begins with Jo. The search specification can be set on any field type. The query syntax is described in the following topic.

Query Syntax

The query syntax for the searchspec argument supports only a small subset of binary and unary operators. No Siebel Query Language constructs or functions are supported. The query syntax is summarized in Table 36.

Table 36. Query Syntax for QueryPage

Syntax Type	Notes
`expression`	
[*XML Tag*] {*Operator*} {*Value*}	*Operator* can be binary or unary. The {*Value*} need only be specified for binary operators.
(*`expression`*) *`conjunction`* (*`expression`*)	A conjugated expression must be enclosed in parentheses to avoid ambiguity. However, nonconjugated expressions must not be enclosed in parentheses.
`conjunction`	
`OR`	None
`AND`	None
`unary operator`	
IS NULL	Used to find a match for a value that has no value
IS NOT NULL	None
EXISTS	See "The EXISTS and NOT EXISTS Operators" on page 122
NOT EXISTS	None
`binary operator`	
=	None
~=	Denotes a case-insensitive exact search (no wildcards used)
<	Must be specified as < to ensure well-formed XML.
<=	Must be specified as <=
>	None
>=	None
<>	Must be specified as <>
LIKE	Wildcard characters are treated as such only in the context of the operator LIKE.
~LIKE	Denotes a case-insensitive wildcard search

Table 36. Query Syntax for QueryPage

Syntax Type	Notes
`value`	
`'literal'`	Literal data is always enclosed in single quotes. To use a single quote within a literal, place another single quote immediately beside that quote. In this way, the query recognizes the quote as a literal and not as an operator. For example, the string ab'c is specified as ab''c. To use the special characters such as asterisk (*), question mark (?), and backslash (\) in queries, preceded them with the \ character. For example, to use the ? wildcard operator in a query, precede it with the \ character as follows: \?

CAUTION: Queries that are case-insensitive or with leading wildcard characters will perform slowly and must only be used when necessary.

The following is an example from a request that follows the query syntax:

```
<Contact searchspec=”[ContactFirstName] = ‘John’”>
```

where *XML Tag* is Contact FirstName, the operator is = and the value is the literal value John.

An example of an expression with a conjunction is as follows:

```
<Contact searchspec=”([ContactFirstName] = ‘Jane’”) AND ([ContactLastName] = ‘Doe’”)>
```

Examples of searchspec usage are given in Table 37.

Table 37. Searchspec Examples

Operator	Usage of searchspec in request	Description
Case Sensitive LIKE with * wildcard	<ListOfContact startrownum="0" pagesize="100" recordcountneeded="true"> <Contact searchspec=”[ContactFirstName] LIKE ‘Contact*’”> <ContactFirstName/> </Contact></ListOfContact>	Returns all contacts whose ContactFirstName value starts with "Contact" and ends with zero or more characters
Case Insensitive ~ LIKE with * wildcard	<ListOfContact><Contact searchspec=”[ContactFirstName] ~LIKE ‘Contact*’”> <ContactFirstName/> </Contact></ListOfContact>	Returns all contacts whose ContactFirstName value starts with, for example, "Contact" or "contact" or "CoNtAcT" and ends with zero or more characters
Case Sensitive LIKE with ? wildcard	<ListOfContact><Contact searchspec=”[ContactFirstName] LIKE ‘Contact???’”> <ContactFirstName/> </Contact></ListOfContact>	Returns all contacts whose ContactFirstName value starts with "Contact" and ends with any three characters
Case Insensitive ~ LIKE with ? wildcard	<ListOfContact><Contact searchspec=”[ContactFirstName] ~LIKE ‘Contact???’”> <ContactFirstName/> </Contact></ListOfContact>	Returns all contacts whose ContactFirstName value starts with, for example, "Contact" or "contact" or "CoNtAcT", and ends with any three characters
Case Sensitive =	<ListOfContact><Contact searchspec=”[ContactFirstName] = ‘ContactInsert’”> <ContactFirstName/> </Contact></ListOfContact>	Returns all contacts whose ContactFirstName value is equal to "ContactInsert"
Case Insensitive ~=	<ListOfContact><Contact searchspec=”[ContactFirstName] ~= ‘ContactInsert’”> <ContactFirstName/> </Contact></ListOfContact>	Returns all contacts whose ContactFirstName value is equal to, for example, "ContactInsert","contactinsert" or "CoNtAcTiNsErT"
IS NULL	<ListOfContact><Contact searchspec=”[TEXTLG_000] IS NULL> <ContactFirstName/> </Contact></ListOfContact>	Returns all contacts whose TEXTLG_000 value is NULL

Table 37. Searchspec Examples

Operator	Usage of searchspec in request	Description
IS NOT NULL	<ListOfContact><Contact searchspec="[TEXTLG_000] IS NOT NULL> <ContactFirstName/> </Contact></ListOfContact>	Returns all contacts whose TEXTLG_000 value is not NULL
AND	<ListOfContact><Contact searchspec="[BOOL_000] = 'Y' AND [IndexedBoolean0] = 'N'"> <ContactFirstName/><IndexedBoolean0/> </Contact></ListOfContact> Or <ListOfContact> <Contact searchspec="[BOOL_000] = 'Y'"> <ContactFirstName/> <IndexedBoolean0>='N'</IndexedBoolean0> </Contact></ListOfContact> Or <ListOfContact><Contact> <ContactFirstName/> <BOOL_000>='Y'</BOOL_000> <IndexedBoolean0>='N'</IndexedBoolean0> </Contact></ListOfContact>	Returns all contacts whose BOOL_000 value is Y AND IndexedBoolean0 value is N
OR	<ListOfContact><Contact searchspec="[BOOL_000] = 'Y' OR [IndexedBoolean0] = 'N'"> <ContactFirstName/> <BOOL_000/> <IndexedBoolean0/> </Contact></ListOfContact>	Returns all contacts whose BOOL_000 value is Y OR IndexedBoolean0 value is N
> (greater than)	<ListOfContact><Contact searchspec="[IndexedNumber0] > '500'"> <ContactFirstName/> <IndexedNumber0/> </Contact></ListOfContact>	Returns all contacts whose IndexedNumber0 value is greater than 500
>= (greater than or equal to)	<ListOfContact><Contact searchspec="[IndexedNumber0] >= '500'"> <ContactFirstName/> <IndexedNumber0/> </Contact></ListOfContact>	Returns all contacts whose IndexedNumber0 value is greater than or equal to 500

Table 37. Searchspec Examples

Operator	Usage of searchspec in request	Description

<
 	
 (less than)
	`<ListOfContact><Contact`
	`searchspec=”[IndexedNumber0] < '500'">`
	`<ContactFirstName/>`
	`<IndexedNumber0/>`
	`</Contact></ListOfContact>`
		Returns all contacts whose IndexedNumber0 value is less than 500

<=
 	
 (less than or equal to)
	`<ListOfContact><Contact`
	`searchspec=”[IndexedNumber0] <= '500'">`
	`<ContactFirstName/>`
	`<IndexedNumber0/>`
	`</Contact></ListOfContact>`
		Returns all contacts whose IndexedNumber0 value is less than or equal to 500

<>
 	
 (not equal to)
	`<ListOfContact><Contact`
	`searchspec=”[IndexedNumber0] <> '500'">`
	`<ContactFirstName/>`
	`<IndexedNumber0/>`
	`</Contact></ListOfContact>`
		Returns all contacts whose IndexedNumber0 value is not equal to 500

The EXISTS and NOT EXISTS Operators

You can use the unary operators EXISTS and NOT EXISTS with the searchspec argument of the QueryPage operation for any multi-valued field of objects supported for Web Services v2.0.

EXISTS and NOT EXISTS only work for multi-valued fields and not on other fields, including multi-select picklists. Very few multi-valued fields are available, but one example is the address fields on the Account and Contact parent objects. Multi-valued fields contain data from multiple records, in the case of the address fields, multiple countries, counties, and so on.

EXISTS returns true if the field expression matches for at least one of the values of the multi-valued field.

For example, to query for all contacts where any one of the values of the AlternateAddressExternalSystemId field for all associated addresses of Contact is XYZ:

```
<Contact searchspec="EXISTS ([AlternateAddressExternalSystemId] = 'XYZ')">
```

NOT EXISTS returns true if the field expression matches for none of the values of the multi-valued field.

For example, to query for all contacts where none of the values of the AlternateAddressExternalSystemId field for all associated addresses of Contact is XYZ:

```
<Contact searchspec="NOT EXISTS ([AlternateAddressExternalSystemId] = 'XYZ')">
```

As another example, to query for all accounts where any one of the values of the PrimaryBillToCounty field is Suffolk and any one of the values of the Primary Bill To Street Address field contains Ipswich:

```
<Account searchspec="EXISTS ([PrimaryBillToCounty] = 'Suffolk' AND
[PrimaryBillToStreetAddress] LIKE '%Ipswich%')
```

Specifying the Sort Order and Sort Sequence

You can specify the sort order and sort sequence for returned records using the sortorder and sortsequence arguments respectively.

The sortorder and sortsequence arguments are specified as attributes of a *FieldName* element in requests, for example:

```
<CampaignName sortorder=”ASC” sortsequence=”1”/>
```

which specifies that the records returned are sorted on the CampaignName field in an ascending order.

The sortsequence argument is used to specify the order in which sorting is applied if a sortorder value is specified on more than one field, for example:

```
<CampaignName sortorder=”ASC” sortsequence=”1”/>

<Status sortorder=”ASC” sortsequence=”2”/>"
```

Examples of sortorder and sortsequence usage are given in Table 38.

NOTE: The sortorder argument is not supported for fields of date type CLOB or BLOB. Such fields could be used with the searchpsec argument with a LIKE operator, however, for performance reasons, use of CLOB and BLOB fields even for filtering must be avoided.

CAUTION: Using multiple sort criteria may impact query performance and should only be used when necessary.

Table 38. Sortorder and sortsequence Examples

Sort type	Usage of sortoder and sortsequence	Description
Sort in ascending order	<ListOfContact startrownum="0" pagesize="100" recordcountneeded="true"> <Contact searchspec=”[ContactFirstName] LIKE ‘Contact*’”> <ContactLastName sortorder="ASC"></ContactLastName> </Contact></ListOfContact>	Returns all contacts whose ContactFirstName value starts with "Contact" and ends with zero or more characters and sorts them by ContactLastName in ascending order

Table 38. Sortorder and sortsequence Examples

Sort type	Usage of sortoder and sortsequence	Description

Sort in descending order	`<ListOfContact startrownum="0" pagesize="100" recordcountneeded="true">`
	`<Contact>`
	`<ContactLastName sortorder="DESC">~LIKE 'Contact*'</ContactLastName>`
	`</Contact></ListOfContact>`
		Returns all contacts whose ContactLastName value starts with "Contact" and ends with zero or more characters and sorts them by ContactLastName in descending order

Sort More than One Field	`<ListOfContact startrownum="0" pagesize="100" recordcountneeded="true">`
	`<Contact searchspec=”[ContactFirstName] LIKE ‘Contact*’”>`
	`<ContactLastName sortorder="DESC" sortsequence="1"></ContactLastName>`
	`<ContactFirstName sortorder="DESC" sortsequence="2"></ContactFirstName>`
	`<ExternalSystemId sortorder="DESC" sortsequence="3"></ExternalSystemId>`
	`</Contact></ListOfContact>`
		Returns all contacts whose ContactFirstName value starts with "Contact" and ends with zero or more characters and sorts them by ContactLastName in descending order and then by ContactFirstName in descending order and then by ExternalSystemId in descending order.

Specifying Books in Queries

For queries on record types that support books, you can use the BookId or BookName arguments to constrain the query to only the records in a particular book. For more information about the use of books, see the online help for Oracle CRM On Demand.

Querying Lists

You can use the optional NamedSearchSpec argument to query a list. NamedSearchSpec can specify the name of a System, Public or Private list, however, a user cannot specify the name of a list to which he or she does not have access to in the UI.

The set of fields to be returned must be included in the SOAP request. The set of fields defined in the list as well as the sort order of the list are not used to define the response, however, the query uses the filter criteria defined in the UI for the list.

All languages are supported.

Arguments

Table 39 describes the arguments taken by the QueryPage method.

Table 39. Arguments Taken by the QueryPage Method

Name	Description	Required	Default	I/O
ListOf*(Object)*. For example, ListOfAccount	The list of object instances queried (input), and after query execution, the list of object instances returned (output).	Yes	Not applicable	Input/ Output
pagesize	The maximum number of records displayed on a page following a query.	No	10	Input
startrownum	Indicates the row from which the QueryPage method starts to return records. For example, if you want to return records 1-100, you set StartRowNum to 0. Then, if you want to return records 101-200, you set StartRowNum to 100, and run the query again. You continue doing this until the last page is returned. In this way, you can return all records for a particular query.	No	0	Input
recordcountneeded	Indicates whether a record count for the object is to be returned. **NOTE:** There is a performance impact if this argument is set to true. You cannot return a record count for requests in which the search specification contains calculated fields. For more information, see "Using the pagesize, startrownum, and recordcountneeded Arguments" on page 116.	No	false	Input
searchspec	Indicates the search specification for a field or for all the fields of an object.	No	Not applicable	Input
sortsequence	An integer value that determines the order of the sort specification.	No	Not applicable	Input
sortorder	Determines the sort order for the records returned by the query, either ASC for ascending or DESC for descending.	No	ASC	Input
NamedSearchSpec	Indicates the search specification for a named list for an object.	No	Not applicable	Input

Table 39. Arguments Taken by the QueryPage Method

Name	Description	Required	Default	I/O
LOVLanguageMode	The language mode for picklists, see "The LOVLanguageMode Argument" on page 104.	No	LIC	Input
ViewMode	Specifies the level of access to records specified in the method call, see "The ViewMode Argument" on page 104.	Yes	Broadest	Input
BookId	The book ID.	No	Not applicable	Input
BookName	The book name. This argument is ignored if a value for BookId is supplied.	No	Not applicable	Input
IncludeSubBooks	Whether subbooks are to be included.	No	False	Input

Return Value of the Call

An object or list of objects of the type on which the method was called:

- **LastPage.** A Boolean value that indicates whether or not the last value in the query set has been returned.
- **recordcount.** An integer value that indicates the record count for the object. This value is only returned when recordcountneeded=true in the SOAP request.

Sample SOAP Requests

This topic contains a number of sample SOAP requests and responses to illustrate the use of the QueryPage API.

Sample SOAP Request 1 - QueryPage with Field-level Filter

The following SOAP request queries for contact records that have a ContactType like 'Customer*'. It demonstrates the use of optional attributes at the List of objects level: pagesize, startrownum, and recordcountneeded, and the use of optional elements <ViewMode> and <LOVLanguageMode>.

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%USERNAME%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%PASSWORD%%</wsse:Password>

      </wsse:UsernameToken>
```

```
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <ContactQueryPage_Input xmlns="urn:crmondemand/ws/ecbs/contact/">
      <ViewMode>Personal</ViewMode>
      <LOVLanguageMode>LIC</LOVLanguageMode>
      <ListOfContact pagesize="100" startrownum="0" recordcountneeded="true">
        <Contact>
          <ContactFirstName />
          <ContactLastName />
          <ContactType>LIKE 'Customer*'</ContactType>
          <ContactEmail />
          <ListOfActivity>
            <Activity>
              <Subject />
            </Activity>
          </ListOfActivity>
        </Contact>
      </ListOfContact>
    </ContactQueryPage_Input>
  </soap:Body>
</soap:Envelope>
```

Sample SOAP Response 1

The following SOAP response shows the list of 5 objects returned by SOAP request 1.

```
<?xml version="1.0" encoding="utf-16"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns:ContactQueryPage_Output xmlns:ns="urn:crmondemand/ws/ecbs/contact/">
      <ListOfContact xmlns="urn:/crmondemand/xml/Contact/Data" recordcount="5" lastpage="true">
        <Contact>
          <ContactEmail></ContactEmail>
          <ContactFirstName>D</ContactFirstName>
          <ContactLastName>Dan</ContactLastName>
          <ContactType>Customer</ContactType>
          <ListOfActivity lastpage="true">
```

```
    <Activity>
      <Subject>Send D Dan welcome email</Subject>
    </Activity>
  </ListOfActivity>
</Contact>
<Contact>
  <ContactEmail></ContactEmail>
  <ContactFirstName>L</ContactFirstName>
  <ContactLastName>Lak</ContactLastName>
  <ContactType>Customer</ContactType>
  <ListOfActivity lastpage="true">
    <Activity>
      <Subject>Send L Lak welcome email</Subject>
    </Activity>
  </ListOfActivity>
</Contact>
<Contact>
  <ContactEmail></ContactEmail>
  <ContactFirstName>NewCon6</ContactFirstName>
  <ContactLastName>Con6</ContactLastName>
  <ContactType>Customer</ContactType>
  <ListOfActivity lastpage="true">
    <Activity>
      <Subject>Add activity to C Contacts</Subject>
    </Activity>
  </ListOfActivity>
</Contact>
<Contact>
  <ContactEmail>n@l.com</ContactEmail>
  <ContactFirstName>Nathan</ContactFirstName>
  <ContactLastName>Lak</ContactLastName>
  <ContactType>Customer</ContactType>
  <ListOfActivity lastpage="true">
    <Activity>
      <Subject>Send Nathan Lak welcome email</Subject>
    </Activity>
```

```
          <Activity>
            <Subject>Send Jordan Lak welcome email</Subject>
          </Activity>
          <Activity>
            <Subject>Send J Lak welcome email</Subject>
          </Activity>
          <Activity>
            <Subject>ActivityInsert Test R19</Subject>
          </Activity>
          <Activity>
            <Subject>ActivityInsert Test R19 2</Subject>
          </Activity>
          <Activity>
            <Subject>ActivityInsert Test R19 2</Subject>
          </Activity>
        </ListOfActivity>
      </Contact>
      <Contact>
        <ContactEmail></ContactEmail>
        <ContactFirstName>NewCon5</ContactFirstName>
        <ContactLastName>NewCon5</ContactLastName>
        <ContactType>Customer</ContactType>
        <ListOfActivity lastpage="true"></ListOfActivity>
      </Contact>
    </ListOfContact>
  </ns:ContactQueryPage_Output>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample SOAP Request 2 - QueryPage with searchspec, sortsequence and sortorder Attributes

The following SOAP request queries for Contact records by specifying a search criterion with an OR conjunction in the searchspec attribute.

The samples also demonstrate use of the optional pagesize, startrownum, and recordcountneeded attributes at the <ListOf*Object*> level.

In SOAP request 2a, pagesize = 5 and startrownum = 0, this request returns the 5 records starting at row 0 (first record) that match the search criteria.

In SOAP request 2b, pagesize = 5 and startrownum = 4, this request returns the 5 records starting at row 4 (fifth record) that match the search criteria.

Notice that the last record returned in SOAP response 2a and the first record returned in SOAP response 2b are identical.

In addition, the sample demonstrates the sortsequence and sortorder attributes at the element level to specify the sort order and sort sequence of the whole result set (including rows of records not returned due to the pagesize limit). Note that across both responses, (SOAP response 2a and SOAP response 2b), the records are sorted as per the specified sort sequence and sort order.

Sample SOAP Request 2a

In this SOAP request, pagesize = 5 and startrownum = 0:

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%USERNAME%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">%%PASSWORD%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <ContactQueryPage_Input xmlns="urn:crmondemand/ws/ecbs/contact/">

      <ListOfContact pagesize="5" startrownum="0" recordcountneeded="true">

        <Contact searchspec="[ContactLastName] LIKE 'C*' OR [ContactType] = 'Customer'">

          <ContactFirstName></ContactFirstName>

          <ContactLastName sortorder="DESC" sortsequence="2"></ContactLastName>

          <ContactType sortorder="DESC" sortsequence="1"></ContactType>

        </Contact>

      </ListOfContact>

    </ContactQueryPage_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response 2a

The following SOAP response shows the list of 5 objects returned by SOAP request 2a:

```
<?xml version="1.0" encoding="utf-16"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <SOAP-ENV:Body>

    <ns:ContactQueryPage_Output xmlns:ns="urn:crmondemand/ws/ecbs/contact/">

      <ListOfContact xmlns="urn:/crmondemand/xml/Contact/Data" recordcount="54" lastpage="false">

        <Contact>

          <ContactFirstName>James</ContactFirstName>

          <ContactLastName>Conners</ContactLastName>

          <ContactType>Prospect</ContactType>

        </Contact>

        <Contact>

          <ContactFirstName>Ruth</ContactFirstName>

          <ContactLastName>Chou</ContactLastName>

          <ContactType>Prospect</ContactType>

        </Contact>

        <Contact>

          <ContactFirstName>Jason</ContactFirstName>

          <ContactLastName>Cheney</ContactLastName>

          <ContactType>Prospect</ContactType>

        </Contact>

        <Contact>

          <ContactFirstName>Sean</ContactFirstName>

          <ContactLastName>Thomas</ContactLastName>

          <ContactType>Customer</ContactType>

        </Contact>

        <Contact>

          <ContactFirstName>Carlos</ContactFirstName>

          <ContactLastName>Santos</ContactLastName>

          <ContactType>Customer</ContactType>

        </Contact>

      </ListOfContact>

    </ns:ContactQueryPage_Output>

  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

Sample SOAP Request 2b

In this SOAP request, pagesize = 5 and startrownum = 4:

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%USERNAME%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%PASSWORD%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <ContactQueryPage_Input xmlns="urn:crmondemand/ws/ecbs/contact/">

      <ListOfContact pagesize="5" startrownum="4" recordcountneeded="true">

        <Contact searchspec="[ContactLastName] LIKE 'C*' OR [ContactType] = 'Customer'">

          <ContactFirstName></ContactFirstName>

          <ContactLastName sortorder="DESC" sortsequence="2"></ContactLastName>

          <ContactType sortorder="DESC" sortsequence="1"></ContactType>

        </Contact>

      </ListOfContact>

    </ContactQueryPage_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response 2b

The following SOAP response shows the list of 5 objects returned by the request in SOAP request 2b:

```
<?xml version="1.0" encoding="utf-16"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <SOAP-ENV:Body>

    <ns:ContactQueryPage_Output xmlns:ns="urn:crmondemand/ws/ecbs/contact/">

      <ListOfContact xmlns="urn:/crmondemand/xml/Contact/Data" recordcount="54" lastpage="false">

        <Contact>

          <ContactFirstName>Carlos</ContactFirstName>

          <ContactLastName>Santos</ContactLastName>
```

```
          <ContactType>Customer</ContactType>
        </Contact>
        <Contact>
          <ContactFirstName>Joanna</ContactFirstName>
          <ContactLastName>Pressman</ContactLastName>
          <ContactType>Customer</ContactType>
        </Contact>
        <Contact>
          <ContactFirstName>Marianne</ContactFirstName>
          <ContactLastName>Pok</ContactLastName>
          <ContactType>Customer</ContactType>
        </Contact>
        <Contact>
          <ContactFirstName>Brian</ContactFirstName>
          <ContactLastName>Pittenger</ContactLastName>
          <ContactType>Customer</ContactType>
        </Contact>
        <Contact>
          <ContactFirstName>Marie</ContactFirstName>
          <ContactLastName>Perkins</ContactLastName>
          <ContactType>Customer</ContactType>
        </Contact>
      </ListOfContact>
    </ns:ContactQueryPage_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


Sample SOAP Request 3 - QueryPage with NamedSearchSpec

The following SOAP request queries for contact records in a specific contact list by specifying the contact list name in the <NamedSearchSpec> element.

Table 40 shows the records in the C Contacts list from the Oracle CRM On Demand UI. There are a total of 11 records in the list, therefore the recordcount value returned in the SOAP response equals 11.

Table 40. Example of Records in a Contact List

Row	Last Name	First Name	Contact Type	Owner Alias
0	Cuthbery	David	Customer	Donna Jones
1	Coxe	Robin	Customer	Donna Jones
2	Connolly	John	Customer	Jeff Smith
3	Connors	James	Prospect	Jeff Smith
4	Cone	Alexander	Customer	Ian McAllistair
5	Con6	NewCon6	Customer	Joanne Brown
6	Cohen	Adam	Customer	Ian McAllistair
7	Clatt	Mark	Customer	Donna Jones
8	Chu	Will	Customer	Jeff Smith
9	Chou	Ruth	Prospect	Donna Jones
10	Cheney	Jason	Prospect	Ian McAllistair

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%USERNAME%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%PASSWORD%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <ContactQueryPage_Input xmlns="urn:crmondemand/ws/ecbs/contact/">

      <NamedSearchSpec>C Contacts</NamedSearchSpec>

      <ListOfContact pagesize="5" startrownum="0" recordcountneeded="true">

        <Contact>

          <ContactFirstName />

          <ContactLastName />

          <ContactType />
```

```
        </Contact>
      </ListOfContact>
    </ContactQueryPage_Input>
  </soap:Body>
</soap:Envelope>
```

Sample SOAP Response 3

The following SOAP response shows the list of contacts returned by sample SOAP request 3.

```
<?xml version="1.0" encoding="utf-16"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns:ContactQueryPage_Output xmlns:ns="urn:crmondemand/ws/ecbs/contact/">
      <ListOfContact xmlns="urn:/crmondemand/xml/Contact/Data" recordcount="11" lastpage="false">
        <Contact>
          <ContactFirstName>David</ContactFirstName>
          <ContactLastName>Cuthbery</ContactLastName>
          <ContactType>Customer</ContactType>
        </Contact>
        <Contact>
          <ContactFirstName>Robin</ContactFirstName>
          <ContactLastName>Coxe</ContactLastName>
          <ContactType>Customer</ContactType>
        </Contact>
        <Contact>
          <ContactFirstName>John</ContactFirstName>
          <ContactLastName>Connolly</ContactLastName>
          <ContactType>Customer</ContactType>
        </Contact>
        <Contact>
          <ContactFirstName>James</ContactFirstName>
          <ContactLastName>Conners</ContactLastName>
          <ContactType>Prospect</ContactType>
        </Contact>
        <Contact>
          <ContactFirstName>Alexander</ContactFirstName>
          <ContactLastName>Cone</ContactLastName>
```

```
          <ContactType>Customer</ContactType>
        </Contact>
      </ListOfContact>
    </ns:ContactQueryPage_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


Update

Updates the selected record with the new value.

Usage

You use the Update method to update one or more records of a particular object in an Oracle CRM On Demand instance.

The Update method can be used to update parent records and child records. If a child node is specified in the request, the Update method updates the child in the existing parent record. If the child node is missing, the Update method updates only the existing parent record.

NOTE: If the administrator customizes a record type to add a required field, Oracle CRM On Demand does not check for the required field when existing records are updated. When you update the record without the required field through a Web services request, or merge it with a record that does not have the required field, the record is updated or merged without error. This is the intended behavior; when a field is made required, it is the responsibility of the administrator to update all existing records to populate the required field. When inserting new records however, Oracle CRM On Demand checks for the required field.

Arguments

Table 41 describes the arguments taken by the Update method.

Table 41. Arguments Taken by the Update Method

Name	Description	Required	Default	I/O
ListOf*(Object)*. For example, ListOfAccount	The object instance to be updated.	Yes	Not applicable	Input/ Output
LOVLanguageMode	The language mode for picklists, see "The LOVLanguageMode Argument" on page 104.	No	LIC	Input

Table 41. Arguments Taken by the Update Method

Name	Description	Required	Default	I/O
Echo	Controls whether data sent to Oracle CRM On Demand through integration Web services are recorded as transactions.	No	On	Input
ViewMode	Specifies the level of access to records specified in the method call, see "The ViewMode Argument" on page 104.	Yes	Broadest	Input

Return Value of the Call

The status key for the updated objects.

Sample SOAP Request - Update

The following SOAP request updates an existing contact record.

The ContactFirstName and ContactLastName are specified as the user key to identify the contact records in the request.

Refer to "User Key Fields on the Contact Object" on page 413 for a list of valid user key field combinations for the contact object.

NOTE: If there are multiple records in Oracle CRM On Demand with the same ContactFirstName and ContactLastName, the update request fails with a multiple match error. It is recommended to use the ID user key field to uniquely identify records as it avoids multiple match errors and also improves performance.

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%USERNAME%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">%%PASSWORD%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <ContactUpdate_Input xmlns="urn:crmondemand/ws/ecbs/contact/">

      <Echo>On</Echo>
```

```
      <LOVLanguageMode>LIC</LOVLanguageMode>
      <ListOfContact>
        <Contact>
          <ContactFirstName>ExistingCon1</ContactFirstName>
          <ContactLastName>ExistingCon1</ContactLastName>
          <ContactEmail>e@e.com</ContactEmail>
        </Contact>
      </ListOfContact>
    </ContactUpdate_Input>
  </soap:Body>
</soap:Envelope>
```

Sample SOAP Response

The following SOAP response contains the returned status keys for the updated objects:

```
<?xml version="1.0" encoding="utf-16"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns:ContactUpdate_Output xmlns:ns="urn:crmondemand/ws/ecbs/contact/">
      <ListOfContact xmlns="urn:/crmondemand/xml/Contact/Data">
        <Contact>
          <ModifiedDate>2011-08-10T16:42:36Z</ModifiedDate>
          <CreatedDate>2011-08-09T21:17:05Z</CreatedDate>
          <ModifiedById>HRZ780-I5RUE</ModifiedById>
          <CreatedById>HRZ780-I5RUE</CreatedById>
          <ModId>8</ModId>
          <Id>ADSA-96IZCS</Id>
          <CreatedBy>Joanne Brown, 08/09/2011 16:17:05</CreatedBy>
          <ModifiedBy>Joanne Brown, 08/10/2011 11:42:36</ModifiedBy>
        </Contact>
      </ListOfContact>
    </ns:ContactUpdate_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Service API Calls

The Oracle CRM On Demand Web Services service methods are listed in Table 42. The service methods are those methods that are not called on Oracle CRM On Demand record types. Instead, they are used to perform administrative tasks. The table also shows the Web service for each of the methods. You can download the WSDL file for each service from the Web Services Administration page in the Oracle CRM On Demand application.

Table 42. Web Services On Demand Service Methods

Method Name	Web Service	Comments
"AuthAuthorizationUser" on page 140	Authorization	Authenticates the current user and returns information about the user's company and the privileges enabled for the user's role.
"DeletedItemQueryPage" on page 142	Deleted Item	Gets information about deleted items.
"DeleteEvents" on page 148	Integration Event	Deletes events from an integration event queue.
"GetEvents" on page 150	Integration Event	Gets events from an integration event queue.
"GetLists" on page 158	Lists	Gets the set of lists for an object.
"GetMapping" on page 161	Mapping Service	Gets a list of the display names of fields for a particular record type and their associated XML tags.
"GetPicklistValues" on page 165	Picklist	Gets lists of picklist values.
"GetServerTime" on page 168	Time	Gets the server time.
"LoginHistoryQueryPage" on page 169	Login History	Gets information about user login history.
"MergeRecords" on page 173	Merge	Merges records.
"MetadataChangeSummaryQueryPage" on page 175	Metadata Change Summary	Gets a summary of changes to metadata.
"SalesProcessQueryPage" on page 178	Sales Process	Gets sales process information.
"SetPassword" on page 185	Password	Sets the passwords of users who use Oracle CRM On Demand.
"SetSessionTimeZone" on page 187	Time	Sets the time zone for a session.
"UpdatePicklist" on page 188	Picklist	Updates picklist values.

Table 42. Web Services On Demand Service Methods

Method Name	Web Service	Comments
"UpdateCascadingPicklists" on page 193	Picklist	Updates cascading picklist values.
"UserUsageQueryPage" on page 197	User Usage	Gets information about Web services utilization.

AuthAuthorizationUser

Authenticates the current user and returns information about the user's company and the list of privileges enabled for the user's role.

Usage

You use the AuthAuthorizationUser method to authenticate the current user and return information relevant to the user.

There are no input argument, the method simply authenticates the current user name and password, and returns the information for the authenticated user.

The optional SOAP header parameter <AuthenticateOnly> is intended for use only with the AuthAuthorizationUser method:

- Setting the <AuthenticateOnly> parameter to true and sending a subsequent request with security credentials results in the previous session being closed (that is, a logoff is performed), and a new session is created with the supplied credentials.
- If the SOAP header parameter <SessionKeepAlive> is set to true, and a subsequent request does not contain security credentials, the previous session remains open and the request will be considered authenticated.

NOTE: If both <AuthenticateOnly> and <SessionKeepAlive> are set to true, <AuthenticateOnly> takes precedence.

Arguments

Table 43 describes the arguments taken by the AuthAuthorizationUser method.

Table 43. Arguments Taken by the AuthAuthorizationUser Method

Name	Description	Required	Default	I/O
AuthenticatedUser	Information about the authenticated user.	Not applicable	Not applicable	Output

Return Value of the Call

The following information is returned.

- **UserId.** The user ID of the authenticated user.
- **TenantId.** The tenant for the authenticated user (SYS_TENANT_ID).
- **ListofPrivileges**. The set of privileges that have been enabled for the user's role.
- **OrganizationName.** The company name for the user.
- **Location.** The location of the company.
- **RoleId.** The role ID for the user's role.
- **Role**. The user's role.

Sample SOAP Request - AuthAuthorizationUser

The following SOAP request invokes AuthAuthorizationUser. No arguments are required.

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%%USERNAME%%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <AuthWS_AuthAuthorizationUser_Input xmlns="urn:crmondemand/ws/auth/" />

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response

The following illustrates the SOAP response (most of the <Privilege> elements are omitted as indicated by the ellipses (...)):

```
<?xml version="1.0" encoding="utf-16"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <SOAP-ENV:Body>

    <ns:AuthWS_AuthAuthorizationUser_Output xmlns:ns="urn:crmondemand/ws/auth/">
```

```
      <AuthenticatedUser xmlns="urn:/crmondemand/xml/auth">
        <User>
          <UserId>1-CWTEQ</UserId>
          <TenantId>1-CWJ8T</TenantId>
          <OrganizationName>Farley 75</OrganizationName>
          <Location>Toronto</Location>
          <RoleId>1-CWJU5</RoleId>
          <Role>Vertical Admin</Role>
          <ListOfPrivileges>
            <Privilege>
              <PrivilegeName>FULL_ACCESS_ANALYTICS</PrivilegeName>YTICS</PrivilegeName>
            </Privilege>
...
...
          </ListOfPrivileges>
        </User>
      </AuthenticatedUser>
    </ns:AuthWS_AuthAuthorizationUser_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

DeletedItemQueryPage

Returns details of deleted items.

Objects Supported

DeletedItemQueryPage is supported for all Web Services v1.0 and Web Services v2.0 accessible objects that can be deleted.

Usage

You use the DeletedItemQueryPage method to execute a query against the list of deleted records, and return a subset of the records that match the search criteria set by the method arguments.

The Type of the DeletedItems object returned by the DeletedItemQueryPage method is not always the same as that used in the UI of the Oracle CRM On Demand application, as shown in Table 44.

NOTE: In Table 44, the * characters are asterisk characters, and do not represent wildcard characters.

You must use the types shown in the table in queries for deleted item records. (The type is language independent.)

Table 44. Deleted Item Types Returned by DeletedItemQueryPage Method

UI Record Type	Deleted Item Type
Account	Account
Contact	Contact
Opportunity	Opportunity
Lead	Lead
Service Request	Service Request
Campaign	Campaign
Appointment	Action***Appointment
Solution	Solution
Account Note	Account Note
Account Private Note	Account Private Note
Contact Note	Contact Note
Contact Private Note	Contact Private Note
Note	Note
Opportunity Note	Opportunity Note
Opportunity Private Note	Opportunity Private Note
Organizations Note	Organizations Note
Service Request Note	Service Request Note
Account Attachment	Account Attachment
Action Attachment	Action Attachment
Contact Attachment	Contact Attachment
Opportunity Attachment	Opportunity Attachment
Service Request Attachment	Service Request Attachment
Organization	Organization
Position	Position
Task	Action***Task
Revenue	Revenue
Lead Attachment	Lead Attachment
Solution Attachment	Solution Attachment

Table 44. Deleted Item Types Returned by DeletedItemQueryPage Method

UI Record Type	Deleted Item Type
Campaign Attachment	Campaign Attachment
Campaign Note	Campaign Note
Forecast Revenue	Forecast Revenue
Asset	Asset Mgmt - Asset
Referral	VONDINS Referral***Referral
Sales Stage Attachment	Sales Stage Attachment
Portfolio	VONDINS Portfolio***Portfolio
Household	Household
Portfolio Child	VONDINS Portfolio Child***Portfolio
Medical Education Event	Pharma ME Event
Vehicle	Auto Vehicle
Channel Partner	Channel Partner
Fund Attachment	Fund Attachment
Fund Request Attachment	Fund Request Attachment
Smart Call	Pharma Template Call
Custom Object 01	OnDemand Custom Object 1
Custom Object 02	OnDemand Custom Object 2
Custom Object 03	OnDemand Custom Object 3
Custom Object *n* (where n = 04 and higher)	CustomObject*n* (where *n* = 4 and higher)
Sample Dropped	Pharma Call Sample Dropped
Product Detailed	Pharma Call Product Detailed
Contact Interest	Contact Interests
Fund	Fund
Fund Request	Fund Request
Fund Note	Fund Note
Fund Request Note	Fund Request Note
Custom Object 01 Note	OnDemand Custom Object 1 Note
Custom Object 02 Note	OnDemand Custom Object 2 Note
Custom Object 03 Note	OnDemand Custom Object 3 Note
Custom Object 01 Attachment	OnDemand Custom Object 1 Attachment
Custom Object 02 Attachment	OnDemand Custom Object 2 Attachment

Table 44. Deleted Item Types Returned by DeletedItemQueryPage Method

UI Record Type	Deleted Item Type
Custom Object 03 Attachment	OnDemand Custom Object 3 Attachment
Dealer Note	Dealer Note
Dealer Attachment	Dealer Attachment

Arguments

Table 45 describes the arguments taken by the DeletedItemQueryPage method.

Table 45. Arguments Taken by the DeletedItemQueryPage Method

Name	Description	Required	Default	I/O
ListOfDeletedItem	The list of object instances queried (input), and after query execution, the list of object instances returned (output).	Yes	Not applicable	Input/ Output
PageSize	The maximum number of records displayed on a page following a query.	No	10	Input
StartRowNum	Indicates the row from which the DeletedItemQueryPage method starts to return records. Use the StartRowNum argument to return a set of records for any given method. For example, if you want to return records 1-100, you set StartRowNum to 0. Then, if you want to return records 101-200, you set StartRowNum to 100, and run the query again. You continue doing this until the last page is returned. In this way, you can return all records for a particular query.	No	0	Input
LastPage	A value that indicates whether or not the last value in the query set has been returned.	Not applicable	Not applicable	Output

Return Value of the Call

The following information is returned for deleted items:

- **DeletedItemId**. The ID of the deleted item.
- **DeletedById**. The user ID of the user who deleted the item.

- **DeletedBy**. The name of the user who deleted the item.
- **DeletedDate**. The date on which the item was deleted.
- **Name**. The name of the deleted record.
- **ObjectId**. The object ID of the deleted record.
- **Type**. The type of the deleted record.
- **ExternalSystemId**. The external system ID of the item.

Sample SOAP Request - DeletedItemQueryPage

The following SOAP request queries the list of deleted items with the object type = "Contact". It demonstrates the use of the optional arguments <PageSize> and <StartRowNum> to return a specific number of records starting at a specific row.

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%%USERNAME%%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <DeletedItemWS_DeletedItemQueryPage_Input xmlns="urn:crmondemand/ws/deleteditem/">

      <ListOfDeletedItem>

        <DeletedItem>

          <DeletedItemId />

          <DeletedById />

          <DeletedBy />

          <DeletedDate />

          <Name />

          <ObjectId />

          <Type>='Contact'</Type>

          <ExternalSystemId />

        </DeletedItem>

      </ListOfDeletedItem>
```

```
      <PageSize>3</PageSize>
      <StartRowNum>0</StartRowNum>
    </DeletedItemWS_DeletedItemQueryPage_Input>
  </soap:Body>
</soap:Envelope>
```

Sample SOAP Response

The following SOAP response contains three deleted contact records:


```
<?xml version="1.0" encoding="utf-16"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns:DeletedItemWS_DeletedItemQueryPage_Output xmlns:ns="urn:crmondemand/ws/deleteditem/">
      <ns:LastPage>true</ns:LastPage>
      <ListOfDeletedItem xmlns="urn:/crmondemand/xml/deleteditem">
        <DeletedItem>
          <DeletedItemId>ADSA-921PLC</DeletedItemId>
          <DeletedBy>Joanne Brown</DeletedBy>
          <DeletedById>HRZ780-I5RUE</DeletedById>
          <DeletedDate>08/02/2011 15:19:24</DeletedDate>
          <Name>Person One</Name>
          <ObjectId>ADSA-921PL5</ObjectId>
          <Type>Contact</Type>
          <ExternalSystemId></ExternalSystemId>
        </DeletedItem>
        <DeletedItem>
          <DeletedItemId>ADSA-93EVML</DeletedItemId>
          <DeletedBy>Joanne Brown</DeletedBy>
          <DeletedById>HRZ780-I5RUE</DeletedById>
          <DeletedDate>08/04/2011 12:16:39</DeletedDate>
          <Name>Jordan Lak</Name>
          <ObjectId>ADSA-93EGXJ</ObjectId>
          <Type>Contact</Type>
          <ExternalSystemId>ABC321</ExternalSystemId>
        </DeletedItem>
        <DeletedItem>
          <DeletedItemId>ADSA-93EVFP</DeletedItemId>
```

```
            <DeletedBy>Joanne Brown</DeletedBy>
            <DeletedById>HRZ780-I5RUE</DeletedById>
            <DeletedDate>08/04/2011 12:20:44</DeletedDate>
            <Name>J Lak</Name>
            <ObjectId>ADSA-93ET5I</ObjectId>
            <Type>Contact</Type>
            <ExternalSystemId>abc223</ExternalSystemId>
          </DeletedItem>
        </ListOfDeletedItem>
      </ns:DeletedItemWS_DeletedItemQueryPage_Output>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

DeleteEvents

Deletes events from an integration event queue.

Objects Supported

DeleteEvents is supported for the following objects: Account, Activity, Allocation, Asset, Business Plan, Campaign, Contact, Contact State License, Custom Object 01 - 03, Custom Object 04 and higher, Dealer, Household, Inventory Period, Lead, MedEvent, Objective, Opportunity, Period, Portfolio, Product, ProductCategory, Sample Inventory, Sample Transaction, Sample Lot, Service Request, Solution, Territory, Transaction Item, Vehicle.

Usage

You use the DeleteEvents method of the Integration Event Web service to delete events from an integration event queue. Integration events are actions that are triggered based on meeting certain workflow criteria. An integration event stores information about data that has changed:

- User key information about the changed record, for example: objectID, externalsystemID
- Audit information, for example, created date, createdby, modified date, modified by

Integration events are stored in one or more company queues on the hosted environment. The maximum number of events in the queues is set by Customer Care. Contact Customer Care to request support for the Integration Event Web Service and to specify the maximum total size of the integration queues you require.

For more information about integration events and setting up workflow criteria, see the Oracle CRM On Demand online help.

The QueueName argument, if specified, must match the name of a queue defined for a company within Oracle CRM On Demand. The QueueName value is the localized display name value for the user. If the supplied name is not valid, an error message is returned.

If QueueName is not specified, the value is empty, or if the value is Default Queue, events are deleted from the default queue.

If the DateTime argument is supplied, all events older than the specified date and time are deleted. If the LastEventId argument is supplied, all events older than the specified event are deleted. If DateTime and LastEventId are not specified, all events are deleted from the queue.

If a valid queue name is provided but the LastEventId value provided does not match an event in that queue, an error is returned.

When there is a pending request to a queue, another call to the same queue cannot be invoked until the previous request completes. However, another call to a different queue can be invoked before the previous request completes.

You can delete events for all of the supported record types, or a subset of record types, depending on how you prepare the WSDL and Schema files associated with the Integration Event service, see "Preparing the Integration Event WSDL File and Schema Files" on page 157.

Arguments

Table 46 describes the arguments taken by the DeleteEvents method.

Table 46. Arguments Taken by the DeleteEvents Method

Name	Description	Required	Default	I/O
QueueName	The name of the integration event queue.	No	Default Queue	Input/ Output
DateTime	A date and time.	No	Not applicable	Input
LastEventId	An event ID	No	Not applicable	Input/ Output

Return Value of the Call

The ID of the last event deleted.

Sample SOAP Request - DeleteEvents

The following SOAP request deletes events from the queue specified. If both the <LastEventId> and <DateTime> arguments are supplied, the <LastEventId> is used to determine which events to delete.

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>
```

```
      <wsse:UsernameToken>
        <wsse:Username>%%%USERNAME%%%</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <IntegrationEventWS_DeleteEvents_Input xmlns="urn:crmondemand/ws/integrationevent/">
      <DateTime>08/09/2011 10:40:53</DateTime>
      <LastEventId>20110809152116757_QNMHRZ780-I5RT8_Contact_Contact_ADSA-96CRZK_1_ADSA-96CSA5.xml</
LastEventId>
      <QueueName>Default Queue</QueueName>
    </IntegrationEventWS_DeleteEvents_Input>
  </soap:Body>
</soap:Envelope>
```

Sample Soap Response

The following SOAP response contains the deleted event:

```
<?xml version="1.0" encoding="utf-16"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns:IntegrationEventWS_DeleteEvents_Output xmlns:ns="urn:crmondemand/ws/integrationevent/">
      <ns:LastEventId>20110809152116757_QNMHRZ780-I5RT8_Contact_Contact_ADSA-96CRZK_1_ADSA-96CSA5.xml</
ns:LastEventId>
      <ns:QueueName>Default Queue</ns:QueueName>
    </ns:IntegrationEventWS_DeleteEvents_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

GetEvents

Returns events from an integration event queue.

Objects Supported

GetEvents is supported for the following objects: Account, Activity, Allocation, Asset, Business Plan, Campaign, Contact, Contact State License, Custom Object 01 - 03, Custom Object 04 and higher, Dealer, Household, Inventory Period, Lead, MedEvent, Objective, Opportunity, Period, Portfolio, Product, ProductCategory, Sample Inventory, Sample Transaction, Sample Lot, Service Request, Solution, Territory, Transaction Item, Vehicle.

Usage

You use the GetEvents method of the Integration Event Web service to return events from an integration event queue. Integration events are actions that are triggered based on meeting certain workflow criteria. An integration event stores information about data that has changed:

- User key information about the changed record, for example: objectID, externalsystemID
- Audit information, for example, created date, created by, modified date, modified by

Integration events are stored in one or more company queues on the hosted environment. The maximum number of events in the queues is set by Customer Care. Contact Customer Care to request support for the Integration Event Web Service and to specify the maximum total size of the integration queues you require.

For more information about integration events and setting up workflow criteria, see the Oracle CRM On Demand online help.

If the EventCount argument is not supplied, all events are returned.

The QueueName argument, if specified, must match the name of a queue defined for a company within Oracle CRM On Demand. The QueueName value is the localized display name value for the user. If the supplied name is not valid, an error message is returned.

If QueueName is not specified, the value is empty, or if the value is Default Queue, events are returned from the default queue.

When there is a pending request to a queue, another call to the same queue cannot be invoked until the previous request completes. However, another call to a different queue can be invoked before the previous request completes.

You can return events for all supported record types, or a subset of record types, depending on how you prepare the WSDL and Schema files associated with the Integration Event service, see "Preparing the Integration Event WSDL File and Schema Files" on page 157.

In some cases the names of objects in the list of events returned differ from the name of the object used in the UI of Oracle CRM On Demand, as shown in Appendix B, "Mapping of Display Names to Service API Object Names".

Arguments

Table 47 describes the arguments taken by the GetEvents method.

Table 47. Arguments Taken by the GetEvents Method

Name	Description	Required	Default	I/O
EventCount	The maximum number of events to be returned.	No	Not applicable	Input
QueueName	The integration event queue name.	No	Default queue	Input/ Output
ListOfEvent	A list of events	Not applicable	Not applicable	Output
LastEventID	An event ID	Not applicable	Not applicable	Output

Return Value of the Call

A list of the events returned from the integration event queue. Also, the ID of the last event returned.

The QueueName value is returned, if it is specified in the GetEvents request.

For each event in the list of events, there are the following attributes:

- **name**. The name of the associated Workflow.
- **object**. The record type.
- **operation**. The operation performed. The attribute values can be:
  - insert - for items inserted
  - update - for items updated
  - purge - for items that have been purged from the Deleted Items area
  - delete - for items that have been deleted and are still in the Deleted Items area
  - associate - for child items that have been associated with a parent record type
  - dissociate - for child items that have been dissociated from a parent record type

  NOTE: Only the Account, Contact, and Opportunity objects support the associate and dissociate operations. The integration events generated for these objects vary depending on whether the request is made through a Web service request or the UI. For more information, about these differences in integration events, see the information about workflow rules in the online help for Oracle CRM On Demand.

Sample SOAP Request and Responses

This topic contains sample SOAP requests that invoke GetEvents.

Sample SOAP Request 1 - GetEvents with No Arguments

In the following SOAP request, no arguments are passed in (there are no required arguments for GetEvents), all the events from the default queue are retrieved:

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%%USERNAME%%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <IntegrationEventWS_GetEvents_Input xmlns="urn:crmondemand/ws/integrationevent/"></
IntegrationEventWS_GetEvents_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response 1

The following shows the response for sample request 1. The events from the default queue are returned.

```
<?xml version="1.0" encoding="utf-16"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <SOAP-ENV:Body>

    <ns:IntegrationEventWS_GetEvents_Output xmlns:ns="urn:crmondemand/ws/integrationevent/">

      <ListOfEvent xmlns="urn:/crmondemand/xml/integrationevent">

        <Event object="Contact" name="cContact Update" operation="update">

          <SiebelMessage>

            <ListOfContact xmlns="urn:/crmondemand/xml/contact">

              <Contact>

                <ContactId>ADSA-96CRZK</ContactId>

                <ModId>1</ModId>

                <CreatedDate>08/09/2011 10:17:48</CreatedDate>

                <ExternalSystemId></ExternalSystemId>

                <ContactFirstName>Testing</ContactFirstName>
```

```
        <IntegrationId>ADSA-96CRZK</IntegrationId>
        <ContactLastName>U2</ContactLastName>
        <ModifiedById>HRZ780-I5RUE</ModifiedById>
        <ModifiedDate>08/09/2011 10:21:16</ModifiedDate>
        <AlternateAddressId>ADSA-6CRZM</AlternateAddressId>
        <CreatedById>HRZ780-I5RUE</CreatedById>
      </Contact>
    </ListOfContact>
  </SiebelMessage>
</Event>
<Event object="Contact" name="cContact - Send Welcome Package" operation="insert">
  <SiebelMessage>
    <ListOfContact xmlns="urn:/crmondemand/xml/contact">
      <Contact>
        <ContactId>ADSA-96D7AO</ContactId>
        <ModId>0</ModId>
        <AccountId>No Match Row Id</AccountId>
        <CreatedDate>08/09/2011 10:40:52</CreatedDate>
        <ExternalSystemId></ExternalSystemId>
        <ContactFirstName>Testing2</ContactFirstName>
        <IntegrationId>ADSA-96D7AO</IntegrationId>
        <ContactLastName>U1</ContactLastName>
        <ModifiedById>HRZ780-I5RUE</ModifiedById>
        <ModifiedDate>08/09/2011 10:40:53</ModifiedDate>
        <AlternateAddressId>ADSA-96D7AQ</AlternateAddressId>
        <CreatedById>HRZ780-I5RUE</CreatedById>
      </Contact>
    </ListOfContact>
  </SiebelMessage>
</Event>
<Event object="Contact" name="cContact Update" operation="update">
  <SiebelMessage>
    <ListOfContact xmlns="urn:/crmondemand/xml/contact">
      <Contact>
        <ContactId>ADSA-96D7EA</ContactId>
        <ModId>2</ModId>
```

```
            <CreatedDate>08/09/2011 10:45:39</CreatedDate>
            <ExternalSystemId></ExternalSystemId>
            <ContactFirstName>C</ContactFirstName>
            <IntegrationId>ADSA-96D7EA</IntegrationId>
            <ContactLastName>C8</ContactLastName>
            <ModifiedById>HRZ780-I5RUE</ModifiedById>
            <ModifiedDate>08/09/2011 10:57:52</ModifiedDate>
            <AlternateAddressId>ADSA-96D7EC</AlternateAddressId>
            <CreatedById>HRZ780-I5RUE</CreatedById>
          </Contact>
        </ListOfContact>
      </SiebelMessage>
    </Event>
    <Event object="Account" name="Accounts Create" operation="insert">
      <SiebelMessage>
        <ListOfAccount xmlns="urn:/crmondemand/xml/account">
          <Account>
            <AccountId>ADSA-96DJL4</AccountId>
            <ModId>0</ModId>
            <CreatedDate>08/09/2011 11:04:50</CreatedDate>
            <CurrencyCode>USD</CurrencyCode>
            <ExternalSystemId></ExternalSystemId>
            <IntegrationId>ADSA-96DJL4</IntegrationId>
            <Location></Location>
            <ModifiedById>HRZ780-I5RUE</ModifiedById>
            <ModifiedDate>08/09/2011 11:04:51</ModifiedDate>
            <AccountName>A1</AccountName>
            <NumberEmployees>1000</NumberEmployees>
            <PrimaryBillToAddressId>ADSA-96DJL6</PrimaryBillToAddressId>
            <PrimaryShipToAddressId>ADSA-96DJL8</PrimaryShipToAddressId>
            <CreatedById>HRZ780-I5RUE</CreatedById>
          </Account>
        </ListOfAccount>
      </SiebelMessage>
    </Event>
  </ListOfEvent>
```

```
      <ns:LastEventId>20110809160454038_QNMHRZ780-I5RT8_Account_Account_ADSA-96DJL4_0_ADSA-96DCMZ.xml</
ns:LastEventId>

    </ns:IntegrationEventWS_GetEvents_Output>

  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

Sample SOAP Request 2 - GetEvents with EventCount and QueueName

The following SOAP request retrieves one event from the queue High Priority Queue specified by using the <EventCount> and <QueueName> arguments.

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%%USERNAME%%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <IntegrationEventWS_GetEvents_Input xmlns="urn:crmondemand/ws/integrationevent/">

      <QueueName>High Priority Queue</QueueName>

      <EventCount>1</EventCount>

    </IntegrationEventWS_GetEvents_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response 2

The following shows the response for sample SOAP request 2. The events from the named queue are returned.

```
<?xml version="1.0" encoding="utf-16"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <SOAP-ENV:Body>

    <ns:IntegrationEventWS_GetEvents_Output xmlns:ns="urn:crmondemand/ws/integrationevent/">

      <ListOfEvent xmlns="urn:/crmondemand/xml/integrationevent">

        <Event object="Contact" name="cContact Update" operation="update">
```

```
          <SiebelMessage>
            <ListOfContact xmlns="urn:/crmondemand/xml/contact">
              <Contact>
                <ContactId>ADSA-96CRZK</ContactId>
                <ModId>1</ModId>
                <CreatedDate>08/09/2011 10:17:48</CreatedDate>
                <ExternalSystemId></ExternalSystemId>
                <ContactFirstName>Testing</ContactFirstName>
                <IntegrationId>ADSA-96CRZK</IntegrationId>
                <ContactLastName>U2</ContactLastName>
                <ModifiedById>HRZ780-I5RUE</ModifiedById>
                <ModifiedDate>08/09/2011 10:21:16</ModifiedDate>
                <AlternateAddressId>ADSA-96CRZM</AlternateAddressId>
                <CreatedById>HRZ780-I5RUE</CreatedById>
              </Contact>
            </ListOfContact>
          </SiebelMessage>
        </Event>
      </ListOfEvent>
      <ns:LastEventId>20110809152116757_QNMHRZ780-I5RT8_Contact_Contact_ADSA-96CRZK_1_ADSA-96CSA5.xml</
ns:LastEventId>
      <ns:QueueName>Sample Queue</ns:QueueName>
    </ns:IntegrationEventWS_GetEvents_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Preparing the Integration Event WSDL File and Schema Files

You can use the methods of the Integration Events service to track changes for the supported record types. You must download the integrationevent.wsdl file and the XSD file for each record type that you require to track. The integration event WSDL file imports the XSD files for each record type.

The integrationevent.wsdl file is dynamically generated based on which objects have an active workflow rule with an integration event action configured.

For each object with an integration event action configured, a child element of <SiebelMessage> is added to the WSDL files, for example, <ListOfAccount>.

For child objects, an element is added for the parent object, for example, an entry for the Account parent object is added for the Account Team child. For child objects with multiple parents, an element is added for each possibe parent, for example, Account, Contact, and Opportunity parent entries would be added for a Revenue child.

Also, for each object, an <import> element is added to the WSDL file with the correct namespace and schema location for the XSD file for the object. For example:

```
<xsd:import namespace="urn:/crmondemand/xml/account" schemaLocation="Account.xsd" />
```

Either the Web Services v1.0 or Web Services 2.0 namespace is used, depending on the object. You can download XSD Schema files for all objects for the Web Services v1.0 API, and for the Web Services v2.0 objects for which Integration Events support is offered. The correct namespace is generated in the WSDL file accordingly.

To prepare the WSDL

1. Go to the Web Services Administration page in the Oracle CRM On Demand application.
2. Download the Integration Events WSDL.
3. Download the Generic Schema files for objects for which you wish to generate integration events and copy these to the folder where you downloaded the WSDL file.

   NOTE: You can download Custom or Generic Schema files for all objects for the Web Services v1.0 API, and for the Web Services v2.0 API objects for which Integration Events support is offered.

4. If you have created custom fields or renamed fields for an object, download the Custom XSD file for the object.
5. In the integrationevent.wsdl file, remove references to any Schema files for objects you will not be using.
6. Add the integrationevent.wsdl file to your development environment.

NOTE: You must repeat steps 3 through 6 each time a new custom field is added that will be tracked through integration events, or when you begin tracking a new object type. Otherwise, the client application cannot process events containing these newly added fields or objects.

GetLists

Gets the set of lists that the current user has access to for a specified object.

Usage

You use the GetLists method to query the details of the lists for an object.

The ObjectName argument specifies the object to be queried. The names of public and private lists to which the user has access is returned. The set of list names is equivalent to the set of list names visible to the user in the Manage Lists page UI.

ListQueryPage is supported for all objects that support list management.

The list names returned can be used in a QueryPage operation to retrieve the set of records matching the list filter criteria.

Arguments

Table 48 describes the arguments taken by the GetLists method.

Table 48. Arguments Taken by the GetLists Method

Name	Description	Required	Default	I/O
ObjectName	The name of the object.	Yes	Not applicable	Input
ListOfLists	The list of lists returned for the object.	Not applicable	Not applicable	Output

Return Value of the Call

The following information is returned for each list.

- **Name.** The name of the list.
- **ListType.** Whether the list is a public list, or a private list created by the current user.

Sample SOAP Request - GetLists

The following SOAP request retrieves the set of lists visible to the authenticated user for the account record type as specified in the <ObjectName> argument:

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%%USERNAME%%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <ListsWS_GetLists_Input xmlns="urn:crmondemand/ws/lists/">

      <ObjectName>Account</ObjectName>

    </ListsWS_GetLists_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response

The following response contains the lists for the account object:

```
<?xml version="1.0" encoding="utf-16"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <SOAP-ENV:Body>

    <ns:ListsWS_GetLists_Output xmlns:ns="urn:crmondemand/ws/lists/">

      <ListofLists xmlns="urn:/crmondemand/xml/lists">

        <List>

          <Name>All Account Competitor Accounts</Name>

          <ListType>System</ListType>

        </List>

        <List>

          <Name>All Account Partner Accounts</Name>

          <ListType>System</ListType>

        </List>

        <List>

          <Name>All Accounts</Name>

          <ListType>System</ListType>

        </List>

        <List>

          <Name>All Customer Accounts</Name>

          <ListType>System</ListType>

        </List>

        <List>

          <Name>All Prospect Accounts</Name>

          <ListType>System</ListType>

        </List>

        <List>

          <Name>All Referenceable Accounts</Name>

          <ListType>System</ListType>

        </List>

        <List>

          <Name>All Top Accounts</Name>

          <ListType>System</ListType>

        </List>

        <List>
```

```
          <Name>My Accounts</Name>
          <ListType>System</ListType>
        </List>
        <List>
          <Name>My Recently Created Accounts</Name>
          <ListType>System</ListType>
        </List>
        <List>
          <Name>My Recently Modified Accounts</Name>
          <ListType>System</ListType>
        </List>
        <List>
          <Name>Recently Created Accounts</Name>
          <ListType>System</ListType>
        </List>
        <List>
          <Name>Recently Modified Accounts</Name>
          <ListType>System</ListType>
        </List>
      </ListofLists>
    </ns:ListsWS_GetLists_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

GetMapping

Returns the display names and XML tags of the fields of a record type or one of its child components.

Objects Supported

GetMapping is supported for all objects accessible through Web Services V1.0 and Web Services 2.0 and all of their child components.

Usage

You use the GetMapping method to return the mapping between the display names of all the fields in a particular record type and the XML tags for each field.

In some cases, the names of objects used in the ObjectName argument differ from the name of the object used in the UI of Oracle CRM On Demand, as shown in Appendix B, “Mapping of Display Names to Service API Object Names”.

The GetMapping method works with all Custom Objects. However, you must use a different naming convention for Custom Objects 1-3, compared to Custom Object 4 and higher, where there are no spaces in the object name. SOAP requests must use the following naming convention for object names:

- **Custom Object 1-3**. Custom Object 1, Custom Object 2, Custom Object 3
- **Custom Object 4 and higher**. CustomObject4...CustomObject14, and so on

For some fields, the DisplayName returned has the value hidden. These fields are available in the WSDL file for the object, but are not available in the Oracle CRM On Demand UI. Examples of fields for which the value hidden is returned include the LanguageCode, LocaleCode, RoleId, and TimeZoneId fields of the User object.

Arguments

Table 49 describes the arguments taken by the GetMapping method.

Table 49. Arguments Taken by the GetMapping Method

Name	Description	Required	Default	I/O
ObjectName	The name of the record type for which you wish to return a list of mappings.	Yes	Not applicable	Input/ Output

Return Value of the Call

A list of the display names for fields and their associated XML mappings:

- **LastUpdated**. The date the field was last updated.
- **DisplayName**. The display name of the field, in the user’s language. For some fields not available in the UI, the value hidden is returned.
- **ElementName**. The XML element name for the field.
- **DataType**. The field type of the field, for example, Check box, Picklist, and so on.

Sample SOAP Request - GetMapping

The following SOAP request retrieves the display names and XML element names of the fields of the book record type as specified in the <ObjectName> argument:

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>
```

```
    <wsse:Security>
      <wsse:UsernameToken>
        <wsse:Username>%%%USERNAME%%%</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <MappingWS_GetMapping_Input xmlns="urn:crmondemand/ws/mapping/">
      <ObjectName>Book</ObjectName>
    </MappingWS_GetMapping_Input>
  </soap:Body>
</soap:Envelope>
```

Sample SOAP Response

The following response contains the mappings for a number of fields for the book object:

```
<?xml version="1.0" encoding="utf-16"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns:MappingWS_GetMapping_Output xmlns:ns="urn:crmondemand/ws/mapping/">
      <ns:ObjectName>Book</ns:ObjectName>
      <ListOfField xmlns="urn:/crmondemand/xml/mappingservice">
        <Field>
          <DisplayName>Can Contain Data</DisplayName>
          <ElementName>CanContainDataFlag</ElementName>
          <DataType>Checkbox</DataType>
        </Field>
        <Field>
          <DisplayName>Id</DisplayName>
          <ElementName>BookId</ElementName>
          <DataType>ID</DataType>
        </Field>
        <Field>
          <DisplayName>Parent Book</DisplayName>
          <ElementName>ParentBookName</ElementName>
```

```
          <DataType>Picklist</DataType>
        </Field>
        <Field>
          <DisplayName>Created</DisplayName>
          <ElementName>CreatedBy</ElementName>
          <DataType>Text (Long)</DataType>
        </Field>
        <Field>
          <DisplayName>Book Type</DisplayName>
          <ElementName>BookType</ElementName>
          <DataType>Picklist</DataType>
        </Field>
        <Field>
          <DisplayName>Modified: Date</DisplayName>
          <ElementName>ModifiedDate</ElementName>
          <DataType>Date/Time</DataType>
        </Field>
        <Field>
          <DisplayName>Created: Date</DisplayName>
          <ElementName>CreatedDate</ElementName>
          <DataType>Date/Time</DataType>
        </Field>
        <Field>
          <DisplayName>Description</DisplayName>
          <ElementName>Description</ElementName>
          <DataType>Text (Long)</DataType>
        </Field>
        <Field>
          <DisplayName>Book Name</DisplayName>
          <ElementName>BookName</ElementName>
          <DataType>Text (Short)</DataType>
        </Field>
      </ListOfField>
    </ns:MappingWS_GetMapping_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

GetPicklistValues

Gets picklist values from Oracle CRM On Demand.

Objects Supported

GetPicklistValues is supported for all Web Services v1.0 and v2.0 accessible parent-level objects.

Usage

You use the GetPicklistValues method to enable external applications to present lists of values to users, typically in a language-dependent manner. The method can get lists of possible values for both cascading and regular picklist fields.

Because On Demand Web Services is language-independent, it is the client application's responsibility to convert code from the language-independent code (LIC) used by Oracle CRM On Demand to language-dependent values (LDVs) typically used by the external presentation layer.

The returned list of values corresponds to the organization to which the current user belongs (that is, the user whose credentials have been passed during the log-in call).

Cascading picklists restrict the values of one picklist, the related picklist, based on the value selected in another picklist, the parent picklist. For example, a parent picklist might present a list of IT areas and drive the value of a related picklist called SubAreas. When the user selects, for example, the value Installation for Area, the SubAreas picklist is dynamically constrained to show only the picklist values that are associated with the Installation area, for example, Server Crash and No Admin Login.

If the provided picklist has a parent, only the values that have a parent are returned. When a picklist has a parent, the result set includes the parent and the child values and at the end includes an empty set that contains all values available for the requested picklist.

If a picklist is not cascading, the following elements are returned empty:

- ParentFieldName
- ParentDisplayValue
- ParentCode

If a "10/2004" namespace is used, the FieldName and ParentFieldName elements respectively accept and return the integration tag value for custom fields, otherwise, they accept and return the generic custom field tag names (that is, CustomPicklist1 and so on).

Arguments

Table 50 describes the arguments taken by the GetPicklistValues method.

Table 50. Arguments Taken by the GetPicklistValues Method

Name	Description	Required	Default	I/O
RecordType	The record type; this is case insensitive	Yes	Not applicable	Input
FieldName	The name of the picklist field.	Yes	Not applicable	Input
LanguageCode	The code of the language in which language-dependent values are to be returned, for example, ENU, DEU, FRA, ESN, and so on. If the code is not specified, the default language for the current session's user is used.	No	*User's Default Language*	Input
ListOfParentPicklistValue	A sequence of ParentPicklistValue elements.	Yes	Not applicable	Output

Return Value of the Call

A list of picklist values. For a cascading picklist, this includes the values for the related picklist that apply for particular values of the parent picklist. For a regular picklist, values for parent picklist are not included.

The ParentPicklistValue element contains the following child elements:

- **Language**. The language.
- **ParentFieldName**. The parent picklist field name as an integration tag.
- **ParentDisplayValue**. A display value translated into the specified language.
- **ParentCode**. A parent Language Independent Code (LIC).
- **ListOfPickListValue**. A sequence of PicklistValue elements containing the related picklist values that correspond to the parent picklist value.

The PicklistValue element contains the following child elements:

- **DisplayValue**. The display value translated into the specified language.
- **Code**. The Language Independent Code (LIC).

Sample SOAP Request - GetPicklist

The following SOAP request retrieves the picklist values in French for the Priority picklist of the account record type.

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>
```

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%%USERNAME%%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <PicklistWS_GetPicklistValues_Input xmlns="urn:crmondemand/ws/picklist/">

      <RecordType>Account</RecordType>

      <FieldName>Priority</FieldName>

      <LanguageCode>FRA</LanguageCode>

    </PicklistWS_GetPicklistValues_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response

The following response contains values for the Priority picklist of Account, including the French language display values:

```
<?xml version="1.0" encoding="utf-16"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <SOAP-ENV:Body>

    <ns:PicklistWS_GetPicklistValues_Output xmlns:ns="urn:crmondemand/ws/picklist/">

      <ListOfParentPicklistValue xmlns="urn:/crmondemand/xml/picklist">

        <ParentPicklistValue>

          <Language>FRA</Language>

          <ParentFieldName></ParentFieldName>

          <ParentDisplayValue></ParentDisplayValue>

          <ParentCode></ParentCode>

          <Disabled></Disabled>

          <ListOfPicklistValue>

            <PicklistValue>
```

```
              <Code>Low</Code>
              <DisplayValue>Faible</DisplayValue>
              <Disabled>N</Disabled>
            </PicklistValue>
            <PicklistValue>
              <Code>Medium</Code>
              <DisplayValue>Moyen</DisplayValue>
              <Disabled>N</Disabled>
            </PicklistValue>
            <PicklistValue>
              <Code>High</Code>
              <DisplayValue>Elevé</DisplayValue>
              <Disabled>N</Disabled>
            </PicklistValue>
          </ListOfPicklistValue>
        </ParentPicklistValue>
      </ListOfParentPicklistValue>
    </ns:PicklistWS_GetPicklistValues_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

GetServerTime

Returns the time from a server.

Usage

You use the GetServerTime method to get the time at the server involved in a Web services API session. The time returned is converted to the time for the locale of the user ID making the request.

Return Value of the Call

The current server time.

Sample SOAP Request - GetServerTime

The following SOAP request retrieves the server time. No arguments are required.

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>
```

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%%USERNAME%%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <TimeWS_GetServerTime_Input xmlns="urn:crmondemand/ws/time/"></TimeWS_GetServerTime_Input>

  </soap:Body>

</soap:Envelope>
```


Sample SOAP Response

The following response contains the time:


```
<?xml version="1.0" encoding="utf-16"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <SOAP-ENV:Body>

    <ns:TimeWS_GetServerTime_Output xmlns:ns="urn:crmondemand/ws/time/">

      <ns:CurrentServerTime>08/05/2011 10:54:27</ns:CurrentServerTime>

      <ns:TimeZone>(GMT-06:00) Central Time (US &amp; Canada)</ns:TimeZone>

    </ns:TimeWS_GetServerTime_Output>

  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```


LoginHistoryQueryPage

Executes a query against the list of user login history, and returns a subset of the records that match the search criteria set by the method arguments.

Usage

You use the LoginHistoryQueryPage method to view the Login History for a user. This is the same information that is displayed in the Company Administration, Sign In Audit page in the Oracle CRM On Demand application. As an example of how you might use this data, you might save the data in a CSV file and then import it into a spreadsheet. You could then use the spreadsheet to generate a report showing, for example, how often a user logs into Oracle CRM On Demand.

Arguments

Table 51 describes the arguments taken by the LoginHistoryQueryPage method.

Table 51. Arguments Taken by the LoginHistoryQueryPage Method

Name	Description	Required	Default	I/O
ListOfLoginHistory	The list of object instances queried (input), and after query execution, the list of object instances returned (output).	Yes	Not applicable	Input/ Output
PageSize	The maximum number of records displayed on a page following a query.	No	10	Input
StartRowNum	Indicates the row from which the LoginHistoryQueryPage method starts to return records. Use the StartRowNum argument to return a set of records for any given method. For example, if you want to return records 1-100, you set StartRowNum to 0. Then, if you want to return records 101-200, you set StartRowNum to 100, and run the query again. You continue doing this until the last page is returned. In this way, you can return all records for a particular query.	No	0	Input
LastPage	A value that indicates whether or not the last value in the query set has been returned.	Not applicable	Not applicable	Output

Return Value of the Call

The following information is returned for each usage record.

A list of user login history. The following are the child elements of ListOfLoginHistory:

- **UserId.** The user ID of the user.

- **FirstName.** The user's first name.
- **LastName.** The user's last name.
- **UserAlias.** The user alias of the user.
- **LoginName.** The login name for the user.
- **LoginStatus.** The login status for the user.
- **LoginTimestamp.** The time at which the user last logged in.
- **ClientType.** The type of client from which the user logged in.
- **IPAddress.** The source IP address for the user.
- **AdditionalInformation.** Additional information for the user.

Sample SOAP Request - LoginHistoryQueryPage

The following SOAP request queries against the list of user login history to return a login record that occurred after the timestamp of 08/01/2011 00:00:00. It also demonstrates the use of the optional <PageSize> and <StartRowNum> arguments to return one record at row 1 (second record).

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%%USERNAME%%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <LoginHistoryWS_LoginHistoryQueryPage_Input xmlns="urn:crmondemand/ws/loginhistory/">

      <ListOfLoginHistory>

        <LoginHistory>

          <UserId />

          <FirstName />

          <LastName />

          <UserAlias />

          <LoginName />

          <LoginStatus />
```

```
          <LoginTimestamp>&gt;='08/01/2011 00:00:00'</LoginTimestamp>
          <ClientType />
          <IPAddress />
          <AdditionalInformation />
        </LoginHistory>
      </ListOfLoginHistory>
      <PageSize>1</PageSize>
      <StartRowNum>1</StartRowNum>
    </LoginHistoryWS_LoginHistoryQueryPage_Input>
  </soap:Body>
</soap:Envelope>
```

Sample SOAP Response

The following response contains the requested login history:

```
<?xml version="1.0" encoding="utf-16"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns:LoginHistoryWS_LoginHistoryQueryPage_Output xmlns:ns="urn:crmondemand/ws/loginhistory/">
      <ns:LastPage>false</ns:LastPage>
      <ListOfLoginHistory xmlns="urn:/crmondemand/xml/loginhistory">
        <LoginHistory>
          <UserId>HRZ780-I5RUE</UserId>
          <FirstName>Joanne</FirstName>
          <LastName>Brown</LastName>
          <UserAlias>Joanne Brown</UserAlias>
          <LoginName>%%%USERNAME%%%</LoginName>
          <LoginStatus>Success</LoginStatus>
          <LoginTimestamp>08/02/2011 13:53:00</LoginTimestamp>
          <ClientType>WS</ClientType>
          <IPAddress>10.156.87.45</IPAddress>
          <AdditionalInformation></AdditionalInformation>
        </LoginHistory>
      </ListOfLoginHistory>
    </ns:LoginHistoryWS_LoginHistoryQueryPage_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

MergeRecords

Merges records for certain record types.

Objects Supported

MergeRecords is only supported for the Account, Contact, Household, Lead, and Portfolio record types. If an invalid record type is provided, an error message is displayed.

Usage

You use the MergeRecords method to merge records. When you merge two records, you specify the record that you want to keep, which is called the *primary record*, and the record that is to be deleted, which is called the *duplicate record*.

You must identify a reference to a primary record and a reference to a duplicate record in the request. To identify the primary record, you must specify either a PrimaryId or a PrimaryExternalSystemId argument. To identify the duplicate record, you must specify either a DuplicateId or a DuplicateExternalSystemId argument.

The following rules apply to merging records:

- Fields in the primary parent record that contain data are retained.
- Fields in the primary record that are blank get the value from the duplicate record, if it has a value and if the MergeWhenPrimaryBlank argument is set to true.
- Fields in the primary parent record that are blank remain blank, if the MergeWhenPrimaryBlank argument is not set, or is set to a value other than true.

The Merge Web service has the same security restrictions as in the Oracle CRM On Demand UI regarding privilege and record permissions.

Table 52 describes the arguments taken by MergeRecords.

Table 52. Arguments taken by MergeRecords

Field Name	Description	Required	Default	I/O
PrimaryId	The ID of the primary record.	One of these arguments is required	Not applicable	Input
PrimaryExternalSystemId	The externalsystemID of the primary record.		Not applicable	Input
DuplicateId	The ID of the duplicate record.	One of these arguments is required	Not applicable	Input
DuplicateExternalSystemId	The externalsystemID of the duplicate record.		Not applicable	Input

Table 52. Arguments taken by MergeRecords

Field Name	Description	Required	Default	I/O
MergeWhenPrimaryBlank	Determines how records are merged when fields in the primary record are blank. True values are set as Y, Yes, True, or 1. False values are any other values including blanks.	No	False	Input
RecordType	The record type; this is case sensitive.	Yes	Not applicable	Input

Return Value of the Call

The following four values are returned:

- **MergedRecordId**. The ID of the merged record, that is, the primary record.
- **MergedRecordExternalId**. The externalsystemID of the merged record.
- **DeletedRecordId**. The ID of the deleted record, that is, the duplicate record.
- **DeletedRecordExternalId**. The externalsystemID of the merged record.

Sample SOAP Request - MergeRecords

The following SOAP request merges a contact record with DuplicateExternalSystemId = abc223 into a contact record with Id = ADSA-93DZIC.

NOTE: To identify the primary record, you can specify either the <PrimaryId> or <PrimaryExternalSystemId> argument, and to specify the duplicate record you can use either the <DuplicateId> or <DuplicateExternalSystemId> argument.

The sample also demonstrates passing in the optional <MergeWhenPrimaryBlank> argument to copy the value for a field in the duplicate record to the primary record when the field in the primary record is blank.

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%%USERNAME%%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>
```

```
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <MergeRecordsWS_MergeRecords_Input xmlns="urn:crmondemand/ws/mergerecords/">
      <PrimaryId>ADSA-93DZIC</PrimaryId>
      <DuplicateExternalSystemId>abc223</DuplicateExternalSystemId>
      <MergeWhenPrimaryBlank>Y</MergeWhenPrimaryBlank>
      <RecordType>Contact</RecordType>
    </MergeRecordsWS_MergeRecords_Input>
  </soap:Body>
</soap:Envelope>
```

Sample SOAP Response

The following response contains the IDs of the merged and deleted records:

```
<?xml version="1.0" encoding="utf-16"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns:MergeRecordsWS_MergeRecords_Output xmlns:ns="urn:crmondemand/ws/mergerecords/">
      <ns:DeletedRecordExternalSystemId>abc223</ns:DeletedRecordExternalSystemId>
      <ns:DeletedRecordId>ADSA-93ET5I</ns:DeletedRecordId>
      <ns:MergedRecordExternalSystemId>ABC123</ns:MergedRecordExternalSystemId>
      <ns:MergedRecordId>ADSA-93DZIC</ns:MergedRecordId>
    </ns:MergeRecordsWS_MergeRecords_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

MetadataChangeSummaryQueryPage

Gets a summary of changes to various categories of metadata.

Usage

You use the MetadataChangeSummaryQueryPage method to determine whether there have been any metadata changes in Oracle CRM On Demand. Only the relevant changes can be extracted and applied to client applications.

With the Administration Services API, field and picklist data, for example, can be accessed and modified (see, for example, "FieldManagementUpsert" on page 225 and "PicklistUpsert" on page 237).

Arguments

Table 53 describes the arguments taken by the MetadataChangeSummaryQueryPage method.

Table 53. Arguments Taken by the MetadataChangeSummaryQueryPage Method

Name	Description	Required	Default	I/O
ListOfMetadataChange Summary	The metadata to be queried and returned.	Yes	Not applicable	Input/ Output
recordcountneeded	Indicates whether a record count for the query is to be returned.	No	false	Input
PageSize	The maximum number of records displayed on a page following a query.	No	10	Input
StartRowNum	Indicates the row from which the method starts to return records.	No	0	Input
searchspec	Indicates the search specification.	No	Not applicable	Input
sortsequence	An integer value that determines the order of the sort specification.	No		Input
sortorder	Determines the sort order for the records returned by the query, either ASC for ascending or DESC for descending.	No	ASC	Input
LastPage	A value that indicates whether or not the last value in the query set has been returned.	Not applicable	Not applicable	Output
recordcount	An integer value that indicates the record count.	Not applicable	Not applicable	Output

Return Value of the Call

The following information is returned:

- **LOVLastUpdated**. The date and time of the last picklist update.
- **CascPicklistsLastUpdated**. The date and time of the last cascading picklist update.
- **FieldManagementLastUpdated**. The date and time of the last field management data update.
- **WorkflowLastUpdated**. The date and time of the last workflow update.
- **AccessProfileLastUpdated**. The date and time of the last access profile update.

- **BookOfBusinessLastUpdated**. The date and time of the last book update.
- **ThemesLastUpdated**. The date and time of the last theme update.

Sample SOAP Request - MetadataChangesSummaryQueryPage

The following SOAP request retrieves the summary of changes to various categories of metadata:

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%%USERNAME%%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <MetadataChangeSummaryQueryPage_Input xmlns="urn:crmondemand/ws/metadatachangesummary/">

      <ListOfMetadataChangeSummary>

        <MetadataChangeSummary>

          <LOVLastUpdated />

          <CascPicklistsLastUpdated />

          <FieldManagementLastUpdated />

          <WorkflowLastUpdated />

          <AccessProfileLastUpdated />

          <BookOfBusinessLastUpdated />

          <ThemesLastUpdated />

        </MetadataChangeSummary>

      </ListOfMetadataChangeSummary>

    </MetadataChangeSummaryQueryPage_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response

The following response contains the returned metadata:

```
<?xml version="1.0" encoding="utf-16"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns:MetadataChangeSummaryQueryPage_Output xmlns:ns="urn:crmondemand/ws/metadatachangesummary/">
     <ListOfMetadataChangeSummary xmlns="urn:/crmondemand/xml/metadatachangesummary/Data" lastpage="true">
        <MetadataChangeSummary>
          <LOVLastUpdated>2011-08-04T14:46:19</LOVLastUpdated>
          <CascPicklistsLastUpdated>2011-08-04T15:06:08</CascPicklistsLastUpdated>
          <FieldManagementLastUpdated>2011-08-03T11:36:21</FieldManagementLastUpdated>
          <WorkflowLastUpdated>2016-01-29T11:32:09</WorkflowLastUpdated>
          <AccessProfileLastUpdated>2016-12-01T20:25:14</AccessProfileLastUpdated>
          <BookOfBusinessLastUpdated>2016-12-19T11:43:06</BookOfBusinessLastUpdated>
        </MetadataChangeSummary>
      </ListOfMetadataChangeSummary>
    </ns:MetadataChangeSummaryQueryPage_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


SalesProcessQueryPage

Returns sales process information for a particular user, the default sales process, or all sales processes for a company.

Usage

You use the SalesProcessQueryPage method to retrieve sales process information including sales stages, order of the sales stages, default probability, and associated opportunity types.

The ListOfSalesProcess argument contains elements for the various fields of a sales process, and for the associated sales stages and opportunity types. There are child elements ListofSalesStage and ListofOpportunityType.

In requests, if the <SalesProcessId> element does not contain a value, information for all of the sales processes for the company are returned.

If the <SalesProcessId> element contains a value, information for sales processes for the user or role with that sales process ID are returned. You can use the User service to query for the SalesProcId value for a user.

If the <Default> element contains the value Y, information for the default sales process is returned.

For most of the child elements of ListOfSalesProcess, you can use the sortorder and sortsequence attributes to specify how fields in the returned sales process information are sorted.

NOTE: The Name and Description fields of SalesProcess do not support the sort specification.

Arguments

Table 54 describes the arguments taken by the SalesProcessQueryPage method.

Table 54. Arguments Taken by the SalesProcessQueryPage Method

Name	Description	Required	Default	I/O
ListOfSalesProcess	The list of sales processes queried (input), and after query execution, the list of sales processes returned (output).	Yes	Not applicable	Input/ Output
PageSize	The maximum number of records displayed on a page following a query.	No	10	Input
StartRowNum	Indicates the row from which the SalesProcessQueryPage method starts to return records. For example, if you want to return records 1-100, you set StartRowNum to 0. Then, if you want to return records 101-200, you set StartRowNum to 100, and run the query again. You continue doing this until the last page is returned. In this way, you can return all records for a particular query.	No	0	Input
recordcountneeded	Indicates whether a record count for the object is to be returned.	No	False	Input
searchspec	Indicates the search specification for a field or for all the fields of an object.	No	Not applicable	Input
sortsequence	An integer value that determines the order of the sort specification.	No	Not applicable	Input
sortorder	Determines the sort order for the records returned by the query, either ASC for ascending or DESC for descending.	No	ASC	Input
LOVLanguageMode	The language mode for picklists, see "The LOVLanguageMode Argument" on page 104.	No	LIC	Input

Table 54. Arguments Taken by the SalesProcessQueryPage Method

Name	Description	Required	Default	I/O
LastPage	A value that indicates whether or not the last value in the query set has been returned.	Not applicable	Not applicable	Output
recordcount	An integer value that indicates the record count for the object	Not applicable	Not applicable	Output

Return Value of the Call

The following information is returned by ListOfsalesProcess which contains a <SalesProcess> element for each sales process that matches the query:

- **Description.** The description of the sales process.
- **Name.** The sales process name.
- **Translate**. Whether the sales process is marked for translation.
- **SalesProcessId.** The sales process ID.
- **Default.** Whether this sales process is the default sales process.
- **ListOfOpportunityType**. Information contained in the following child elements for each opportunity type:
  - **Id**. The opportunity type ID.
  - **Type**. The opportunity type name.
- **ListOfSalesStage**. Information contained in the following child elements for each sales stage:
  - **Name**. The name of the sales stage.
  - **SalesCategoryName**. The category that the sales stage falls under.
  - **Order.** The sequence of the sales stages as they appear in the picklist.
  - **Probability**. The default probability for the sales stage.
  - **Description**. The description of the sales stage.
  - **TranslationLanguage**. The translation language.

For more information about sales process information, see Oracle CRM On Demand online help.

Sample SOAP Requests and Responses

This topic contains sample SOAP requests that invoke SalesProcessQueryPage.

Sample SOAP Request 1 - SalesProcessQueryPage with pagesize, startrownum and recordcountneeded Attributes

The following SOAP request retrieves the sales processes defined for the company but only returns a maximum of one record at the SalesProcess parent object level, a maximum of one record at the OpportunityType child object level, and a maximum of five records at the SalesStage level by specifying pagesize, startrownum, and recordcountneeded attributes at each of the list of objects levels.

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%%USERNAME%%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <SalesProcessQueryPage_Input xmlns="urn:crmondemand/ws/salesproc/">

      <ListOfSalesProcess pagesize="1" startrownum="1" recordcountneeded="true">

        <SalesProcess>

          <Id />

          <Name />

          <Description />

          <Default />

          <ListOfOpportunityType pagesize="1" startrownum="0" recordcountneeded="false">

            <OpportunityType>

              <Id />

              <Type />

            </OpportunityType>

          </ListOfOpportunityType>

          <ListOfSalesStage pagesize="5" startrownum="0" recordcountneeded="true">

            <SalesStage>

              <Name />

            </SalesStage>

          </ListOfSalesStage>
```

```
        </SalesProcess>

      </ListOfSalesProcess>

    </SalesProcessQueryPage_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response 1

The following is the response for sample SOAP request 1:

```
<?xml version="1.0" encoding="utf-16"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <SOAP-ENV:Body>

    <ns:SalesProcessQueryPage_Output xmlns:ns="urn:crmondemand/ws/salesproc/">

      <ListOfSalesProcess xmlns="urn:/crmondemand/xml/salesprocess/Data" recordcount="2" lastpage="true">

        <SalesProcess>

          <Id>HRZ780-IDIT2</Id>

          <Name>Renewal Sales Process</Name>

          <Default>N</Default>

          <Description>Simplified process for renewal sales.</Description>

          <ListOfOpportunityType lastpage="true">

            <OpportunityType>

              <Id>HRZ780-IDIV0</Id>

              <Type>Renewal</Type>

            </OpportunityType>

          </ListOfOpportunityType>

          <ListOfSalesStage recordcount="4" lastpage="true">

            <SalesStage>

              <Name>Re-Engage</Name>

            </SalesStage>

            <SalesStage>

              <Name>Proposal</Name>

            </SalesStage>

            <SalesStage>

              <Name>Closed/Won</Name>

            </SalesStage>

            <SalesStage>

              <Name>Closed/Lost</Name>
```

```
            </SalesStage>
          </ListOfSalesStage>
        </SalesProcess>
      </ListOfSalesProcess>
    </ns:SalesProcessQueryPage_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample SOAP Request 2 - SalesProcessQueryPage with searchspec, sortorder, and sortsequence Attributes

The following SOAP request queries the sales process by specifying the search criteria in the searchspec attribute. It returns all sales processes with a name like `Default*` and all child sales stages of the parent sales process with a name like `Closed*`. It also demonstrates using the sortorder and sortsequence attributes to sort the result set.

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <wsse:Security>
      <wsse:UsernameToken>
        <wsse:Username>%%%USERNAME%%%</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <SalesProcessQueryPage_Input xmlns="urn:crmondemand/ws/salesproc/">
      <ListOfSalesProcess recordcountneeded="true">
        <SalesProcess searchspec="[Name] LIKE 'Default*'">
          <Id />
          <Name />
          <Description />
          <Default />
          <ListOfOpportunityType>
            <OpportunityType>
              <Id />
              <Type />
```

```
            </OpportunityType>
          </ListOfOpportunityType>
          <ListOfSalesStage recordcountneeded="true">
            <SalesStage searchspec="[Name] LIKE 'Closed*'">
              <Name sortorder="ASC" sortsequence="0"></Name>
            </SalesStage>
          </ListOfSalesStage>
        </SalesProcess>
      </ListOfSalesProcess>
    </SalesProcessQueryPage_Input>
  </soap:Body>
</soap:Envelope>
```

Sample SOAP Response 2

The following is the response for sample SOAP request 2:

```
<?xml version="1.0" encoding="utf-16"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns:SalesProcessQueryPage_Output xmlns:ns="urn:crmondemand/ws/salesproc/">
      <ListOfSalesProcess xmlns="urn:/crmondemand/xml/salesprocess/Data" recordcount="1" lastpage="true">
        <SalesProcess>
          <Id>HRZ780-I5RS0</Id>
          <Name>Default Sales Process</Name>
          <Default>Y</Default>
          <Description>This sales process is the default defined for your company.</Description>
          <ListOfOpportunityType lastpage="true">
            <OpportunityType>
              <Id>HRZ780-IDIVS</Id>
              <Type>New Business</Type>
            </OpportunityType>
          </ListOfOpportunityType>
          <ListOfSalesStage recordcount="2" lastpage="true">
            <SalesStage>
              <Name>Closed/Lost</Name>
            </SalesStage>
            <SalesStage>
```

```
                <Name>Closed/Won</Name>
              </SalesStage>
            </ListOfSalesStage>
          </SalesProcess>
        </ListOfSalesProcess>
      </ns:SalesProcessQueryPage_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SetPassword

Allows the system administrator to set the passwords of users who use Oracle CRM On Demand.

Usage

You use the SetPassword method to enable external applications to synchronize user passwords. For security reasons the password API is not available by default. If customers want to use SetPassword, they can call Customer Care to have the functionality enabled. There are two privileges:

- **Change Company Passwords using Web Services**. Enables access to SetPassword.
- **Manage Company Password Access.** Users (usually administrators) who have been granted this privilege can grant other roles the Change Company Passwords using Web Services privilege.

The API allows for the setting of passwords for one or more users at the same time. For each password that is updated, a corresponding user Audit Trail record is created. A user with the ability to set passwords does not have the ability to update the password of another user who has the ability to set passwords.

Arguments

Table 55 describes the arguments taken by SetPassword. These are child elements of <ListofUser>.

Allows the system administrator to set the passwords of users who use Oracle CRM On Demand.

Table 55. Arguments Taken by SetPassword

Field Name	Description	Required	Default	I/O
UserId	The row ID value of the user record.	Yes	Not applicable	Input/Output
EmailAddr	The user's email address	No	Not applicable	Input/Output
ExternalSystemId	The external system Id for the user.	No	Not applicable	Input/Output

Table 55. Arguments Taken by SetPassword

Field Name	Description	Required	Default	I/O
IntegrationId	The integration Id	No	Not applicable	Input/Output
Password	The password for the user.	Yes	Not applicable	Input/Output

Sample SOAP Request - SetPassword

The following SOAP request updates the password for the users in the request.

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%%USERNAME%%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <PasswordWS_SetPassword_Input xmlns="urn:crmondemand/ws/password/">

      <ListOfUser>

        <User>

          <UserId>HRZ780-KGY67</UserId>

          <EMailAddr />

          <Password>oracletest123</Password>

          <IntegrationId />

          <ExternalSystemId />

        </User>

      </ListOfUser>

    </PasswordWS_SetPassword_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response

The following response contains the status of the request:

```
<?xml version="1.0" encoding="utf-16"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <SOAP-ENV:Body>

    <ns:PasswordWS_SetPassword_Output xmlns:ns="urn:crmondemand/ws/password/">

      <ns:Status>Success</ns:Status>

    </ns:PasswordWS_SetPassword_Output>

  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

SetSessionTimeZone

Sets the time zone for a Web Services API session.

Usage

This method sets the time zone for a Web services API session. The time zone is set according to the locale of the user making the request.

Arguments

Table 56 describes the arguments taken by the SetSessionTimeZone method.

Table 56. Arguments Taken by the SetSessionTimeZone Method

Name	Description	Required	Default	I/O
TimeZone	The time zone of the user.	Yes	Not applicable	Input
CurrentServerTime	The server time zone.	Not applicable	Not applicable	Output

Return Value of the Call

The current server time zone.

Sample SOAP Request - SetSessionTimeZone

The following SOAP request updates the timezone for the current session to (GMT -06:00) Central Time (US & Canada).

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>
```

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%%USERNAME%%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <TimeWS_SetSessionTimeZone_Input xmlns="urn:crmondemand/ws/time/">

      <TimeZone>(GMT-06:00) Central Time (US &amp; Canada)</TimeZone>

    </TimeWS_SetSessionTimeZone_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response

The following response returns the server time zone set by the request:

```
<?xml version="1.0" encoding="utf-16"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <SOAP-ENV:Body>

    <ns:TimeWS_SetSessionTimeZone_Output xmlns:ns="urn:crmondemand/ws/time/">

      <ns:CurrentServerTime>(GMT-06:00) Central Time (US &amp; Canada)</ns:CurrentServerTime>

    </ns:TimeWS_SetSessionTimeZone_Output>

  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

UpdatePicklist

Updates picklist values in Oracle CRM On Demand.

Objects Supported

UpdatePicklist is supported for the same objects as supported by the GetPicklistValues method, see "GetPicklistValues" on page 165.

Usage

You use the UpdatePicklist method to update lists of values, typically in a language-dependent manner.

The ListOfPicklistValues argument contains a sequence of PicklistValues elements, each of which has the following child elements:

- **DisplayValue**. The display value in the specified language.
- **Code**. The Language Independent Code (LIC).
- **Order.** The order of the value in the list.
- **NeedTranslate.** Whether the value is needed for translation.
- **Disabled.** Whether the value is disabled.

If the CreateNew argument is set to Y, new picklist values are created with the specified values.

If the CreateNew argument is set to Y, new picklist values are created with the specified values from DisplayValue and Order, which are required values.

If the CreateNew argument is set to N, existing picklist values are updated with the specified values from DisplayValue or Code, as long as these values are valid.

You cannot use the UpdatePicklist method to create custom picklists or multiselect picklists. You must create new picklist fields through the Oracle CRM On Demand application UI.

You cannot update read-only picklist fields.

Arguments

Table 57 describes the arguments taken by the UpdatePicklist method.

Table 57. Arguments Taken by the UpdatePicklist Method

Name	Description	Required	Default	I/O
RecordType	The record type; this is case insensitive	Yes	Not applicable	Input
FieldName	The name of the picklist field.	Yes	Not applicable	Input
CreateNew	Whether new picklist values are to be created. A value of Y specifies that new picklist values are to be created, the default value of N specifies that picklist values are to be updated.	No	N	Input
ListOfPicklistValue	A sequence of PicklistValue elements containing the picklist values to be updated or added.	Yes	Not applicable	Input

Table 57. Arguments Taken by the UpdatePicklist Method

Name	Description	Required	Default	I/O
LanguageCode	The language code for the picklist values to be updated, for example, ENU, DEU, FRA, ESN, and so on. If the code is not specified, the default language for the current session's user is used.	No	*User's Default Language*	Input
Status	A string indicating the success of the call or an error message if unsuccessful.	Not applicable	Not applicable	Output

Return Value of the Call

Returns a status string indicating the success or otherwise of the call.

Sample SOAP Requests and Responses

This topic contains sample SOAP requests that invoke UpdatePicklist.

Sample SOAP Request 1 - UpdatePicklist: Create a New Picklist Value

The following SOAP request creates a new picklist value for the CallFrequency picklist of the account record type.

NOTE: When you create new picklist values, the <Code> element is not taken into consideration, and the LIC will be equal to the <DisplayValue> value.

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%%USERNAME%%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <PicklistWS_UpdatePicklist_Input xmlns="urn:crmondemand/ws/picklist/">

      <RecordType>Account</RecordType>
```

```
      <FieldName>CallFrequency</FieldName>

      <CreateNew>Y</CreateNew>

      <ListOfPicklistValues>

        <PicklistValues>

          <Code>BiYearly</Code>

          <DisplayValue>BiYearly_v</DisplayValue>

          <NeedTranslate>N</NeedTranslate>

          <Order>3</Order>

          <Disabled>N</Disabled>

        </PicklistValues>

      </ListOfPicklistValues>

    </PicklistWS_UpdatePicklist_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Request 2 - UpdatePicklist: Update an Existing Picklist Value

The following SOAP request updates an existing picklist value for the CallFrequency picklist of the Account record type. The picklist value is updated in the user's default language:

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%%USERNAME%%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <PicklistWS_UpdatePicklist_Input xmlns="urn:crmondemand/ws/picklist/">

      <RecordType>Account</RecordType>

      <FieldName>CallFrequency</FieldName>

      <ListOfPicklistValues>

        <PicklistValues>

          <Code>BiYearly_v</Code>
```

```
          <DisplayValue>BiYearly</DisplayValue>
          <Order>3</Order>
          <Disabled>N</Disabled>
        </PicklistValues>
      </ListOfPicklistValues>
    </PicklistWS_UpdatePicklist_Input>
  </soap:Body>
</soap:Envelope>
```

Sample SOAP Request 3 - UpdatePicklist with LanguageCode

The following SOAP request updates an existing picklist value for the CallFrequency picklist of the account record type. The picklist value is updated in French as per the <LanguageCode> value specified in the request:

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <wsse:Security>
      <wsse:UsernameToken>
        <wsse:Username>%%%USERNAME%%%</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <PicklistWS_UpdatePicklist_Input xmlns="urn:crmondemand/ws/picklist/">
      <RecordType>Account</RecordType>
      <FieldName>CallFrequency</FieldName>
      <CreateNew>N</CreateNew>
      <LanguageCode>FRA</LanguageCode>
      <ListOfPicklistValues>
        <PicklistValues>
          <Code>BiYearly_v</Code>
          <DisplayValue>BiYearly_FRA</DisplayValue>
          <NeedTranslate>N</NeedTranslate>
          <Order>3</Order>
          <Disabled>N</Disabled>
```

```
        </PicklistValues>
      </ListOfPicklistValues>
    </PicklistWS_UpdatePicklist_Input>
  </soap:Body>
</soap:Envelope>
```

Sample SOAP Requests

The response is the same for each of the three sample SOAP requests:

```
<?xml version="1.0" encoding="utf-16"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns:PicklistWS_UpdatePicklist_Output xmlns:ns="urn:crmondemand/ws/picklist/">
      <ns:Status>TRUE</ns:Status>
    </ns:PicklistWS_UpdatePicklist_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

UpdateCascadingPicklists

Updates cascading picklist values in Oracle CRM On Demand.

Objects Supported

UpdateCascadingPicklists is supported for the same objects as supported by the GetPicklistValues method, see "GetPicklistValues" on page 165.

Usage

You use the UpdateCascadingPicklists method to update cascading picklists, typically in a language-dependent manner.

The ListOfCascadingPicklistsValue argument contains a sequence of ParentCascPicklistsValue elements, which have the following child elements:

- **ParentDisplayValue**. A display value in the specified language.
- **ParentCode**. A parent Language Independent Code (LIC).
- **ListOfChildPickListValue**. A sequence of ChildPicklistValue elements containing the related picklist values that correspond to the parent picklist value.

The ChildPicklistValue element contains the following child elements:

- **DisplayValue**. The display value in the specified language.

- **Code**. The Language Independent Code (LIC).

If the CreateNew argument is set to Y, a new set of cascading picklist relationships is created with the specified values.

You cannot create a new Picklist field using the UpdateCascadingPicklist method.

Arguments

Table 58 describes the arguments taken by the UpdateCascadingPicklists method.

Table 58. Arguments Taken by the UpdateCascadingPicklists Method

Name	Description	Required	Default	I/O
RecordType	The record type; this is case insensitive	Yes	Not applicable	Input
ParentFieldName	The name of the parent picklist field.	Yes	Not applicable	Input
FieldName	The name of the related picklist field.	Yes	Not applicable	Input
CreateNew	Whether a new set of cascading picklist relationships between parent and related picklists is to be created with the input values. A value of Y specifies that a new set of relationships is to be created, the default value of N specifies that picklist values are to be updated.	No	N	Input
Description	A description of the cascading picklist.	No	Not applicable	Input
ListOfCascadingPicklistValue	A sequence of ParentCascPicklistValue elements containing the parent picklist values to be updated or added.	Yes	Not applicable	Input
LanguageCode	The language code for the picklist values to be updated, for example, ENU, DEU, FRA, ESN, and so on. If the code is not specified, the default language for the current session's user is used.	No	*User's Default Language*	Input
Status	A string indicating the success of the call or an error message if unsuccessful.	Not applicable	Not applicable	Output

Return Value of the Call

Returns a status string indicating the success or otherwise of the call.

Sample SOAP Requests and Responses

This topic contains sample SOAP requests that invoke UpdateCascadingPicklists.

Sample SOAP Request 1 - UpdateCascadingPicklist: Create a New Cascading Picklist

The following SOAP request creates a new cascading picklist for the Account record type:

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%%USERNAME%%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <PicklistWS_UpdateCascadingPicklists_Input xmlns="urn:crmondemand/ws/picklist/">

      <RecordType>Account</RecordType>

      <ParentFieldName>AccountType</ParentFieldName>

      <FieldName>Priority</FieldName>

      <CreateNew>Y</CreateNew>

      <ListOfCascadingPicklistsValue>

        <ParentCascPicklistsValue>

          <ParentCode>Competitor</ParentCode>

          <ParentDisplayValue>Competitor</ParentDisplayValue>

          <ListOfChildPicklistValue>

            <ChildPicklistValue>

              <Code>Low</Code>

            </ChildPicklistValue>

          </ListOfChildPicklistValue>

        </ParentCascPicklistsValue>

        <ParentCascPicklistsValue>

          <ParentCode>Customer</ParentCode>

          <ListOfChildPicklistValue>

            <ChildPicklistValue>
```

```
              <Code>Low</Code>
              <DisplayValue>Low</DisplayValue>
            </ChildPicklistValue>
            <ChildPicklistValue>
              <Code>High</Code>
              <DisplayValue>High</DisplayValue>
            </ChildPicklistValue>
          </ListOfChildPicklistValue>
        </ParentCascPicklistsValue>
      </ListOfCascadingPicklistsValue>
    </PicklistWS_UpdateCascadingPicklists_Input>
  </soap:Body>
</soap:Envelope>
```

Sample SOAP Request 2 - UpdateCascadingPicklist: Update an existing cascading picklist

The following SOAP request updates an existing cascading picklist for the account record type. The picklist value is updated in the user's default language:

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <wsse:Security>
      <wsse:UsernameToken>
        <wsse:Username>%%%USERNAME%%%</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <PicklistWS_UpdateCascadingPicklists_Input xmlns="urn:crmondemand/ws/picklist/">
      <RecordType>Account</RecordType>
      <ParentFieldName>AccountType</ParentFieldName>
      <FieldName>Priority</FieldName>
      <ListOfCascadingPicklistsValue>
        <ParentCascPicklistsValue>
          <ParentCode>Partner</ParentCode>
```

```
          <ParentDisplayValue>Partner</ParentDisplayValue>
          <ListOfChildPicklistValue>
            <ChildPicklistValue>
              <Code>Medium</Code>
            </ChildPicklistValue>
          </ListOfChildPicklistValue>
        </ParentCascPicklistsValue>
        <ParentCascPicklistsValue>
          <ParentCode>Customer</ParentCode>
          <ListOfChildPicklistValue>
            <ChildPicklistValue>
              <Code>High</Code>
              <DisplayValue>High</DisplayValue>
            </ChildPicklistValue>
          </ListOfChildPicklistValue>
        </ParentCascPicklistsValue>
      </ListOfCascadingPicklistsValue>
    </PicklistWS_UpdateCascadingPicklists_Input>
  </soap:Body>
</soap:Envelope>
```

Sample SOAP Responses

The response is the same for both of the sample SOAP requests:

```
<?xml version="1.0" encoding="utf-16"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns:PicklistWS_UpdateCascadingPicklists_Output xmlns:ns="urn:crmondemand/ws/picklist/">
      <ns:Status>TRUE</ns:Status>
    </ns:PicklistWS_UpdateCascadingPicklists_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

UserUsageQueryPage

Executes a query against the list of Web Services utilization, and returns a subset of the records that match the search criteria set by the method arguments.

Usage

You use the UserUsageQueryPage method to query the details of your company's Web services utilization. This is the same data that you can view on the Web Services Utilization page in the Oracle CRM On Demand application. As an example of how you might use this data, you might save the data in a CSV file and then import it into a spreadsheet. You could then use the spreadsheet to generate a report showing, for example, how often a user updates his or her records.

In the input request, if you supply a value for UserId in the ListOfUserUsage argument, the method returns the utilization records for the specified user. If you do not have Administrator privileges, you can only specify your own user ID.

If you do not supply a value for UserId in the input request, the method returns:

- The utilization records for the current user, if you do not have Administrator privileges.
- The utilization records for the whole company, if you do have Administrator privileges.

Arguments

Table 59 describes the arguments taken by the UserUsageQueryPage method.

Table 59. Arguments Taken by the UserUsageQueryPage Method

Name	Description	Required	Default	I/O
ListOfUserUsage	The list of Web service utilization queried (input), and after query execution, the list of Web service utilization returned (output).	Yes	Not applicable	Input/ Output
PageSize	The maximum number of records displayed on a page following a query.	No	10	Input
StartRowNum	Indicates the row from which the method starts to return records.	No	0	Input
LastPage	A value that indicates whether or not the last value in the query set has been returned.	Not applicable	Not applicable	Output

Return Value of the Call

The following information is returned for each usage record.

A list of Web service utilization. The following are the child elements of ListOfUserUsage:

- **SessionId**. The session identifier of the Web service request.
- **UserAlias.** The user alias of the user who executed the Web service request.
- **UserId.** The user ID of the user who executed the Web service request.
- **WebServiceName.** The name of the Web service to which the request was made.

- **WebServiceNameSpace**. The namespace used in the request.
- **Operation.** The operation for the Web service request.
- **StartTime.** The start time of the Web service request.
- **EndTime.** The end time of the Web service request.
- **EntryType**. The entry type for the Web service request.
- **InputMessageSize.** The size of the input message.
- **OutputMessageSize.** The size of the output message.
- **ErrorMsg.** Any error message associated with the Web service request.

Sample SOAP Request - UserUsageQueryPage

The following SOAP request returns the list of Web services utilization records. It also demonstrates the use of the optional <PageSize> and <StartRowNum> arguments to return only five records starting at row 2 (third record).

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%%USERNAME%%%</wsse:Username>

        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%%PASSWORD%%%</wsse:Password>

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <UserUsageWS_UserUsageQueryPage_Input xmlns="urn:crmondemand/ws/userusage/">

      <PageSize>5</PageSize>

      <StartRowNum>2</StartRowNum>

      <ListOfUserUsage>

        <UserUsage>

          <SessionId />

          <UserAlias />

          <UserId />

          <WebServiceName />

          <WebServiceNameSpace />
```

```
        <Operation />
        <StartTime />
        <EndTime />
        <EntryType />
        <InputMessageSize />
        <OutputMessageSize />
        <ErrorMsg />
      </UserUsage>
    </ListOfUserUsage>
  </UserUsageWS_UserUsageQueryPage_Input>
 </soap:Body>
</soap:Envelope>
```

Sample SOAP Response

The following illustrates the SOAP response. For brevity, a number of <UserUsage> elements are omitted as indicated by the ellipses (...).

```
<?xml version="1.0" encoding="utf-16"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <SOAP-ENV:Body>
  <ns:UserUsageWS_UserUsageQueryPage_Output xmlns:ns="urn:crmondemand/ws/userusage/">
    <ns:LastPage>false</ns:LastPage>
    <ListOfUserUsage xmlns="urn:/crmondemand/xml/userusage">
      <UserUsage>
        <SessionId>ADSA-920RIQ</SessionId>
        <UserAlias>Joanne Brown</UserAlias>
        <UserId>HRZ780-I5RUE</UserId>
        <WebServiceName></WebServiceName>
        <WebServiceNameSpace></WebServiceNameSpace>
        <Operation>Login</Operation>
        <StartTime>08/02/2011 13:53:00</StartTime>
        <EndTime>08/02/2011 13:53:00</EndTime>
        <EntryType>Login</EntryType>
        <InputMessageSize></InputMessageSize>
        <OutputMessageSize></OutputMessageSize>
        <ErrorMsg></ErrorMsg>
      </UserUsage>
```

```
...
...
        <UserUsage>
          <SessionId>ADSA-921BAY</SessionId>
          <UserAlias>Joanne Brown</UserAlias>
          <UserId>HRZ780-I5RUE</UserId>
          <WebServiceName></WebServiceName>
          <WebServiceNameSpace></WebServiceNameSpace>
          <Operation>Logout</Operation>
          <StartTime>08/02/2011 14:38:15</StartTime>
          <EndTime>08/02/2011 14:38:15</EndTime>
          <EntryType>Logout</EntryType>
          <InputMessageSize></InputMessageSize>
          <OutputMessageSize></OutputMessageSize>
          <ErrorMsg></ErrorMsg>
        </UserUsage>
      </ListOfUserUsage>
    </ns:UserUsageWS_UserUsageQueryPage_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Administrative Services API Calls

The Administrative Services are used to automate the administration of your company's configurations. The Oracle Migration Tool On Demand client is the command-line based utility that eliminates the need to manually copy customized configurations from one environment to another. You can use this downloadable client to extract and import specific configuration details and the Administrative Web services provide programmatic access to the configurations.

NOTE: The Administrative Services must be enabled for your company. To request enablement of the Administrative Services, contact Customer Care.

For more information about the downloadable client, see *Oracle Migration Tool On Demand Guide*.

The Administrative Services and their methods are as shown in Table 60. There are the following methods for each service, though not all services have each method, as shown in the table:

- **<Configuration>Read**. Extracts configuration data matching the input criteria.
- **<Configuration>ReadAll**. Extracts all configuration data of a particular type for a company.
- **<Configuration>Upsert**. Updates existing configuration data or inserts it, if it does not exist.
- **<Configuration>Create**. Creates configuration data.

- **<Configuration>Delete**. Deletes configuration data.

For access to configuration data, various privileges are required, depending on the type of data. The privileges are detailed in the *Oracle Migration Tool On Demand Guide*. Users with an Administrator role will have the necessary privileges to use all of the Administrative Services, although read-only access to some configuration data is available for users who are not administrators, see "Access to the Administrative Services APIs for Non-administrative Users" on page 204.

The Administrative Services do not support stateful login mechanisms, that is, Single Sign-on (SSO), and use of session IDs to authenticate users. For more information about stateful versus stateless session management, see Chapter 4, "Establishing and Managing the Web Services Session."

Table 60. Administrative Services

Service	Method	Usage
Access Profile	AccessProfileRead	Administering access profile configuration data.
	AccessProfileReadAll	
	AccessProfileUpsert	
Action Bar Layout	ActionBarLayoutReadAll	Administering action bar layout configuration data.
	ActionBarLayoutUpsert	
Assessment Scripts	SalesAssessmentTemplateRead	Administering assessment script configuration data.
	SalesAssessmentTemplateReadAll	
Cascading Picklist	CascadingPicklistRead	Administering cascading picklist configuration data.
	CascadingPicklistReadAll	
Company Profile	CurrentOrganizationRead	Administering the company profile data.
Currency Picklist	CurrencyRead	Administering currency picklist configuration data.
	CurrencyReadAll	
	CurrencyUpsert	
Custom Record Type	CustomRecordTypeRead	Administering custom record type configuration data.
	CustomRecordTypeReadAll	
	CustomRecordTypeUpsert	
Custom Web Link	CustomWebLinkRead	Administering Web link configuration data.
	CustomWebLinkReadAll	
	CustomWebLinkUpsert	
Custom Web Tab	CustomWebTabDelete	Administering custom Web tab configuration data.
	CustomWebTabRead	
	CustomWebTabReadAll	
	CustomWebTabUpsert	

Service	Method	Usage
Field Management	FieldManagementCreate	Administering field management configuration data.
	FieldManagementRead	
	FieldManagementReadAll	
	FieldManagementUpsert	
Homepage Layout	HomepageLayoutReadAll	Administering homepage layout configuration data.
Industry	IndustryRead	Administering industry configuration data.
	IndustryReadAll	
Page Layout Field	PageLayoutFieldRead	Administering field information for page layout configuration data.
	PageLayoutFieldReadAll	
	PageLayoutFieldUpsert	
Page Layout Related Information	PageLayoutRelatedInformationRead	Administering related information for page layout configuration data.
	PageLayoutRelatedInformationReadAll	
	PageLayoutRelatedInformationUpsert	
Picklist	PicklistRead	Administering picklist configuration data.
	PicklistReadAll	
	PicklistUpsert	
Role Management	RoleRead	Administering role management configuration data.
	RoleReadAll	
	RoleUpsert	

You can download the WSDL file for each service from the Web Services Administration page in the Oracle CRM On Demand application.

All Web Service operations for the Administrative Services are audited automatically. The audits include all successful operations as well as processing errors such as Request Size exceeded, Invalid SOAP request, Rate limit error, and so on. You can find the audit records in the Web Services Utilization page in Oracle CRM On Demand (click Admin, and then Web Services Utilization).

You can view logs for the Oracle Migration Tool On Demand client in the <*Installation directory*>\log directory, for example, C:\Oracle Migration Tool On Demand\log. Any requests by the client are also captured in the Web Services Utilization page in Oracle CRM On Demand.

The following topics describe each of the methods.

Access to the Administrative Services APIs for Non-administrative Users

Users with an Administrator role have the necessary privileges to use all of the methods of the Administrative Services. However, for the following services, read-only access is available for users who do not have the Administrator role:

- Access Profile
- Currency Picklist
- Custom Record Type
- Field Management
- Page Layout Field
- Page Layout Related Information
- Picklist
- Role Management

Administrators can read, create, update and delete configurations. Other users can only issue Read and ReadAll requests for the services listed; they must only have been granted access to Web services to issue requests (the Enable Web Services Access privilege).

AccessProfileRead

Extracts access profile configuration data.

Usage

Use the AccessProfileRead method to extract access profile configuration data matching the input criteria.

The access profile is specified with the <Name> child element of apQuery:AccessProfile.

See "AccessProfileUpsert" on page 206 for information about the access profile data specified in apData:AccessProfile.

Arguments

Table 61 describes the arguments taken by the AccessProfileRead method.

Table 61. Arguments Taken by the AccessProfileRead Method

Name	Description	Required	Default	I/O
apQuery:AccessProfile	The access profile query criteria.	Yes	Not applicable	Input
apData:AccessProfile	The extracted access profile data.	Not applicable	Not applicable	Output

Return Value of the Call

The access profile is returned.

AccessProfileReadAll

Extracts all of the access profile configuration data for a company.

Usage

Use the AccessProfileReadAll method to read all of the access profiles.

There is no input argument; the method simply returns all of the access profile data.

See "AccessProfileUpsert" on page 206 for information about the access profile data specified in apData:AccessProfile.

Arguments

Table 62 describes the arguments taken by the AccessProfileReadAll method.

Table 62. Arguments Taken by the AccessProfileReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfAccessProfile	The extracted data for all access profiles.	Not applicable	Not applicable	Output

Return Value of the Call

The access profiles are returned.

AccessProfileUpsert

Updates an existing access profile or inserts a new access profile.

Usage

Use the AccessProfileUpsert method to insert a profile or update an existing access profile.

The apdata:AccessProfile argument contains the following elements containing access profile data:

- **Name**. The name of the access profile. When performing an insert operation, this name is used as the display name, if a display name is not provided in the <AccessProfileTranslation> element.
- **Description**. A description of the access profile.
- **AvailableForTeam**. Whether the profile can be assigned to team members.
- **AvailableForBook**. Whether the profile can be assigned to book users.
- **Disabled.** Whether the profile cannot be assigned, but existing assignments continue to work.
- **ListOfAccessProfileTranslation**. The text fields in the languages activated for your company:
  - **LanguageCode**. The three-letter language code for the language.
  - **Title**. The display name of the access profile in the language identified by the language code.
  - **Description**. The description of the access profile in the language identified by the language code.
- **ListOfAccessProfileEntry**. The access rights defined on the access profile for each object and child object:
  - **AccessObjectName**. The names of objects as used in access profiles.
  - **PermissionCode**. The permission codes corresponding to the various access levels for objects.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of three-letter language codes to languages.
- Mappings of access profile object names to the display names for the objects in the Oracle CRM On Demand user interface.
- The permission codes for the various access levels in Oracle CRM On Demand.

Arguments

Table 63 describes the arguments taken by the AccessProfileUpsert method.

Table 63. Arguments Taken by the AccessProfileUpsert Method

Name	Description	Required	Default	I/O
apData:AccessProfile	The access profile data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

ActionBarLayoutReadAll

Extracts all of the action bar layout configuration data for a company.

Usage

Use the ActionBarLayoutReadAll method to read all of the action bar layouts.

There is no input argument; the method simply returns all of the action bar layout data.

See "ActionBarLayoutUpsert" on page 208 for information about the action bar layout data specified in ablData:ActionBarLayout.

Arguments

Table 64 describes the arguments taken by the ActionBarLayoutReadAll method.

Table 64. Arguments Taken by the ActionBarLayoutReadAll Method

Name	Description	Required	Default	I/O
ablData:ListOfActionBarLayout	The extracted data for all action bar layouts.	Not applicable	Not applicable	Output

Return Value of the Call

The action bar layouts are returned.

ActionBarLayoutUpsert

Updates an existing action bar layout or inserts a new action bar layout.

Usage

Use the ActionBarLayoutUpsert method to insert or update an existing action bar layout.

The abldata:ActionBarLayout argument contains the following elements containing action bar layout data:

- **LayoutName**. The name of the action bar layout.
- **Description**. A description of the action bar layout.
- **ListOfSections**. Contains all of the available sections for the action bar layout. The data for a particular section is contained in the following elements:
  - **Name**. The name of the section, which can be one of the following:
    - Calendar Action Applet
    - Global Search Multi Field
    - Quick Create
    - Quick Favorite Lists
    - Quick Favorite Records
    - Quick History
    - Quick Message Center
  - **Displayed**. Whether the section is displayed in the action bar.
  - **Sequence**. The position of the section within the action bar layout. This corresponds to step 2 in the Action Bar Layout wizard in the UI. The integer value can be one of the following:
    - **-2**. The section is in the Not Available Sections list.
    - **-1**. The section is in the Available Sections list.
    - **Positive integers**. The section is in the Displayed Sections list. The integer value indicates the location of the section within the lists, with 0 (zero) meaning the section is located at the top of the list.

See *Oracle Migration Tool On Demand Guide* for information about mappings of action bar section names to the display names for the sections in the Oracle CRM On Demand user interface.

Arguments

Table 65 describes the arguments taken by the ActionBarLayoutUpsert method.

Table 65. Arguments Taken by the ActionBarLayoutUpsert Method

Name	Description	Required	Default	I/O
ablData:ActionBarLayout	The action bar layout data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

CascadingPicklistRead

Extracts cascading picklist configuration data for an object.

Usage

Use the CascadingPicklistRead method to extract the set of cascading picklist data matching the input criteria.

The object is specified with the <ObjectName> child element of apQuery:CascadingPicklistSet. You can also specify a particular parent picklist or related picklist in the query.

The apData:ListOfCascadingPicklistSet output argument can contain any number of CascadingPicklist elements, each of which contains the following data:

- **ParentPicklist**. The parent picklist.
- **RelatedPicklist**. The related picklist.
- **Description**. The description of the cascading picklist.
- **ListOfPicklistValueAssociations**. Contains all the associations for a cascading picklist. The associations for a particular picklist are contained in the following elements:
  - **ParentPicklistValue**. A parent picklist value for the cascading picklist.
  - **RelatedPicklistValue**. The related values for the parent picklist. There is one these elements for each related picklist value.

Arguments

Table 66 describes the arguments taken by the CascadingPicklistRead method.

Table 66. Arguments Taken by the CascadingPicklistRead Method

Name	Description	Required	Default	I/O
apQuery:CascadingPicklistSet	The cascading picklist query criteria.	Yes	Not applicable	Input
apData:ListOfCascadingPicklistSet	The extracted cascading picklist data.	Not applicable	Not applicable	Output

Return Value of the Call

The cascading picklist data for the object is returned.

CascadingPicklistReadAll

Extracts all of the cascading picklist configuration data for a company.

Usage

Use the CascadingPicklistReadAll method to read all of the cascading picklists.

There is no input argument; the method simply returns all of the cascading picklist data for each object.

See "CascadingPicklistRead" on page 209 for information about the cascading picklist data returned in apData:ListOfCascadingPicklistSet.

Arguments

Table 67 describes the arguments taken by the CascadingPicklistReadAll method.

Table 67. Arguments Taken by the CascadingPicklistReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfCascadingPicklistSet	The extracted data for all cascading picklists.	Not applicable	Not applicable	Output

Return Value of the Call

The cascading picklists are returned.

CurrencyRead

Extracts currency configuration data.

Usage

Use the CurrencyRead method to extract currency configuration data matching the input criteria.

The currency is specified with the <Name> child element of apQuery:Currency.

See “CurrencyUpsert” on page 212 for information about the currency data specified in apData:ListOfCurrency.

Arguments

Table 68 describes the arguments taken by the CurrencyRead method.

Table 68. Arguments Taken by the CurrencyRead Method

Name	Description	Required	Default	I/O
apQuery:Currency	The currency query criteria.	Yes	Not applicable	Input
apData:ListOfCurrency	The extracted currency data.	Not applicable	Not applicable	Output

Return Value of the Call

The currency data is returned.

CurrencyReadAll

Extracts all of the currency configuration data for a company.

Usage

Use the CurrencyReadAll method to read all of the currency data.

There is no input argument; the method simply returns all of the currency data.

See “CurrencyUpsert” on page 212 for information about the currency data specified in apData:ListOfCurrency.

Arguments

Table 69 describes the arguments taken by the CurrencyReadAll method.

Table 69. Arguments Taken by the CurrencyReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfCurrency	The extracted data for all currencies.	Not applicable	Not applicable	Output

Return Value of the Call

The currency data is returned.

CurrencyUpsert

Updates existing currency data.

Usage

Use the CurrencyUpsert method to update currency data.

It is not possible to define a new currency or delete an existing currency in Oracle CRM On Demand, therefore CurrencyUpsert only supports update and not insertion of data. The only updating supported is the active/inactive setting and symbol for active currencies.

The apdata:Currency argument contains the following elements containing currency data:

- **Name**. The name of the currency.
- **Code**. The three-letter code for the currency.
- **Symbol**. The symbol for the currency.
- **IssuingCountry**. The country that issues the currency.
- **Active**. Whether the currency is active for the company.

Arguments

Table 70 describes the arguments taken by the CurrencyUpsert method.

Table 70. Arguments Taken by the CurrencyUpsert Method

Name	Description	Required	Default	I/O
apData:Currency	The currency data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

CurrentOrganizationRead

Extracts company profile data.

Usage

Use the CurrentOrganizationRead method to extract company profile data.

Arguments

Table 71 describes the arguments taken by the CurrentOrganizationRead method.

Table 71. Arguments Taken by the CurrentOrganizationRead Method

Name	Description	Required	Default	I/O
apData:ListOfOrganization	The extracted company profile data.	Not applicable	Not applicable	Output

Return Value of the Call

The company profile data is returned. See the Oracle CRM On Demand online help for information about the company profile settings.

CustomRecordTypeRead

Extracts custom record type configuration data.

Usage

Use the CustomRecordTypeRead method to extract custom record type configuration data matching the input criteria.

The custom record type is specified with the <Name> child element of apQuery:CustomRecordType.

If the <IncludeAll> child element of apQuery:CustomRecordType specifies true, the method returns configuration data including preconfigured and customized data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

See "CustomRecordTypeUpsert" on page 215 for information about the custom record type data specified in apData:ListOfCustomRecordType.

Arguments

Table 72 describes the arguments taken by the CustomRecordTypeRead method.

Table 72. Arguments Taken by the CustomRecordTypeRead Method

Name	Description	Required	Default	I/O
apQuery:CustomRecordType	The custom record type query criteria.	Yes	Not applicable	Input
apData:ListOfCustomRecordType	The extracted custom record type data.	Not applicable	Not applicable	Output

Return Value of the Call

The custom record type data is returned.

CustomRecordTypeReadAll

Extracts all of the custom record type configuration data for a company.

Usage

Use the CustomRecordTypeReadAll method to read all of the custom record types.

If the input argument <IncludeAll> specifies true, the method returns all configuration data, including preconfigured and customized data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

See "CustomRecordTypeUpsert" on page 215 for information about the custom record type data specified in apData:ListOfCustomRecordType.

Arguments

Table 73 describes the arguments taken by the CustomRecordTypeReadAll method.

Table 73. Arguments Taken by the CustomRecordTypeReadAll Method

Name	Description	Required	Default	I/O
IncludeAll	If true, extracts all configuration data and not just customized configuration data.	No	false	Input
apData:ListOfCustomRecordType	The extracted data for all custom record types.	Not applicable	Not applicable	Output

Return Value of the Call

The custom record types are returned.

CustomRecordTypeUpsert

Updates an existing custom record type or inserts a new custom record type.

Usage

Use the CustomRecordTypeUpsert method to insert or update custom record type data.

The apdata:CustomRecordType argument contains the following elements containing custom record type data:

- **Name**. The object name of the custom record type.
- **SingularName**. The singular display name used in the UI.
- **PluralName**. The plural display name used in the UI.
- **ShortName**. The short display name used in the UI.

  NOTE: SingularName, PluralName and ShortName are language independent names. If these names are present in the input, the behavior is similar to Mark for Translation being selected in the UI. These elements are not required for customizing language dependent singular/plural/short names for an object, as they are handled by ListOfCustomRecordTypeTranslation.

- **IconName.** The filename of the icon for the custom object.
- **ListOfCustomRecordTypeTranslation**. The object display names in the languages activated for your company:
  - **LanguageCode**. The three-letter language code for the language.
  - **SingularName**. The singular name of the custom record type in the language identified by the language code.

- **PluralName**. The plural name of the custom record type in the language identified by the language code.
- **ShortName**. The short name of the custom record type in the language identified by the language code.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of three-letter language codes to languages.
- Mappings of icon filenames to icons for custom objects in the Oracle CRM On Demand user interface.

Arguments

Table 74 describes the arguments taken by the CustomRecordTypeUpsert method.

Table 74. Arguments Taken by the CustomRecordTypeUpsert Method

Name	Description	Required	Default	I/O
apData:CustomRecordType	The custom record type data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

CustomWebLinkRead

Extracts custom Web link configuration data.

Usage

Use the CustomWebLinkRead method to extract custom Web link configuration data for an object.

The object is specified with the <ObjectName> child element of apQuery:WebLink.

See "CustomWebLinkUpsert" on page 218 for information about the custom Web link data specified in apData:WebLinkSet.

Arguments

Table 75 describes the arguments taken by the CustomWebLinkRead method.

Table 75. Arguments Taken by the CustomWebLinkRead Method

Name	Description	Required	Default	I/O
apQuery:WebLinkSet	The custom Web link query criteria.	Yes	Not applicable	Input
apData:ListOfWebLinkSet	The extracted custom Web link data.	Not applicable	Not applicable	Output

Return Value of the Call

The custom Web link is returned.

CustomWebLinkReadAll

Extracts all of the custom Web link configuration data for a company.

Usage

Use the CustomWebLinkReadAll method to read all of the custom Web links.

There is no input argument; the method simply returns all of the custom Web link data.

See "CustomWebLinkUpsert" on page 218 for information about the custom Web link data specified in apData:WebLinkSet.

Arguments

Table 76 describes the arguments taken by the CustomWebLinkReadAll method.

Table 76. Arguments Taken by the CustomWebLinkReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfWebLinkSet	The extracted data for all custom Web links.	Not applicable	Not applicable	Output

Return Value of the Call

The custom Web links are returned.

CustomWebLinkUpsert

Updates custom Web link data or inserts new custom Web link data for an object

Usage

Use the CustomWebLinkUpsert method to insert custom Web link or update existing custom Web link data for an object.

The object is specified with the <ObjectName> child element of apData:WebLinkSet.

The apdata:WebLinkSet argument contains the following elements containing custom Web link data:

- **Name**. The name of the custom Web link field. When performing an insert operation, this name is used as the display name, if a display name is not provided in the <WebLinkTranslation> element.
- **IntegrationTag**. The field's integration tag name.
- **DisplayText**. The label for the hyperlink of the custom Web link.
- **WebLinkTarget**. The behavior when a user clicks the hyperlink. Open in Current window, Open in Custom Tab, or Open in New window.
- **TargetCustomWebTab**. The required custom Web tab, if Open in Custom Tab is specified for WebLinkTarget.
- **RefreshParentWindow**. Whether the parent window is refreshed after a new window is opened. Only applicable if Open in New window is specified for WebLinkTarget.
- **DisplayOptions**. A character indicating the pages where the custom Web link is displayed: D for Detail Page, E for Edit Page, or B for Both.
- **ActiveLinkCondition.** The condition that controls whether the custom Web link is active on the Detail and Edit pages.
- **DisplayLinkCondition.** The condition that controls whether the custom Web link is visible on the Detail and Edit pages.
- **URL.** The URL that is invoked when the user clicks the custom Web link.
- **ListOfWebLinkTranslations**. The display names in the languages activated for your company:
  - **DisplayName**. The display name of the custom Web link in the language identified by the language code.
  - **DisplayText**. The label for the hyperlink, in the language identified by the language code.
  - **LangCode**. The three-letter language code for the language.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of three-letter language codes to languages.
- Mappings of object names to the display names for the objects in the Oracle CRM On Demand user interface.

For more information about custom Web links, see the Oracle CRM On Demand online help.

Arguments

Table 77 describes the arguments taken by the CustomWebLinkUpsert method.

Table 77. Arguments Taken by the CustomWebLinkUpsert Method

Name	Description	Required	Default	I/O
apData:WebLinkSet	The custom Web link data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

CustomWebTabDelete

Deletes a custom Web tab.

Usage

Use the CustomWebTabDelete method to delete a custom Web tab matching the input criteria.

The custom Web tab is specified with the <DisplayName> child element of apQuery:CustomWebTab.

See “CustomWebTabUpsert” on page 221 for information about the custom Web tab data specified in apData:CustomWebTab.

Arguments

Table 78 describes the arguments taken by the CustomWebTabDelete method.

Table 78. Arguments Taken by the CustomWebTabDelete Method

Name	Description	Required	Default	I/O
apQuery:CustomWebTab	The custom Web tab query criteria.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status of the call is returned.

CustomWebTabRead

Extracts custom Web tab configuration data.

Usage

Use the CustomWebTabRead method to extract custom Web tab configuration data matching the input criteria.

The custom Web tab is specified with the <DisplayName> child element of apQuery:CustomWebTab.

See “CustomWebTabUpsert” on page 221 for information about the custom Web tab data specified in apData:CustomWebTab.

Arguments

Table 79 describes the arguments taken by the CustomWebTabRead method.

Table 79. Arguments Taken by the CustomWebTabRead Method

Name	Description	Required	Default	I/O
apQuery:CustomWebTab	The custom Web tab query criteria.	Yes	Not applicable	Input
apData:CustomWebTab	The extracted custom Web tab data.	Not applicable	Not applicable	Output

Return Value of the Call

The custom Web tab is returned.

CustomWebTabReadAll

Extracts all of the custom Web tab configuration data for a company.

Usage

Use the CustomWebTabReadAll method to read all of the custom Web tabs.

There is no input argument; the method simply returns all of the custom Web tab data.

See “CustomWebTabUpsert” on page 221 for information about the custom Web tab data specified in apData:CustomWebTab.

Arguments

Table 80 describes the arguments taken by the CustomWebTabReadAll method.

Table 80. Arguments Taken by the CustomWebTabReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfCustomWebTab	The extracted data for all custom Web tabs.	Not applicable	Not applicable	Output

Return Value of the Call

The custom Web tabs are returned.

CustomWebTabUpsert

Updates an existing custom Web tab or inserts a new custom Web tab.

Usage

Use the CustomWebTabUpsert method to insert a custom Web tab or update an existing custom Web tab.

The apdata:CustomWebTab argument contains the following elements containing custom Web tab data:

- **DisplayName**. The name of the custom Web tab. When performing an insert operation, this name is used as the display name, if a display name is not provided in the <CustomWebTabTranslation> element.
- **Description**. A description of the custom Web tab.
- **Type.** The type of custom Web Tab, which can be URL or HTML.
- **HTMLHeadAdditions**. The HTML to be added within the <head> element of the custom Web tab. (For type HTML only).
- **WebTabHTML**. The HTML to be added within the <body> element of the custom Web tab. (For type HTML only).
- **Url**. The URL for the custom Web tab.
- **FrameHeight.** The frame height defined for the custom Web tab.
- **FrameWidth.** The frame width defined for the custom Web tab.
- **IconName.** The filename of the icon used for the custom Web tab.
- **ListOfCustomWebTabTranslation**. The display names in the languages activated for your company:
  - **LanguageCode**. The three-letter language code for the language.

- **DisplayName**. The display name of the custom Web tab in the language identified by the language code.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of three-letter language codes to languages.

Arguments

Table 81 describes the arguments taken by the CustomWebTabUpsert method.

Table 81. Arguments Taken by the CustomWebTabUpsert Method

Name	Description	Required	Default	I/O
apData:CustomWebTab	The custom Web tab data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

FieldManagementCreate

Creates field management configuration data.

Usage

Use the FieldManagementCreate method to create field management configuration data.

The object is specified with the <ObjectName> child element of apData:CustomField. The field management data for the object is contained in elements similar to those of the apData:FieldManagement argument used by the FieldManagementUpsert method, see "FieldManagementUpsert" on page 225.

Arguments

Table 82 describes the arguments taken by the FieldManagementCreate method.

Table 82. Arguments Taken by the FieldManagementCreate Method

Name	Description	Required	Default	I/O
apData:CustomField	The field management query criteria.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status of the call is returned.

FieldManagementRead

Extracts field management configuration data.

Usage

Use the FieldManagementRead method to extract field management configuration data for an object.

The object is specified with the <ObjectName> child element of apQuery:FieldSet.

If the <IncludeAll> child element of apQuery:FieldSet specifies true, the method returns configuration data including preconfigured and customized data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

Regardless of whether only customized data is returned, or preconfigured and customized data, translated data for all languages enabled by the company are included in the output.

See "FieldManagementUpsert" on page 225 for information about the field management data specified in apData:FieldSet.

FieldManagementRead only extracts field definitions for fields that have been created by the user, or preconfigured fields that have been modified. Unmodified preconfigured fields are bypassed because they exist on all company instances by default.

Arguments

Table 83 describes the arguments taken by the FieldManagementRead method.

Table 83. Arguments Taken by the FieldManagementRead Method

Name	Description	Required	Default	I/O
apQuery:FieldSet	The field management query criteria.	Yes	Not applicable	Input
apData:ListOfFieldSet	The extracted field management data.	Not applicable	Not applicable	Output

Return Value of the Call

The field management data is returned.

FieldManagementReadAll

Extracts all of the field management configuration data for a company.

Usage

Use the FieldManagementReadAll method to read all of the field management data.

If the input argument <IncludeAll> specifies true, the method returns all configuration data, including preconfigured and customized data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

Regardless of whether only customized data is returned, or preconfigured and customized data, translated data for all languages enabled by the company are included in the output.

See "FieldManagementUpsert" on page 225 for information about the field management data specified in apData:FieldSet.

Arguments

Table 84 describes the arguments taken by the FieldManagementReadAll method.

Table 84. Arguments Taken by the FieldManagementReadAll Method

Name	Description	Required	Default	I/O
IncludeAll	If true, extracts all configuration data and not just customized configuration data.	No	false	Input
apData:ListOfFieldSet	The extracted field management data.	Not applicable	Not applicable	Output

Return Value of the Call

The field management data is returned.

FieldManagementUpsert

Updates existing field management data or inserts field management data for an object.

Usage

Use the FieldManagementUpsert method to insert field management data or update existing field management data for an object.

The object is specified with the <ObjectName> child element of apData:FieldSet.

The apdata:FieldSet argument contains the following elements containing field management data:

- **Name**. The system name of the field.
- **DisplayName**. The name of the field used in the UI. When performing an insert operation, this name is used as the display name, if a display name is not provided in the <FieldManagementTranslation> element.
- **FieldType**. The type of field.
- **IntegrationTag**. The field's custom integration tag name. This tag can be modified by the user.
- **GenericIntegrationTag**. The field's default integration tag name assigned by Oracle CRM On Demand. This tag cannot be modified by the user. The default integration tag name has the format *fieldtypeDisplay_Name*. For example, stMy_Custom_Field would be assigned to a custom field of type Text (Short) with the display name of My Custom Field.

  NOTE: For non-custom fields, the <GenericIntegrationTag> and <IntegrationTag> values are the same.

- **Required**. Whether the field is a required field (true), or is not a required field (false).
- **ReadOnly**. Whether the field is a read-only field (true), or is not a read-only field (false).

- **DefaultValue.** The default value of the field.
- **FieldValidation**. The expression for the field validation rules for the field.
- **PostDefault**. Whether the field is prepopulated with the default value when a user creates a new record and the record is saved.
- **ValidationErrorMsg**. The custom error message displayed if the field fails field validation.
- **ListOfFieldTranslations**. The field management data in the languages activated for your company:
  - **LanguageCode**. The three-letter language code for the language.
  - **DisplayName**. The display name of the field in the language identified by the language code.
  - **ValidationErrorMsg**. The custom validation error message in the language identified by the language code.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of three-letter language codes to languages.
- Mappings of object names to the display names for the objects in the Oracle CRM On Demand user interface.

For more information about field management, including field validation, see the Oracle CRM On Demand online help.

Arguments

Table 85 describes the arguments taken by the FieldManagementUpsert method.

Table 85. Arguments Taken by the FieldManagementUpsert Method

Name	Description	Required	Default	I/O
apData:FieldSet	The field management data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

HomepageLayoutReadAll

Extracts all of the homepage layout configuration data for a company.

Usage

Use the HomepageLayoutReadAll method to read all of the homepage layouts.

There is no input argument; the method simply returns all of the homepage layout data.

The apdata:HomepageLayout argument contains the following elements containing homepage layout data:

- **ObjectName**. The name of the object.
- **LayoutName**. The name of the homepage layout.
- **Description**. A description of the homepage layout.
- **ListOfAvailableSection**. Contains all of the available sections for the homepage layout. The data for a particular section is contained in the following elements:
  - **SectionName**. The name of the section.
  - **SectionLayoutName**. The name of the list of available sections, such as Available Sections.
- **ListOfSelectedSection**. Contains all of the selected sections for the homepage layout. The data for a particular section is contained in the following elements:
  - **SectionName**. The name of the section.
  - **SectionLayoutName**. The name of the lists of sections, such as Left Side and Right Side.
  - **Sequence**. The sequence of the section within the lists of sections:
    - A sequence value of -2 displays the section in the Available column.
    - All even numbers, including 0 (zero) displays the specified section in the Left Side column, where 0 (zero) represents the first section, and so on.
    - All odd numbers display the section in the Right Side column, where a value of 1 represents the first section, and so on.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of section applet names to section names used in the Oracle CRM On Demand user interface.
- Mappings of section layout names to section names used in the Oracle CRM On Demand user interface.
- Mappings of object names to the display names for the objects in the Oracle CRM On Demand user interface.

Arguments

Table 86 describes the arguments taken by the HomepageLayoutReadAll method.

Table 86. Arguments Taken by the HomepageLayoutReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfHomepageLayout	The extracted data for all homepage layouts.	Not applicable	Not applicable	Output

Return Value of the Call

The homepage layouts are returned.

IndustryRead

Extracts industry configuration data.

Usage

Use the IndustryRead method to extract industry configuration data for your company.

The industry is specified with the <Name> child element of apQuery:Industry.

See "IndustryReadAll" on page 229 for information about the industry data specified in apData:Industry.

Arguments

Table 87 describes the arguments taken by the IndustryRead method.

Table 87. Arguments Taken by the IndustryRead Method

Name	Description	Required	Default	I/O
apQuery:Industry	The industry query criteria.	Yes	Not applicable	Input
apData:ListOfIndustry	The extracted industry data.	Not applicable	Not applicable	Output

Return Value of the Call

The industry data is returned.

IndustryReadAll

Extracts all of the industry configuration data for a company.

Usage

Use the IndustryReadAll method to read all of the industry configuration data for your company.

There is no input argument; the method simply returns all of the industry data.

The apdata:Industry argument contains the following elements containing industry data:

- **Name**. The name of the industry.
- **Active**. Whether the industry is enabled.
- **Type**. The type of industry classification used, for example, 4-digit SIC.
- **DisplayName**. The display name of the industry.
- **SICode.** The code for the industry according to the Standard Industrial Classification (SIC) system.
- **MarkforTranslation**. Whether the industry display name is translated.
- **ListOfIndustryTranslation**. The industry names in the languages activated for your company:
  - **LanguageCode**. The three-letter language code for the language.
  - **Title**. The display name of the industry in the language identified by the language code.

Arguments

Table 88 describes the arguments taken by the IndustryReadAll method.

Table 88. Arguments Taken by the IndustryReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfIndustry	The extracted industry data for the company.	Not applicable	Not applicable	Output

Return Value of the Call

The industry data for the company is returned.

PageLayoutFieldRead

Extracts all of the field configuration data for a detail page of an object.

Usage

Use the PageLayoutFieldRead method to read the field data for an object's detail page.

The object is specified with the <ObjectName> child element of apQuery:PageLayout.

If the <CurrentUserOnly> child element of apQuery:PageLayout specifies true, page layout field data for the current user only is returned.

If the <IncludeAll> child element of apQuery:PageLayout specifies true, the method returns configuration data including preconfigured and customized data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

See "PageLayoutFieldUpsert" on page 231 for information about the page layout field data specified in apData:PageLayout.

Arguments

Table 89 describes the arguments taken by the PageLayoutFieldRead method.

Table 89. Arguments Taken by the PageLayoutFieldRead Method

Name	Description	Required	Default	I/O
apQuery:PageLayout	The page layout query criteria.	Yes	Not applicable	Input
apData:ListOfPageLayout	The extracted page layout field data.	Not applicable	Not applicable	Output

Return Value of the Call

The page layout fields are returned.

PageLayoutFieldReadAll

Extracts all of the field configuration data for detail pages for a company.

Usage

Use the PageLayoutFieldReadAll method to read all of the field data for detail pages.

If the input argument <CurrentUserOnly> specifies true, all of the page layout field data for the current user only is returned.

If the input argument <IncludeAll> specifies true, the method returns all configuration data, including preconfigured and customized data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

There is no input argument; the method simply returns all of the page layout field data.

See "PageLayoutFieldUpsert" on page 231 for information about the page layout field data specified in apData:PageLayout.

Arguments

Table 90 describes the arguments taken by the PageLayoutFieldReadAll method.

Table 90. Arguments Taken by the PageLayoutFieldReadAll Method

Name	Description	Required	Default	I/O
CurrentUserOnly	If true, extracts data for the current user only.	No	false	Input
IncludeAll	If true, extracts all configuration data and not just customized configuration data.	No	false	Input
apData:ListOfPageLayout	The extracted page layout field data.	Not applicable	Not applicable	Output

Return Value of the Call

The page layout fields are returned.

PageLayoutFieldUpsert

Updates existing page layout field data or inserts new page layout field data.

Usage

Use the PageLayoutFieldUpsert method to insert or update field data for a detail page layout.

The apdata:PageLayout argument contains the following elements containing page layout field data:

- **ObjectName**. The name of the object.
- **LayoutName**. The name of the page layout.
- **Description**. A description of the page layout.
- **ListOfPageLayoutField**. Contains all of the fields for the page layout. The data for a particular field is contained in the following child elements of <PageLayoutField>:
  - **FieldName**. The name of the field.
  - **ReadOnly**. Whether the field is read-only.
  - **Required**. Whether the field is required.

- **Section**. An integer identifying the section in which the field appears. The value 0 represents the first section, 1 the next section, and so on.
- **Column**. The column within the section in which the field appears. The value 0 represents the left-most column.
- **Row**. The row within the section in which the field appears. The value 0 represents the first row within a section and column, 1 the next row, and so on.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of object names to the display names for the objects in the Oracle CRM On Demand user interface.

Arguments

Table 91 describes the arguments taken by the PageLayoutFieldUpsert method.

Table 91. Arguments Taken by the PageLayoutFieldUpsert Method

Name	Description	Required	Default	I/O
apData:PageLayout	The page layout field data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

PageLayoutRelatedInformationRead

Extracts all of the related information for a detail page layout for an object.

Usage

Use the PageLayoutRelatedInformationRead method to read all of the related information data for an object.

The object is specified with the <ObjectName> child element of apQuery:PageLayoutRelatedInformation.

If the <CurrentUserOnly> child element of apQuery:PageLayoutRelatedInformation specifies true, related information data for the current user only is returned.

If the <IncludeAll> child element of apQuery:PageLayoutRelatedInformations specifies true, the method returns configuration data including preconfigured and customized data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

See "PageLayoutRelatedInformationUpsert" on page 234 for information about the page layout related information specified in apData:PageLayoutRelatedInformation.

Arguments

Table 92 describes the arguments taken by the PageLayoutRelatedInformationRead method.

Table 92. Arguments Taken by the PageLayoutRelatedInformationRead Method

Name	Description	Required	Default	I/O
apQuery:PageLayoutRelatedInformation	The related information query criteria.	Yes	Not applicable	Input
apData:ListOfPageLayoutRelatedInformation	The extracted page layout related information.	Not applicable	Not applicable	Output

Return Value of the Call

The page layout related information is returned.

PageLayoutRelatedInformationReadAll

Extracts all of the related information for detail page layouts for a company.

Usage

Use the PageLayoutRelatedInformationReadAll method to read all of the related information data for detail page layouts.

If the input argument <CurrentUserOnly> is true, all of the related information data for the current user only is returned.

If the input argument <IncludeAll> specifies true, the method returns all configuration data, including preconfigured and customized data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

See "PageLayoutRelatedInformationUpsert" on page 234 for information about the page layout related information specified in apData:PageLayoutRelatedInformation.

Arguments

Table 93 describes the arguments taken by the PageLayoutRelatedInformationReadAll method.

Table 93. Arguments Taken by the PageLayoutRelatedInformationReadAll Method

Name	Description	Required	Default	I/O
CurrentUserOnly	If true, extracts data for the current user only	No	false	Input
IncludeAll	If true, extracts all configuration data and not just customized configuration data.	No	false	Input
apData:ListOfPageLayoutRelatedInformation	The extracted page layout related information.	Not applicable	Not applicable	Output

Return Value of the Call

The page layout related information is returned.

PageLayoutRelatedInformationUpsert

Updates existing page layout related information or inserts new page layout related information.

Usage

Use the PageLayoutRelatedInformationUpsert method to insert or update related information for a detail page layout.

The apdata:PageLayoutRelatedInformation argument contains the following elements containing related information for a page layout:

- **ObjectName**. The name of the object.
- **LayoutName**. The name of the page layout.
- **ListOfPageRelatedInformation**. Contains all of the related information for the page layout. The data for a particular related information section is contained in the following child elements of <PageRelatedInformation>:
  - **Name**. A child object ID, equivalent to the name of a related information section. See *Oracle Migration Tool On Demand Guide* for reference tables about the mapping of child object IDs to UI display names for the child objects.

- **Availability**. An integer with the range of values -1 through 1, specifying in which list the related information appears in the page layout: Not Available, Available, or Displayed respectively.

  For values of 1 (Displayed), the order of the <PageRelatedInformation> elements determines the order in which the related information sections are listed in the layout in the UI.

- **RelatedInformationLayout**. The name of the corresponding related information layout.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of object names to the display names for the objects in the Oracle CRM On Demand user interface.

Arguments

Table 94 describes the arguments taken by the PageLayoutRelatedInformationUpsert method.

Table 94. Arguments Taken by the PageLayoutRelatedInformationUpsert Method

Name	Description	Required	Default	I/O
apData:PageLayoutRelatedInformation	The page layout related information to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

PicklistRead

Extracts picklist configuration data.

Usage

Use the PicklistRead method to extract picklist configuration data for an object.

The object is specified with the <ObjectName> child element of apQuery:PicklistSet.

If the <IncludeAll> child element of apQuery:PicklistSet specifies true, the method returns configuration data including preconfigured and customized data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

See "PicklistUpsert" on page 237 for information about the picklist data specified in apData:PicklistSet.

Arguments

Table 95 describes the arguments taken by the PicklistRead method.

Table 95. Arguments Taken by the PicklistRead Method

Name	Description	Required	Default	I/O
apQuery:PicklistSet	The picklist query criteria.	Yes	Not applicable	Input
apData:ListOfPicklistSet	The extracted picklist data.	Not applicable	Not applicable	Output

Return Value of the Call

The picklists for the object are returned.

PicklistReadAll

Extracts all of the picklist configuration data for a company.

Usage

Use the PicklistReadAll method to read all of the picklists.

If the input argument <IncludeAll> specifies true, the method returns all configuration data, including preconfigured and customized data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

See "PicklistUpsert" on page 237 for information about the picklist data specified in apData:PicklistSet.

Arguments

Table 96 describes the arguments taken by the PicklistReadAll method.

Table 96. Arguments Taken by the PicklistReadAll Method

Name	Description	Required	Default	I/O
IncludeAll	If true, extracts all configuration data and not just customized configuration data.	No	false	Input
apData:ListOfPicklistSet	The extracted data for all picklists for all objects.	Not applicable	Not applicable	Output

Return Value of the Call

The picklists are returned.

PicklistUpsert

Updates an existing set of picklists or inserts a new set of picklists.

Usage

Use the PicklistUpsert method to insert a set of picklists or update an existing set of picklists for an object.

The object is specified with the <ObjectName> element of apdata:PicklistSet.

The apdata:PicklistSet argument contains the following elements containing picklist data:

- **Name**. The name of the picklist. When performing an insert operation, this name is used as the display name, if a display name is not provided in the <PicklistTranslation> element.
- **PickListValue**. The picklist value.
- **ValueId**. The number that determines the position of the value in the picklist.
- **Disabled.** Whether the picklist value is disabled.
- **ListOfPicklistTranslation**. The picklist value in the languages activated for your company:
  - **LanguageCode**. The three-letter language code for the language.
  - **Value**. The picklist value in the language identified by the language code.
  - **Order**. The order number of the picklist value according to language identified by the language code.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of three-letter language codes to languages.
- Mappings of object names to the display names for the objects in the Oracle CRM On Demand user interface.

Arguments

Table 97 describes the arguments taken by the PicklistUpsert method.

Table 97. Arguments Taken by the PicklistUpsert Method

Name	Description	Required	Default	I/O
apData:PicklistSet	The picklist data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

RoleRead

Extracts role configuration data.

Usage

Use the RoleRead method to extract role configuration data.

The role is specified with the <RoleName> child element of apQuery:Role.

See "RoleUpsert" on page 239 for information about the role data specified in apData:Role.

Arguments

Table 98 describes the arguments taken by the RoleRead method.

Table 98. Arguments Taken by the RoleRead Method

Name	Description	Required	Default	I/O
apQuery:Role	The role query criteria.	Yes	Not applicable	Input
apData:ListOfRole	The extracted role data.	Not applicable	Not applicable	Output

Return Value of the Call

The roles for the object are returned.

RoleReadAll

Extracts all of the role configuration data for a company.

Usage

Use the RoleReadAll method to read all of the roles.

See "RoleUpsert" on page 239 for information about the role data specified in apData:Role.

Arguments

Table 99 describes the arguments taken by the RoleReadAll method.

Table 99. Arguments Taken by the RoleReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfRole	The extracted data for all roles for all objects.	Not applicable	Not applicable	Output

Return Value of the Call

The roles are returned.

RoleUpsert

Updates an existing role or inserts a new role.

Usage

Use the RoleUpsert method to insert a role or update an existing role.

The role is specified with the <RoleName> element of apdata:Role.

The apdata:Role argument contains the following elements containing role data:

- **RoleName**. The name of the role. When performing an insert operation, this name is used as the display name, if a display name is not provided in the <RoleTranslation> element.
- **Description**. The role description.
- **DefaultSalesProcess**. The default sales process for new opportunities created by users who are assigned this role.
- **ThemeName**. The default theme for the role.
- **LeadConversionLayout**. The lead conversion layout for the users assigned this role.
- **ActionBarLayout**. The action bar layout for the users assigned this role.

- **ListOfRoleTranslation**. The role name in the languages activated for your company:
  - **LanguageCode**. The three-letter language code for the language.
  - **RoleName**. The role name in the language identified by the language code.
- **ListOfRecordTypeAccess**. The record type access data:
  - **RecordName**. The record type.
  - **HasAccess**. Whether users assigned this role have access to records of this type.
  - **CanCreate.** Whether users assigned this role can create records of this type.
  - **CanReadAll**. Whether users assigned this role can see all records of this type.
- **AccessProfile**. The access profiles associated with the role:
  - **DefaultAccessProfile**. The default access profile.
  - **OwnerAccessProfile**. The owner access profile.
- **ListOfPrivilege**. The privileges associated with the role:
  - **PrivilegeName**. A privilege ID, equivalent to the name of a privilege in the Oracle CRM On Demand application. See *Oracle Migration Tool On Demand Guide* for reference tables about the mapping of privilege IDs to UI display names for the privileges.
  - **Enabled**. Whether the privilege is assigned to the role.
- **TabAccessandOrder**. The tabs available and selected for the role:
  - **ListOfAvailableTab**. The available tabs:
    - **AvailableTab**. The name of the tab.
  - **ListOfSelectedTab**. The selected tabs:
    - **TabName**. The name of the tab.
    - **Order**. The position in which the tab name appears in the Selected Tabs list. A value of 0 (zero) corresponds to the first tab in the Selected Tabs list.
- **ListOfPageLayoutAssignment**. The page layout assignment for each record type:
  - **RecordType**. The record type.
  - **LayoutName**. The name of the page layout. If no name is specified, the default layout is applied.
- **ListOfSearchLayoutAssignment**. The search layout assignment for each record type:
  - **RecordType**. The record type.
  - **LayoutName**. The name of the search layout.
- **ListOfHomepageLayoutAssignment**. The homepage layout assignment for each record type:
  - **RecordType**. The record type.
  - **LayoutName**. The name of the homepage layout.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of three-letter language codes to languages.
- Information on privilege IDs, and their corresponding names, and descriptions.

Arguments

Table 100 describes the arguments taken by the RoleUpsert method.

Table 100. Arguments Taken by the RoleUpsert Method

Name	Description	Required	Default	I/O
apData:Role	The role data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

SalesAssessmentTemplateRead

Extracts sales assessment template configuration data.

Usage

Use the SalesAssessmentTemplateRead method to extract sales assessment template configuration data matching the input criteria.

The sales assessment template is specified with the <Name> child element of apQuery:SalesAssessmentTemplate.

See “SalesAssessmentTemplateReadAll” on page 242 for information about the sales assessment template data specified in apData:SalesAssessmentTemplate.

Arguments

Table 101 describes the arguments taken by the SalesAssessmentTemplateRead method.

Table 101. Arguments Taken by the SalesAssessmentTemplateRead Method

Name	Description	Required	Default	I/O
apQuery:SalesAssessmentTemplate	The sales assessment template query criteria.	Yes	Not applicable	Input
apData:ListOfSalesAssessmentTemplate	The extracted sales assessment template data.	Not applicable	Not applicable	Output

Return Value of the Call

The sales assessment template is returned.

SalesAssessmentTemplateReadAll

Extracts all of the sales assessment template configuration data for a company.

Usage

Use the SalesAssessmentTemplateReadAll method to read all of the sales assessment templates.

The apdata:SalesAssessmentTemplate element contains the following elements containing sales assessment template data:

- **Name**. The name for the sales assessment template (required).
- **Active.** Whether the sales assessment template is active.
- **Filter*n***. The filters used to identify which assessment script is the appropriate one to present to a user for a task.
- **Description**. A description for the sales assessment template.
- **FieldtoMapScoreTo**. The field that displays the score in the parent record.
- **ResponseControl**. The control used for adding responses to assessment questions. Possible values are: Drop Down, Radio Button.
- **FieldtoMapOutcomeValueTo**. The field that displays the outcome value in the parent record.
- **RemoveCommentBox**. Whether the comment box is removed when the user is completing the assessment.
- **OutcomeValueIfThresholdMet**. The outcome value to display if the threshold score is met.
- **OutcomeValueIfThresholdNotMet**. The outcome value to display if the threshold score is not met.

- **ThresholdScore**. The score used to calculate the outcome of the script. If the score is equal to or higher than the threshold, the outcome is met.
- **Type**. The type of assessment script. Possible values are: Contact Script, Lead Qualification, Opportunity Assessment, Service Request - Script, Service Request - Survey, and Activity Assessment. (required).
- **ListOfSalesAssessmentTemplateAttribute**. The attributes for the sales assessment template. The following are child elements of the SalesAssessmentTemplateAttributeData element.
  - **Question.** A question to be answered by the user (required).
  - **CriteriaName**. A descriptive name for the question (required).
  - **AnswerMapToField**. The field on the parent record for displaying the answer to the question.
  - **Order.** The position in the sequence of the questions presented to the user.
  - **Weight.** A percentage value indicating the importance of the question. The sum of the weights for all questions must equal 100 (required).
  - **ListOfSalesAssessmentTemplateAttributeValue.** The attribute values of the sales assessment template. The following are child elements of the element SalesAssessmentAttributeValueData:
    - **Order.** The position in the sequence of the answers presented to the user when completing the assessment.
    - **Value.** An answer for the question (required).
    - **Score.** The score for the answer (required).

Arguments

Table 102 describes the arguments taken by the SalesAssessmentTemplateReadAll method.

Table 102. Arguments Taken by the SalesAssessmentTemplateReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfSalesAssessmentTemplate	The extracted data for all sales assessment templates.	Not applicable	Not applicable	Output

Return Value of the Call

The sales assessment templates are returned.

Data Loader API Calls

In Oracle CRM On Demand you can use the Oracle Data Loader On Demand tool to perform bulk data import operations. Client applications can also take advantage of the API used by the tool to make Web services calls to submit bulk import requests.

Users with the Admin Import privilege can download the Oracle Data Loader On Demand tool from the Oracle CRM On Demand UI. When you download the tool, a WSDL file is provided as part of the.zip file: OracleDataLoaderOnDemandImportServices.wsdl.

Table 103 lists the calls available with the Data Loader API.

Table 103. Calls Available in the Data Loader API

Name	Description
BulkOpImportCreateRequest	Creates a bulk import request.
BulkOpImportSendData	Imports the data. The BulkOpImportCreateRequest call must be made before BulkOpImportSendData. Subsequently, BulkOpImportSendData calls can be made repeatedly to send data for importing.
BulkOpImportGetRequestDetail	Gets the details of a bulk import request.

BulkOpImportCreateRequest

Creates a bulk import request.

Usage

Use the BulkOpImportCreateRequest method to create a bulk import request.

The arguments taken by the method correspond to elements in the WSDL file, as described in Table 104.

The xsdLocal1:ImportOptions argument specifies the import options, which are similar to the options available in the Import Wizard in the Oracle CRM On Demand application:

- **CSVDelimiter**. The type of CSV delimiter used in the file: comma (,) or semi-colon (;).
- **DateTimeFormat**. The date and time format used in the CSV file: usa, can, eur, uk, other.
- **DuplicateCheckOption**. Whether the Row Id (rowid) or external ID (externalid) are used for duplicate checking.
- **ErrorLogLevel**. The level of error logging:
  - all. All messages
  - errors. Error messages only.
  - errorswarnings. Error messages and warning messages only.

Typical customer import data is contained in a CSV file with the following format:

```
ColumnName1, ColumnName2, ...

Data1, Data2, ...

Data1, Data2, ...

...
```

The xsdLocal1:ListOfFieldMappings element specifies the mapping of CSV column names to field names in Oracle CRM On Demand. For example:

```
<ns1:ListOfFieldMappings>
   <ns1:FieldMapping>
      <ns1:CSVColumnName>
         Web Site
      </ns1:CSVColumnName>
      <ns1:FieldName>
         Home Page
      </ns1:FieldName>
   </ns1:FieldMapping>
</ns1:ListOfFieldMappings>
```

CSVColumnHeaders specifies the list of CSV column names, for example:

```
<ns0:CSVColumnHeaders>
   "Account Name","External Unique Id","Web Site","Account Location"
</ns0:CSVColumnHeaders>
```

The mappings are validated and if everything is correct, the import request is created.

Arguments

Table 104 describes the arguments taken by the BulkOpImportCreateRequest method.

Table 104. Arguments Taken by the BulkOpImportCreateRequest Method

Name	Description	Required	Default	I/O
DataFileName	The name of the data file to be imported, for example, account1.csv	Yes	Not applicable	Input
RecordType	The record type to be imported.	Yes	Not applicable	Input
xsdLocal1:ImportOptions	The list of import options.	No	Not applicable	Input
xsdLocal1:ListOfFieldMappings	The mapping of CSV file column names and their corresponding fields.	Yes	Not applicable	Input
CSVColumnHeaders	The list of column names from the CSV file.	Yes	Not applicable	Input
EnableImportAudit	Enables or disables auditing of record creation/update during import requests. This setting takes effect only f the user has the privilege Manage Record Auditing for Imports, otherwise it is ignored. Accepted values are true or false.	No	true	Input
CSVRowCount	The number of data records in the data file to be imported.	Yes	Not applicable	Input
Operation	The operation for the current request, such as insert, update.	Yes	Not applicable	Input
ClientVersion	The current Data Loader client version number.	No	Not applicable	Input
RequestId	The ID of the request created.	Not applicable	Not applicable	Output
Status	The status of this Web service request.	Not applicable	Not applicable	Output
ErrorMessage	Any error message from this Web service request, or a message indicating success.	Not applicable	Not applicable	Output

Return Value of the Call

The request ID, status, and any error message are returned.

BulkOpImportSendData

Imports the data for a bulk import request.

Usage

Use the BulkOpImportSendData method to import the data for a bulk import request.

The BulkOpImportCreateRequest call must be invoked before BulkOpImportSendData by the client.

...

The ListOfCSVData element specifies the row data from the CSV file. For example:

```
<ns1:ListOfCSVData>
   <ns1:CSVData>
      <ns1:Row>
         "High Quality Moving Company","HQMC","www.highmoving.net","San Jose"
      </ns1:Row>
   </ns1:CSVData>
</ns1:ListOfCSVData>
```

The CSV data is processed and the actual import performed.

Arguments

Table 105 describes the arguments taken by the BulkOpImportSendData method.

Table 105. Arguments Taken by the BulkOpImportSendData Method

Name	Description	Required	Default	I/O
RequestId	The request ID that was created.	Yes	Not applicable	Input/ Output
TotalNumberOfCSVDataLists	The total number of BulkOpImportSendData calls needed to send all CSV file data.	Yes	Not applicable	Input
xsdLocal1:ListOfCSVData	The actual list of CSV data contained in the CSV file.	Yes	Not applicable	Input

Table 105. Arguments Taken by the BulkOpImportSendData Method

Name	Description	Required	Default	I/O
StartRecordNumber	The starting line number of the CSV data.	Yes	Not applicable	Input
ClientVersion	The current data loader client version number.	No	Not applicable	Input
Status	The status of this Web service call.	Not applicable	Not applicable	Output
ErrorMessage	Any error message from this Web service call, or a message indicating success.	Not applicable	Not applicable	Output

Return Value of the Call

The request ID, status, and any error message are returned.

BulkOpImportGetRequestDetail

Gets the details of a bulk import request.

Usage

Use the BulkOpImportGetRequestDetail method to get the details of a bulk import request, which is identified by the RequestId argument.

Arguments

Table 106 describes the arguments taken by the BulkOpImportGetRequestDetail method.

Table 106. Arguments Taken by the BulkOpImportGetRequestDetail Method

Name	Description	Required	Default	I/O
RequestId	The request ID.	Yes	Not applicable	Input/ Output
ClientVersion	The current data loader client version number.	No	Not applicable	Input
RequestStatus	The status of the request.	Not applicable	Not applicable	Output
RecordType	The record type of the request.	Not applicable	Not applicable	Output

Table 106. Arguments Taken by the BulkOpImportGetRequestDetail Method

Name	Description	Required	Default	I/O
RequestedBy	The requestor of the request.	Not applicable	Not applicable	Output
CompletionTime	The time and date the request completed.	Not applicable	Not applicable	Output
RequestTime	The time and date the request was submitted to be processed.	Not applicable	Not applicable	Output
NumberSubmitted	The number of records to be imported by the request.	Not applicable	Not applicable	Output
NumberProcessed	The number of records processed by the request.	Not applicable	Not applicable	Output
NumberPartialImported	The number of records partially imported by the request.	Not applicable	Not applicable	Output
NumberNotImported	The number of records not imported by the request.	Not applicable	Not applicable	Output
NumberSuccessImported	The number of records successfully imported by the request.	Not applicable	Not applicable	Output
Status	The status of this Web service call.	Not applicable	Not applicable	Output
ErrorMessage	Any error message from this Web service call.	Not applicable	Not applicable	Output

Return Value of the Call

The output arguments.

7 Oracle CRM On Demand Objects Accessible Through Web Services

This chapter contains reference information about the objects accessible through the Web Services On Demand API. These objects correspond to record types and enable access to data stored within an instance of Oracle CRM On Demand. This chapter contains the following topics:

Reference Information About the Parent Objects

The reference information about the parent objects (starting with "Account" on page 255) includes:

- A description of each object, as well as information on usage of the object.
- Information about the relationships between objects; for each object, the associated parent and child objects are listed
- The methods that can be invoked to insert, update, delete, and find data. For more information on these methods, see Chapter 6, "Web Services On Demand API Calls."
- The fields that are accessible for the objects:
  - The required and read-only fields
  - The user keys, see "Oracle CRM On Demand User Keys" on page 252
  - The audit fields, see "Audit Fields" on page 252
  - The status key, see "Oracle CRM On Demand Status Keys" on page 253
  - The pick map fields, see "Oracle CRM On Demand Pick Maps" on page 254

- For Web Services v1.0 only, the filterable fields, see “Filterable Fields” on page 254
- The picklist fields

Oracle CRM On Demand User Keys

A *user key* is a field or group of fields that uniquely identifies a record. Generally, a subset of the record’s fields are used as a user key. However, one field on its own can act as a user key, depending on whether the field can identify the record as unique. Each user key can be used independently to identify a record.

The most basic user key is the single field *Object*Id; for example, for the user object the UserId field is a user key. Every record in the database has at least the following independent user keys:

- *Object*Id
- ExternalSystemId.

In addition, there are various field combinations for different objects that can also be used to define uniqueness.

It is only possible to query for or update a particular record in a table if the values of all the fields in any user key are known. In some instances, the *Object*Id or ExternalSystemId of a record might not be known, but the values for some other user key might be known, in which case the record can be successfully queried or updated using that user key. For example, for a Note child object, the Subject and Description fields form a user key, because they can be used in conjunction with each other to determine whether the record is unique or not. Such a combination is not guaranteed to provide complete uniqueness, but it can be used to query for uniqueness.

The user keys for each object are detailed for each object in “Core Parent Objects (Web Services v1.0)” on page 254, and “Core Parent Objects (Web Services v2.0)” on page 394.

Audit Fields

The audit fields for an object provide information about who created an instance of the object, when it was created, who has last updated an instance of the object, and when it was last updated. All objects, both parent and child level, accessible through the Web services API contain the read-only audit fields contained in Table 107.

Table 107. Audit Fields for the Oracle CRM On Demand Objects

Field Name	Description
CreatedBy	This field is a combination of the full name of the person who created this instance of the object, and the date on which the instance was created. This information is contained within the field in the following format: “`Creator Full Name, CreatedDate`”
CreatedById	The Row ID of the user who created the record.

Table 107. Audit Fields for the Oracle CRM On Demand Objects

Field Name	Description
CreatedDate	The DateTime stamp of when the record was created.
ModifiedBy	This field is a combination of the full name of the person who modified this instance of the object, and the date on which the instance was modified. This information is contained within the field in the following format: *"Modified By Full Name, ModifiedDate"*
ModifiedById	The Row ID of the user who last modified the record.
ModifiedDate	The DateTime stamp of when the record was last modified.

Oracle CRM On Demand Status Keys

An Oracle CRM On Demand *status key* is a field or a number of fields that is returned following an operation on an Oracle CRM On Demand object.

For Web Services v1.0, the status key of objects contained through the Web services API contains all user key and audit fields in addition to some other fields that are identified as status keys for the object. The status keys for the Oracle CRM On Demand objects are outlined in "Core Parent Objects (Web Services v1.0)" on page 254.

For Web Services v2.0, a standard set of status key fields is returned for all parent and child objects:

- Id
- CreatedBy
- CreatedById
- CreatedDate
- ModifiedBy
- ModifiedById
- ModifiedDate
- ModId

This set of fields is standard across all Web Services v2.0 objects as long as the field is available in the associated WSDL file.

The ModId field is the modification key, which can be used with all Web Services v2.0 API methods apart from QueryPage. If this key is specified in the request, it helps check and protect against concurrent updates.

Oracle CRM On Demand Pick Maps

An Oracle CRM On Demand *pick map* allows you to set a foreign key for an object using a different field from the foreign key field.

For example, when updating an account, you might want to set the owner of the account to a specific user. If the UserId of the user is known it can be set in the OwnerId field, which is the foreign key. However, if the UserId is not known, and only the alias of the user is known, that alias can be entered in the Owner field, which is a pick map field. When Oracle CRM On Demand recognizes that the Owner field has been set, it automatically sets the OwnerId field to the UserId for the user.

Pick maps can be used by a number of Oracle CRM On Demand objects to update foreign key references in this way. For each object, a list of pick map fields, and the foreign key fields that they map to, are detailed in "Core Parent Objects (Web Services v1.0)" on page 254 and "Core Parent Objects (Web Services v2.0)" on page 394.

Filterable Fields

A filterable field is a field in which you can apply a search query. For the Web Services V1.0 API, all fields in parent objects are filterable and some fields on child objects are filterable; these fields are shown in the tables of filterable fields for each object in "Core Parent Objects (Web Services v1.0)" on page 254. For the Web Services V2.0 API, all fields in parent objects are filterable.

Core Parent Objects (Web Services v1.0)

This topic include the standard objects of Oracle CRM On Demand. Objects that are only available with industry-specific editions of the application are covered in the following topics:

- "Life Sciences Edition Parent Objects (Web Services v1.0)" on page 368
- "Financial Services Edition Parent Objects (Web Services v1.0)" on page 371
- "Automotive Edition Parent Objects (Web Services v1.0)" on page 381

The following Oracle CRM On Demand objects are detailed in this topic:

- "Account" on page 255
- "Activity" on page 274
- "Asset" on page 283
- "Book" on page 285
- "Campaign" on page 288
- "Contact" on page 297
- "Current User" on page 316
- "CustomObject1 - CustomObject3" on page 318
- "Lead" on page 327

Account

The account object stores information about the companies that you do business with and is also used to track partners and competitors. The methods called on the account object require a list (array) of account objects as an input argument. This list of accounts identifies the records on which the operation is to be carried out.

Usage

It is important to understand the purpose of the following interfaces in the Account Web Service for accessing contact data related to accounts:

- **ListofAccountContact**. Use this interface if you need to access or update a unique account-contact relationship, where there is only one record for each related {Account, Contact} pair.
- **ListOfContactRole**. Use this interface if you need to access or update a unique account-contact-role relationship, where there is only one record for each {Account, Contact, Role} triple. There can be multiple rows for each {Account, Contact} pair (one for each role).
- **ListofContact**. Use this interface for regular account-contact relationships.

Parent Objects

Account, CustomObject1 - CustomObject3

Child Components

Activity, Address, Asset, Attachment, Book, Competitor, Contact, CustomObject3, Lead, Multiple Contact Roles, Note, Opportunity, Team, Related Account, Revenue, Service Request, and Team.

For information about using attachments with this object, see Appendix A, "Using Attachments With Web Services On Demand."

Methods Called

Table 108 details the methods called by the Account service.

Table 108. Methods Called by Account Service

Method	Name as Defined in Service
"Delete" on page 85	AccountDelete
"DeleteChild" on page 88	AccountDeleteChild
"Insert" on page 89	AccountInsert
"InsertChild" on page 90	AccountInsertChild
"InsertOrUpdate" on page 90	AccountInsertOrUpdate
"QueryPage" on page 91	AccountQueryPage
"Update" on page 101	AccountUpdate
"UpdateChild" on page 102	AccountUpdateChild

Fields

Table 109 details the required and read-only fields for the account object.

Table 109. Required and Read-Only Fields for the Account Object

Child Component	Field Name	Type
Account (parent)	AccountName	Required
	AccountConcatField	Read-only
	Audit Fields	Read-only

Table 109. Required and Read-Only Fields for the Account Object

Child Component	Field Name	Type
Activity	AccountLocation	Read-only
	CreatedDetail	Read-only
	MEEventName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	CODInteractionTime	Read-only
	CODWrapUpTime	Read-only
	CODHandleTime	Read-only
	CODIVRTime	Read-only
	CODQueueHoldTime	Read-only
	CODTotalHoldTime	Read-only
	DescriptionShadow	Read-only
	DealerName	Read-only
	Audit Fields	Read-only
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	AccountId	Read-only
	Id	Read-only
	ModId	Read-only
	Audit Fields	Read-only
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	Audit Fields	Read-only

Table 109. Required and Read-Only Fields for the Account Object

Child Component	Field Name	Type
Contact	Age	Read-only
	OwnerFullName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	LastActivityDate	Read-only
	FirstNameShadow	Read-only
	LastNameShadow	Read-only
CustomObject3	AccountCustomObject3CreatedById	Read-only
	AccountCustomObject3CreatedDate	Read-only
	AccountCustomObject3ModifiedById	Read-only
	AccountCustomObject3ModifiedDate	Read-only
	CustomObject3Id	Read-only
Lead	OwnerFullName	Read-only
	SalesRepFullName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	ReferredByFullName	Read-only
	FuriganaAccountName	Read-only
	FuriganaContactFirstName	Read-only
	FuriganaContactLastName	Read-only
	FirstNameShadow	Read-only
	LastNameShadow	Read-only
	CompanyNameShadow	Read-only
	AssignmentStatus	Read-only
	LastAssignmentCompletionDate	Read-only
	LastAssignmentSubmissionDate	Read-only
Multiple Contact Roles	ContactRole	Required
	ContactId	Required
	Audit Fields	Read-only
Opportunity	OwnerFullName	Read-only

Table 109. Required and Read-Only Fields for the Account Object

Child Component	Field Name	Type
	PrimaryRevenueExpectedValue	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	FuriganaAccountName	Read-only
	NameShadow	Read-only
	AssignmentStatus	Read-only
	LastAssignmentCompletionDate	Read-only
	LastAssignmentSubmissionDate	Read-only
RelatedAccount	AccountRelationshipId	Read-only
	RelatedAccountId	Read-only
Revenue	RevenueId	Required
	PartNumber	Required
	Revenue	Required
	ContactFullName	Required
	Audit Fields	Read-only
ServiceRequest	Name	Read-only
	OwnerFullName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	LastAssessmentDate	Read-only
	AssignmentStatus	Read-only
	FuriganaAccountName	Read-only
	FuriganaContactFirstName	Read-only
	FuriganaContactLastName	Read-only
	SRNumberShadow	Read-only
	AbstractShadow	Read-only
	LastAssignmentCompletionDate	Read-only
	LastAssignmentSubmissionDate	Read-only

Table 110 details the status key for the account object, and the child component on which this key resides.

Table 110. Status Key for the Account Object

Child Component	Field Name
Account (parent)	Audit Fields
	ExternalSystemId
	AccountId
	IntegrationId
	LastUpdated
AccountNote	Audit Fields
	ExternalSystemId
	AccountNoteId
	IntegrationId
AccountTeam	Audit Fields
	AccountTeamId
Activity	Audit Fields
	ExternalSystemId
	ActivityId
	IntegrationId
Asset	Audit Fields
	ExternalSystemId
	AssetId
	IntegrationId
Attachment	Audit Fields
	Id
	AccountId
	ModId
Book	Audit Fields
	BookId
	ModId

Table 110. Status Key for the Account Object

Child Component	Field Name
Competitor	Audit Fields
	AccountCompetitorId
	CompetitorExternalId
	CompetitorId
	CompetitorIntegrationId
Contact	Audit Fields
	AccountId
	ContactId
	ExternalSystemId
	IntegrationId
CustomObject3	AccountCustomObject3CreatedById
	AccountCustomObject3CreatedDate
	AccountCustomObject3ModifiedById
	AccountCustomObject3ModifiedDate
Lead	Audit Fields
	AccountId
	ContactId
	ExternalSystemId
	IntegrationId
	LeadId
	OpportunityId
Opportunity	Audit Fields
	AccountId
	ExternalSystemId
	IntegrationId
	OpportunityId
RelatedAccount	Audit Fields
	AccountRelationshipId

Table 110. Status Key for the Account Object

Child Component	Field Name
Revenue	Audit Fields
	ExternalId
	IntegrationId
	RevenueId
ServiceRequest	Audit Fields
	AccountId
	ContactId
	ExternalSystemId
	IntegrationId
	ServiceRequestId
Partner	Audit Fields
	AccountPartnerId
	PartnerExternalId
	PartnerId
	PartnerIntegrationId
	Updated

Table 111 details the pick map fields for the account object and the child objects on which they reside.

Table 111. Pick Map Fields for the Account Object

Child Component	Pick Map Field	Maps To
Account (parent)	Owner	OwnerId
	ParentAccount, ParentAccountLocation	ParentAccountId
	ParentAccountIntegrationId	ParentAccountId
	ParentAccountExternalSystemId	ParentAccountId

Table 111. Pick Map Fields for the Account Object

Child Component	Pick Map Field	Maps To
Activity	AccountIntegrationId	AccountId
	AccountExternalId	AccountId
	CampaignExternalId	CampaignId
	CampaignIntegrationId	CampaignId
	DelegatedBy	DelegatedById
	FundRequest	FundRequestId
	OpportunityIntegrationId	OpportunityId
	OpportunityExternalId	OpportunityId
	PrimaryContactIntegrationId	PrimaryContactId
	PrimaryContactExternalId	PrimaryContactId
	LeadIntegrationId	LeadId
	LeadExternalId	LeadId
	OwnerExternalId	OwnerId
	OwnerIntegrationId	OwnerId
	SRIntegrationId	SRId
	SRExternalId	SRId
	CustomObject1Name	CustomObject1Id
	CustomObject2Name	CustomObject2Id
	CustomObject3Name	CustomObject3Id
Asset	AccountIntegrationId	AccountId
	Manufacturer	ManufacturerId
	PreferredServiceDealer	PreferredServiceDealerID
Book	BookName	BookId
Competitor	RelatedAccountExternalId	RelatedAccountId
	RelatedAccountSystemId	RelatedAccountId

Table 111. Pick Map Fields for the Account Object

Child Component	Pick Map Field	Maps To
Contact	AccountExternalId	AccountId
	AccountIntegrationId	AccountId
	AccountName	AccountId
	ManagerExternalId	ManagerId
	ManagerIntegrationId	ManagerId
	OwnerEmailAddress	OwnerId
	OwnerExternalId	OwnerId
	OwnerIntegrationId	OwnerId
	SourceCampaignName	SourceCampaignId
	SourceCampaignExternalId	SourceCampaignId
	TimeZoneName	TimeZoneId
	CustomObject1Name	CustomObject1Id
	CustomObject2Name	CustomObject2Id
	CustomObject3Name	CustomObject3Id
CustomObject3	Owner	OwnerId

Table 111. Pick Map Fields for the Account Object

Child Component	Pick Map Field	Maps To
Lead	AccountIntegrationId	AccountId
	AccountExternalId	AccountId
	AccountLocation	AccountId
	AccountName	AccountId
	CampaignExternalId	CampaignId
	CampaignIntegrationId	CampaignId
	CampaignName	CampaignId
	ContactExternalId	ContactId
	ContactIntegrationId	ContactId
	OpportunityIntegrationId	OpportunityId
	OpportunityExternalId	OpportunityId
	OwnerExternalId	OwnerId
	OwnerIntegrationId	OwnerId
	SalesRepAlias	SalesRepId
	CustomObject1Name	CustomObject1Id
	CustomObject2Name	CustomObject2Id
	CustomObject3Name	CustomObject3Id
Multiple Contact Roles	ContactIntegrationId	ContactId
	ContactExternalId	ContactId

Table 111. Pick Map Fields for the Account Object

Child Component	Pick Map Field	Maps To
Opportunity	AccountIntegrationId	AccountId
	AccountExternalId	AccountId
	AccountLocation	AccountId
	Account	AccountId
	SourceCampaignExternalId	CampaignId
	SourceCampaign	CampaignId
	KeyContactExternalId	KeyContactId
	KeyContactLastName	KeyContactId
	DealerExternalId	DealerId
	OwnerAlias	OwnerId
	CustomObject1Name	CustomObject1Id
	CustomObject2Name	CustomObject2Id
	CustomObject3Name	CustomObject3Id
Partner	RelatedAccountExternalId	RelatedAccountId
	RelatedAccountIntegrationId	RelatedAccountId
Related Account	RelatedAccountExternalId	RelatedAccountId
	RelatedAccountIntegrationId	RelatedAccountId
Revenue	Product	ProductId
	ProductExternalId	ProductId
	ProductIntegrationId	ProductId
	ProductCategory	ProductCategoryId
	ProductCategoryExternalId	ProductCategoryId
	ProductCategoryIntegrationId	ProductCategoryId
Service Request	AccountIntegrationId	AccountId
	AccountExternalId	AccountId
	AccountLocation	AccountId
	Account	AccountId
	AssetExternalId	AssetId
	Dealer	DealerId
	ContactExternalId	ContactId
	ContactIntegrationId	ContactId

Table 111. Pick Map Fields for the Account Object

Child Component	Pick Map Field	Maps To
	DealerExternalId	DealerId
	OwnerExternalId	OwnerId
	OwnerIntegrationId	OwnerId
	OwnerAlias	OwnerId
	Product	ProductId
	CustomObject1Name	CustomObject1Id
	CustomObject2Name	CustomObject2Id
	CustomObject3Name	CustomObject3Id

Table 112 provides a list of the filterable fields for the child components of the account objects, and a list of the user key combinations for each child component.

Table 112. Filterable Fields and User Key Fields on the Account Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Account (parent)	All	AccountId
		IntegrationId
		ExternalSystemId
		AccountName and Location
Account Note	Subject	Subject and Description
		IntegrationId
		ExternalSystemId
Account Team	FirstName	FirstName and Last Name
	LastName	UserID
	UserID	
	UserRole	
	AccountAccess	
	OpportunityAccess	
	ContactAccess	
	ModifiedDate	
	ModifiedByID	

Table 112. Filterable Fields and User Key Fields on the Account Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Activity	CallType	IntegrationId
		ActivityId
		ExternalSystemId
Address	AddressId	AddressId
	ExternalId	ExternalSystemId
	IntegrationId	IntegrationId
	City	
	Country	
	ModifiedDate	
	Province	
	StateProvince	
	ZipCode	
Asset	AssetId	AssetId
	Contract	IntegrationId
	Date	ExternalSystemId
	ModifiedDate	
	PartNumber	
	ProductCategory	
	Product	
	ProjectManager	
	PurchaseDate	
	Price	
	Quantity	
	SalesRep	
	SerialNumber	
	ShipDate	
	Status	
	Type	
	Warranty	

Table 112. Filterable Fields and User Key Fields on the Account Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Book	None	BookId
		BookName
Contact	AccountContactModifiedById	ExternalSystemId
	AccountContactModifiedDate	IntegrationId
	ContactType	
	ContactFirstName	
	ContactLastName	
	JobTitle	
	Owner	
Competitor	ModifiedDate	CompetitorId
		IntegrationId
		CompetitorExternalSystemId
CustomObject3	AccountCustomObject3ModifiedById	CustomObject3Id
	AccountCustomObject3ModifiedDate	ExternalSystemId
	CustomObject3Id	IntegrationId
	CustomObject3Name	
	ExternalSystemId	
	IntegrationId	
	Type	

Table 112. Filterable Fields and User Key Fields on the Account Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Lead	Campaign	Id
	EstimatedCloseDate	LeadId
	Rating	IntegrationId
	Source	ExternalSystemId
	Status	
	LeadOwner	
	PotentialRevenue	
	ProductInterest	
	SalesPerson	
	LeadId	
	Id	
Multiple Contact Roles	ContactId	ContactId
	ContactExternalId	ContactIntegrationId
	ContactIntegrationId	ContactExternalId
	ContactRole	
	ModifiedDate	
Opportunity	Owner	OpportunityId
	Revenue	Id
	CloseDate	IntegrationId
	Forecast	ExternalSystemId
	ExpectedRevenue	
	Probability	
	Priority	
	ReasonWonLost	
	Status	
	OpportunityId	
	Id	

Table 112. Filterable Fields and User Key Fields on the Account Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Partner	AccountPartnerId	PartnerIntegrationId
	PrimaryContactId	PartnerExternalSystemId
	EndDate	
	PartnerId	
	PartnerExternalSystemId	
	ReverseRelationshipRole	
	RelationshipRole	
	StartDate	
	ModifiedDate	
Related Account	AccountRelationshipId	AccountRelationshipId
	Comments	RelatedAccountId
	EndDate	RelatedAccountExternalId
	ModifiedDate	RelatedAccountIntegrationId
	RelatedAccountExternalId	
	RelatedAccountId	
	RelatedAccountIntegrationId	
	RelationshipRole	
	RelationshipStatus	
	RelationshipType	
	ReverseRelationshipRole	
	StartDate	

Table 112. Filterable Fields and User Key Fields on the Account Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
Revenue	ContactFullName	RevenueId
	Description	ProductId
	ExternalId	ProductExternalId
	IntegrationId	ProductIntegrationId
	Forecast	IntegrationId
	Frequency	ExternalSystemId
	ModifiedDate	
	NumberOfPeriods	
	Product	
	ProductCategoryId	
	ProductCategoryExternalId	
	ProductCategoryIntegrationId	
	ProductExternalId	
	ProductId	
	ProductIntegrationId	
	PurchasePrice	
	Quantity	
	Revenue	
	RevenueId	
	StartCloseDate	
	Status	
	Type	

Table 112. Filterable Fields and User Key Fields on the Account Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Service Request	Subject	Id
	Area	ServiceRequestId
	Owner	IntegrationId
	Priority	ExternalSystemId
	Type	
	Cause	
	Source	
	Status	
	Id	
	ServiceRequestId	

Table 113 details the picklists available for the account object.

Table 113. Picklists Available for the Account Object

Child Component	Field Name
Account (parent)	AccountType
	Priority
	Region
	CallFrequency
	InfluenceType
	Route
	Status
	MarketPotential
	MarketingSegment
Account Team	TeamRole
Competitor	Role
Multiple Contact Roles	ContactRole
Partner	Role
RelatedAccount	Relationship
	Status

Table 113. Picklists Available for the Account Object

Child Component	Field Name
Revenue	Type
	Status
	Frequency

For more information on the fields accessible through the Account Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the account object.

Related Topic

Contact

Activity

The activity object stores information on an activity that a user must carry out, for example, a call-back activity for an account. When an activity is created, the user must set the Activity field explicitly to Task or Appointment.

Usage

Oracle On Demand Web Services uses activities to organize, track, and resolve a variety of tasks, from finding and pursuing opportunities to closing service requests. If a task requires multiple steps that one or more people can carry out, activities greatly simplify the job. Activities can help to:

- Define and assign the task
- Provide information to complete the task
- Track the progress of the task
- Track costs and bill for the task

Parent Objects

Account, Campaign, Contact, Lead, Opportunity, and Service Request

Child Components

Attachment, Book, Contact, ProductsDetailed, SampleDropped, Solution, and User.

For information about using attachments with this object, see Appendix A, “Using Attachments With Web Services On Demand.”

Methods Called

Table 114 details the methods called by the Activity service.

Table 114. Methods Called by Activity Service

Method	Name as Defined in Service
“Delete” on page 85	ActivityDelete
“DeleteChild” on page 88	ActivityDeleteChild
“Insert” on page 89	ActivityInsert
“InsertChild” on page 90	ActivityInsertChild
“InsertOrUpdate” on page 90	ActivityInsertOrUpdate
“QueryPage” on page 91	ActivityQueryPage
“Update” on page 101	ActivityUpdate
“UpdateChild” on page 102	ActivityUpdateChild

Fields

Table 115 details the required and read-only fields for the activity object.

Table 115. Required and Read-Only Fields for the Activity Object

Child Component	Field Name	Type
Activity (parent)	Subject	Required
	Activity	Required
	ActivityId	Read-only
	AddressId	Read-only
	CallType	Read-only
	ContactFirstName	Read-only
	ContactLastName	Read-only
	LeadFirstName	Read-only
	LeadLastName	Read-only
	MedEdEventName	Read-only
	OpportunityName	Read-only
	FundRequest	Read-only
	SmartCall	Read-only
	AssignedQueue	Read-only
	QueueHoldTime	Read-only
	QueueStartTime	Read-only
	TotalHoldTime	Read-only
	ResolutionCode	Read-only
	Audit Fields	Read-only
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	ActivityId	Read-only
	Id	Read-only
	ModId	Read-only
	Audit Fields	Read-only

Table 115. Required and Read-Only Fields for the Activity Object

Child Component	Field Name	Type
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	Audit Fields	Read-only
Contact	ContactId	Read-only
	ContactFirstName	Read-only
	ContactLastName	Read-only
	ContactAccountId	Read-only
	ContactAccountName	Read-only
	ContactAccountLocation	Read-only
	ContactAccountIntegrationId	Read-only
	ContactAccountExternalSystemId	Read-only
	Audit Fields	Read-only
ProductDetailed	ProductId	Required
	Indication	Required
	ProductDetailedId	Read-only
SampleDropped	ProductId	Required
	Quantity	Required
	SampleDroppedId	Read-only
	Audit Fields	Read-only
User	UserId	Read-only
	UserEmail	Read-only
	UserFirstName	Read-only
	UserLastName	Read-only
	UserRole	Read-only
	Audit Fields	Read-only

Table 116 details the status key for the activity object.

Table 116. Status Key for the Activity Object

Child Component	Field Name
Activity (parent)	Audit Fields
	ActivityId
	ExternalSystemId
	IntegrationId
Attachment	Audit Fields
	Id
	ActivityId
Book	Audit Fields
	BookId
	ModId
Contact	Audit Fields
	ActivityContactId
	ContactIntegrationId
	ContactExternalSystemId
	ContactId
User	Audit Fields
	UserId
	UserExternalSystemId
	UserIntegrationId
ProductDetailed	Audit Fields
	ProductDetailedId
	ExternalId
SampleDropped	Audit Fields
	SampleDroppedId
	ExternalId
Solution	Audit Fields
	SolutionId
	ExternalId

Table 117 details the pick map fields for the activity object and the child objects on which they reside.

Table 117. Pick Map Fields for the Activity Object

Child Component	Pick Map Field	Maps To
Activity (parent)	AccountName	AccountId
	AccountLocation	AccountId
	AccountExternalSystemId	AccountId
	AccountIntegration	AccountId
	Owner	OwnerId
	CampaignExternalSystemId	CampaignId
	CampaignIntegrationId	CampaignId
	CampaignName	CampaignId
	LeadExternalSystemId	LeadId
	LeadIntegrationId	LeadId
	MedEdEventExternalSystemId	MedEdEventId
	MedEdEventIntegrationId	MedEdEventId
	OpportunityExternalSystemId	OpportunityId
	OpportunityIntegrationId	OpportunityId
	PortfolioExternalSystemId	PortfolioId
	PortfolioIntegrationId	PortfolioId
	ServiceRequestNumber	ServiceRequestId
	ServiceRequestExternalSystemId	ServiceRequestId
	ServiceRequestIntegrationId	ServiceRequestId
	FundRequestExternalSystemId	FundRequestId
	FundRequestIntegrationId	FundRequestId
Book	BookName	BookId
Contact	ContactExternalId	ContactId
	ContactIntegrationId	ContactId
User	UserExternalSystemId	UserId
	UserIntegrationId	UserId
SampleDropped	ProductIntegrationId	ProductId
	ProductExternalSystemId	ProductId

Table 117. Pick Map Fields for the Activity Object

Child Component	Pick Map Field	Maps To
ProductDetailed	ProductIntegrationId	ProductId
	ProductExternalSystemId	ProductId
Solution	ProductIntegrationId	ProductId
	ProductExternalSystemId	ProductId

Table 118 provides a list of the filterable fields for the child components of the activity objects, and a list of the user key combinations for each child component.

Table 118. Filterable Fields and User Key Fields on the Activity Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Activity (parent)	All	ActivityId
		IntegrationId
		ExternalSystemId
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Book	None	BookId
		BookName
Contact	ContactId	ContactId
	ContactExternalSystemId	ContactExternalSystemId
	ContactIntegrationId	ContactIntegrationId
	ContactFirstName	
	ContactLastName	
	ContactAccountId	
	ContactAccountName	
	ContactAccountLocation	
	ContactAccountIntegrationId	
	ContactAccountExternalSystemId	
	ModifiedDate	

Table 118. Filterable Fields and User Key Fields on the Activity Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
ProductDetailed	ProductDetailedId	ProductDetailedId
	ProductDetailedExternalSystemId	ProductDetailedExternalSystemId
	ProductId	ProductId
	ProductExternalSystemId	ProductExternalSystemId
	ModifiedDate	Name
		Indication
SampleDropped	SampleDroppedId	SampleDroppedId
	SampleDroppedExternalSystemId	SampleDroppedExternalSystemId
	ProductId	ProductId
	ProductExternalSystemId	ProductExternalSystemId
	ModifiedDate	Quantity
		ProductName
Solution	SolutionId	SolutionId
	SolutionExternalSystemId	SolutionExternalSystemId
	ModifiedDate	
	ProductLine	
	PrimaryProductName	
	Name	
	ProductId	
	ProductIntegrationId	
	ProductExternalId	
User	ModifiedDate	UserId
	UserId	UserExternalSystemId
	UserExternalSystemId	UserIntegrationId
	UserIntegrationId	
	UserEmail	
	UserFirstName	
	UserLastName	
	UserRole	

Table 119 details the picklists available for the activity object.

Table 119. Picklists Available for the Activity Object

Field Name
AccountName
AccountLocation
AccountIntegrationId
AccountExternalSystemId
OpportunityName
ServiceRequestNumber
ServiceRequestIntegrationId
ServiceRequestExternalSystemId
DelegatedByExternalSystemId
PrimaryContactIntegrationId
PrimaryContactExternalSystemId
MedEdEventIntegrationId
MedEdEventExternalSystemId
FundRequestExternalId
LeadIntegrationId
LeadExternalSystemId
CampaignIntegrationId
CampaignExternalSystemId
ActivitySubtype
ResolutionCode
PublishInternal
Status
Issue
Indication

For more information on the fields accessible through the Activity Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the activity object.

Asset

The asset object stores information on the assets held by your accounts, for example, the products that an account has purchased. The asset object has no child components.

Usage

Oracle On Demand Web Services uses assets to manage products through their life cycle. It is also used by your accounts to register products, receive product news and literature, track warranty agreements, and receive recommendations on scheduled services.

Parent Objects

Account and Contact.

Methods Called

Table 120 details the methods called by the Asset service.

Table 120. Methods Called by Asset Service

Method	Name as Defined in Service
"Delete" on page 85	AssetDelete
"Insert" on page 89	AssetInsert
"InsertOrUpdate" on page 90	AssetInsertOrUpdate
"QueryPage" on page 91	AssetQueryPage
"Update" on page 101	AssetUpdate

Fields

Table 121 details the required and read-only fields for the asset object.

Table 121. Required and Read-Only Fields for the Asset Object

Child Component	Field Name	Type
Asset (parent)	ProductId	Required
	ProductCategory	Read-only
	PartNumber	Read-only
	Type	Read-only
	Status	Read-only
	Audit Fields	Read-only

Table 122 details the status key for the asset object.

Table 122. Status Key for the Asset Object

Child Component	Field Name
Asset (parent)	Audit Fields
	AssetId
	IntegrationId
	ExternalSystemId

Table 123 details the pick map fields for the asset object.

Table 123. Pick Map Fields for the Asset Object

Child Component	Pick Map Field	Maps To
Asset (parent)	AccountIntegrationId	AccountId
	AccountExternalSystemId	AccountId
	Account, AccountLocation	AccountId
	ProductIntegrationId	ProductId
	ProductExternalSystemId	ProductId
	Product	ProductId

Table 124 provides a list of the filterable fields and a list of user key combinations for the asset object.

Table 124. Filterable Fields and User Key Fields on the Asset Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Asset (parent)	All	AssetId
		IntegrationId
		ExternalSystemId

Table 125 details the picklists available for the asset object.

Table 125. Picklists Available for the Asset Object

Field Name
Warranty
Contract

For more information on the fields accessible through the Asset Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the asset object.

Book

The book object provides a way of segmenting data according to the organizational units of your business, such as territories or products. Administrators can create book hierarchies based on how they want to organize your information, and then set up users to have the appropriate level of access to books.

NOTE: To download the Book WSDL, you must be given access to the Book object. If you do not have access to the Book object, it is not available to download from the Web Services Administration pages. For assistance in gaining access to the Book object, contact your Oracle CRM On Demand service provider.

Parent Objects

Account, Activity, Contact, Custom Object 1 - 3, Lead, Household, Lead, Opportunity, Product Category, Service Request.

Child Components

BookUser, SubBook.

Methods Called

Table 126 details the methods called by the Book service.

Table 126. Methods Called by Book Service

Method	Name as Defined in Service
“Delete” on page 85	BookDelete
“DeleteChild” on page 88	BookDeleteChild
“Insert” on page 89	BookInsert

Table 126. Methods Called by Book Service

Method	Name as Defined in Service
“InsertChild” on page 90	BookInsertChild
“InsertOrUpdate” on page 90	BookInsertOrUpdate
“QueryPage” on page 91	BookQueryPage
“Update” on page 101	BookUpdate
“UpdateChild” on page 102	BookUpdateChild

Fields

Table 127 details the required and read-only fields for the book object.

Table 127. Required and Read-Only Fields for the Book Object

Child Component	Field Name	Type
Book (parent)	BookName	Required
	BookId	Read-only
	Audit Fields	Read-only
BookUser	UserId	Required
	Audit Fields	Read-only
SubBook	UserAlias	Required
	AccessProfileName	Required

Table 128 details the status key for the book object.

Table 128. Status Key for the Book Object

Child Component	Field Name
Book (parent)	Audit Fields
	BookId
	BookName
BookUser	Audit Fields
	BookId
SubBook	BookId
	UserId

Table 129 details the pick map fields for the book object.

Table 129. Pick Map Fields for the Book Object

Child Component	Pick Map Field	Maps To
Book	BookName	BookId
BookUser	BookName	BookId
SubBook	BookName	BookId

Table 130 provides a list of the filterable fields for the child components of the book object, and a list of user key combinations for each child component.

Table 130. Filterable Fields and User Key Fields on the Book Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Book (parent)	All	BookId
		BookName
BookUser	None	BookId
		UserId
		UserExternalSystemId
SubBook	None	BookId
		BookName

Table 131 details the picklists available for the book object.

Table 131. Picklists Available for the Book Object

Child Component	Field Name
Book (parent)	BookType
	ParentBookName
BookUser	UserAlias
	BookRole
	AccessProfileName

For more information on the fields accessible through the Book Web service, go to the Web Services Administration page within the Oracle CRM On Demand application and generate the WSDL file for the book object.

Campaign

The campaign object provides a mechanism for marketing products and services to customers and prospects. The campaign object is the primary way in which new products and services are marketed to customers and prospects.

Parent Objects

Contact, Lead

Child Components

Activity, Attachment, Contact/Recipient, Lead, Note, and Opportunity.

For information about using attachments with this object, see Appendix A, "Using Attachments With Web Services On Demand."

Methods Called

Table 132 details the methods called by the Campaign service.

Table 132. Methods Called by Campaign Service

Method	Name as Defined in Service
"Delete" on page 85	CampaignDelete
"DeleteChild" on page 88	CampaignDeleteChild
"Insert" on page 89	CampaignInsert
"InsertChild" on page 90	CampaignInsertChild
"InsertOrUpdate" on page 90	CampaignInsertOrUpdate
"QueryPage" on page 91	CampaignQueryPage
"Update" on page 101	CampaignUpdate
"UpdateChild" on page 102	CampaignUpdateChild

Fields

Table 133 details the required and read-only fields for the campaign object.

Table 133. Required and Read-Only Fields for the Campaign Object

Child Object	Field Name	Type
Campaign	CampaignName	Required
	SourceCode	Required
	Audit Fields	Read-only
	CreatedByFullName	Read-only
	LastUpdated	Read-only
Activity	AccountLocation	Read-only
	CreatedDetail	Read-only
	MEEventName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	CODInteractionTime	Read-only
	CODWrapUpTime	Read-only
	CODHandleTime	Read-only
	CODIVRTime	Read-only
	CODQueueHoldTime	Read-only
	CODTotalHoldTime	Read-only
	DescriptionShadow	Read-only
	Duration	Read-only
	Audit Fields	Read-only
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	CampaignId	Read-only
	Id	Read-only
	ModId	Read-only
	Audit Fields	Read-only

Table 133. Required and Read-Only Fields for the Campaign Object

Child Object	Field Name	Type
Lead	ContactFirstName	Read-only
	ContactlastName	Read-only
	LeadFullName	Read-only
	OwnerFullName	Read-only
	SalesRepFullName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	ReferredByFullName	Read-only
	FuriganaAccountName	Read-only
	FuriganaContactFirstName	Read-only
	FuriganaContactLastName	Read-only
	FirstNameShadow	Read-only
	LastNameShadow	Read-only
	CompanyNameShadow	Read-only
	AssignmentStatus	Read-only
	LastAssignmentCompletionDate	Read-only
	LastAssignmentSubmissionDate	Read-only
Note	Subject	Required
	Audit Fields	Read-only
Opportunity	OwnerFullName	Read-only
	PrimaryRevenueExpectedValue	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	FuriganaAccountName	Read-only
	ReassignOwnerFlag	Read-only
	NameShadow	Read-only
	AssignmentStatus	Read-only
	LastAssignmentCompletionDate	Read-only
	LastAssignmentSubmissionDate	Read-only
Recipient	ContactID	Required
	ModifiedDate	Read-only

Table 134 details the status key for the campaign object.

Table 134. Status Key for the Campaign Object

Child Component	Field Name
Campaign (parent)	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
Activity	Audit Fields
	CampaignId
	ExternalSystemId
	Id
	IntegrationId
Attachment	Audit Fields
	Id
	CampaignId
CampaignNote	Audit Fields
	CampaignId
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
Lead	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
Opportunity	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated

Table 134. Status Key for the Campaign Object

Child Component	Field Name
Recipient	Audit Fields
	CampaignContactId
	ContactExtrernalId
	ContactIntegrationId
	ContactId

Table 135 details the pick map field for the campaign object.

Table 135. Pick Map Field for the Campaign Object

Child Component	Pick Map Field	Maps To
Campaign (parent)	Owner	OwnerId

Table 135. Pick Map Field for the Campaign Object

Child Component	Pick Map Field	Maps To
Activity	AccountIntegrationId	AccountId
	AccountExternalId	AccountId
	Address	AddressId
	AssignedTo	AssignedToId
	Campaign	CampaignId
	CampaignExternalId	CampaignId
	CampaignIntegrationId	CampaignId
	Dealer	DealerId
	DelegatedBy	DelegatedById
	FundRequest	FundRequestId
	OpportunityIntegrationId	OpportunityId
	OpportunityExternalId	OpportunityId
	PrimaryContactIntegrationId	PrimaryContactId
	PrimaryContactExternalId	PrimaryContactId
	LeadIntegrationId	LeadId
	LeadExternalId	LeadId
	OwnerExternalId	OwnerId
	OwnerIntegrationId	OwnerId
	SRIntegrationId	SRId
	SRExternalId	SRId
	CustomObject1Name	CustomObject1Id
	CustomObject2Name	CustomObject2Id
	CustomObject3Name	CustomObject3Id
Lead	AccountIntegrationId	AccountId
	AccountLocation	AccountId
	CampaignExternalId	CampaignId
	CampaignIntegrationId	CampaignId

Table 135. Pick Map Field for the Campaign Object

Child Component	Pick Map Field	Maps To
	CampaignName	CampaignId
	ContactExternalId	ContactId
	ContactIntegrationId	ContactId
	OpportunityIntegrationId	OpportunityId
	OwnerExternalId	OwnerId
	OwnerIntegrationId	OwnerId
	SalesRepAlias	SalesRepId
	CustomObject1Name	CustomObject1Id
	CustomObject2Name	CustomObject2Id
	CustomObject3Name	CustomObject3Id
Opportunity	AccountIntegrationId	AccountId
	AccountExternalId	AccountId
	AccountLocation	AccountId
	Account	AccountId
	SourceCampaignExternalId	CampaignId
	SourceCampaign	CampaignId
	KeyContactExternalId	KeyContactId
	KeyContactLastName	KeyContactId
	DealerExternalId	DealerId
	OwnerAlias	OwnerId
	Territory	TerritoryId
	CustomObject1Name	CustomObject1Id
	CustomObject2Name	CustomObject2Id
	CustomObject3Name	CustomObject3Id

Table 136 provides a list of the filterable fields for the child components of the campaign objects, and a list of user key combinations for each child component.

Table 136. Filterable Fields and User Key Fields on the Campaign Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Campaign (parent)	All	CampaignId
		IntegrationId
		ExternalSystemID
Activity	Type	Type and Description
	Owner	IntegrationId
	Subject	ExternalSystemId
	DueDate	
	Priority	
	Status	
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
CampaignNote	Subject	Subject and Description
		IntegrationId
		ExternalSystemId
Lead	Campaign	FirstName
	EstimatedCloseDate	Description
	Rating	IntegrationId
	Source	ExternalSystemId
	Status	
	LeadOwner	
	PotentialRevenue	
	ProductInterest	
	SalesPerson	

Table 136. Filterable Fields and User Key Fields on the Campaign Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
Opportunity	Account	OpportunityName
	Owner	IntegrationId
	Revenue	ExternalSystemId
	CloseDate	
	Forecast	
	ExpectedRevenue	
	Probability	
	Priority	
	ReasonWonLost	
	SalesStage	
	Status	
Recipient	ContactId ModifiedDate	ContactId
		ContactIntegrationId
		ContactExternalSystemId
		CampaignContactId

Table 137 details the picklists available for the campaign object.

Table 137. Picklists Available for the Campaign Object

Child Component	Field Name
Campaign (parent)	CampaignType
	Status
Recipient	DeliveryStatus
	ResponseStatus

For more information on the fields accessible through the Campaign Web service, go to the Web Services Administration page within the Oracle CRM On Demand application and generate the WSDL file for the campaign object.

Related Topic

Current User and Opportunity

Contact

The contact object stores information on individuals with whom your organization has a relationship. It allows the user to store information on individuals who are external to your company, but who are associated with the business process. Contacts stored in the Oracle CRM On Demand database can also be associated with an account.

Parent Objects

Account, Activity, Campaign, CustomObject1 - CustomObject3, Opportunity, Product Category, and Life Sciences Edition Parent Objects (Web Services v1.0)

Child Components

Account, Activity, Address, Asset, Attachment, Book, Campaign, CustomObject3, Interests, Lead, Note, Opportunity, Related Contact, Revenue, Service Request and Team.

For information about using attachments with this object, see Appendix A, "Using Attachments With Web Services On Demand."

Methods Called

Table 138 details the methods called by the Contact service.

Table 138. Methods Called by Contact Service

Method	Name as Defined in Service
"Delete" on page 85	ContactDelete
"DeleteChild" on page 88	ContactDeleteChild
"Insert" on page 89	ContactInsert
"InsertChild" on page 90	ContactInsertChild
"InsertOrUpdate" on page 90	ContactInsertOrUpdate
"QueryPage" on page 91	ContactQueryPage
"Update" on page 101	ContactUpdate
"UpdateChild" on page 102	ContactUpdateChild

Fields

Table 139 details the required and read-only fields for the contact object.

Table 139. Required and Read-Only Fields for the Contact Object

Child Component	Field Name	Type
Contact (parent)	FirstName	Required
	LastName	Required
	AlternateAddressId	Read-only
	ContactConcatField	Read-only
	ContactFullName	Read-only
	Audit Fields	Read-only
	Manager	Read-only
	PrimaryAddressId	Read-only
Account	AccountId	Read-only
Activity	AccountLocation	Read-only
	Contact	Read-only
	CreatedDetail	Read-only
	MEEventName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	CODInteractionTime	Read-only
	CODWrapUpTime	Read-only
	CODHandleTime	Read-only
	CODIVRTime	Read-only
	CODQueueHoldTime	Read-only
	CODTotalHoldTime	Read-only
	DescriptionShadow	Read-only
	Audit Fields	Read-only
Address	AddressId	Read-only

Table 139. Required and Read-Only Fields for the Contact Object

Child Component	Field Name	Type
Asset	AssetId	Required
	ContactAssetId	Read-only
	ExternalSystemId	Read-only
	Product	Read-only
	ProductId	Read-only
	ProductPartNumber	Read-only
	ProductType	Read-only
	ProductStatus	Read-only
	CreatedByandDate	Read-only
	ModifiedByandDate	Read-only
	SerialNumber	Read-only
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	ContactId	Read-only
	Id	Read-only
	ModId	Read-only
	Audit Fields	Read-only
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	Audit Fields	Read-only
Campaign	CampaignContactId	Read-only
	Audit Fields	Read-only
Contact Team	ContactTeamId	Read-only
	UserFirstName	Read-only
	UserLastName	Read-only
	UserRole	Read-only

Table 139. Required and Read-Only Fields for the Contact Object

Child Component	Field Name	Type
CustomObject3	ContactCustomObject3CreatedById	Read-only
	ContactCustomObject3CreatedDate	Read-only
	ContactCustomObject3ModifiedById	Read-only
	ContactCustomObject3ModifiedDate	Read-only
	CustomObject3Id	Read-only
Interests	Category	Required
	Interests	Required
	InterestId	Read-only
Lead	ContactIntegrationId	Read-only
	ContactExternalId	Read-only
	OwnerFullName	Read-only
	SalesRepFullName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	ReferredByFullName	Read-only
	FuriganaAccountName	Read-only
	FuriganaContactFirstName	Read-only
	FuriganaContactLastName	Read-only
	FirstNameShadow	Read-only
	LastNameShadow	Read-only
	CompanyNameShadow	Read-only
	AssignmentStatus	Read-only
	LastAssignmentCompletionDate	Read-only
	LastAssignmentSubmissionDate	Read-only

Table 139. Required and Read-Only Fields for the Contact Object

Child Component	Field Name	Type
Opportunity	OwnerFullName	Read-only
	PrimaryRevenueExpectedValue	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	FuriganaAccountName	Read-only
	ReassignOwnerFlag	Read-only
	NameShadow	Read-only
	AssignmentStatus	Read-only
	LastAssignmentCompletionDate	Read-only
	LastAssignmentSubmissionDate	Read-only
Related Contact	ContactRelationshipId	Read-only
	ContactId	Read-only
	RelatedContactFirstName	Read-only
	RelatedContactLastName	Read-only
Revenue	RevenueId	Required
	PartNumber	Required
	Revenue	Required
	ContactFullName	Required
	Audit Fields	Read-only

Table 139. Required and Read-Only Fields for the Contact Object

Child Component	Field Name	Type
ServiceRequest	Name	Read-only
	OwnerFullName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	LastAssessmentDate	Read-only
	AssignmentStatus	Read-only
	FuriganaAccountName	Read-only
	FuriganaContactFirstName	Read-only
	FuriganaContactLastName	Read-only
	SRNumberShadow	Read-only
	AbstractShadow	Read-only
	LastAssignmentCompletionDate	Read-only
	LastAssignmentSubmissionDate	Read-only

Table 140 details the status key for the contact object.

Table 140. Status Key for the Contact Object

Child Component	Field Name
Contact (parent)	Audit Fields
	AccountId
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
Account	Audit Fields
	Name and Location
	AccountExternalId
Activity	Audit Fields
	ActivityExternalId
	ActivityId
	ActivityIntegrationId

Table 140. Status Key for the Contact Object

Child Component	Field Name
Asset	Audit Fields
Attachment	Audit Fields
	Id
	ContactId
Book	Audit Fields
	BookId
	ModId
Campaign	Audit Fields
	CampaignContactId
	CampaignExternalSystemId
	CampaignId
	ModId
ContactNote	Audit Fields
	ContactId
	ExternalSystemId
	Id
	IntegrationId
CustomObject3	CustomObject3Id
	ContactCustomObject3CreatedById
	ContactCustomObject3CreatedDate
	ContactCustomObject3ModifiedById
	ContactCustomObject3ModifiedDate
Interests	Audit Fields
	InterestId
	ExternalSystemId

Table 140. Status Key for the Contact Object

Child Component	Field Name
Lead	Audit Fields
	AccountId
	CampaignId
	ContactId
	ExternalSystemId
	IntegrationId
	LeadId
	OpportunityId
Opportunity	Audit Fields
	AccountId
	ExternalSystemId
	IntegrationId
	LeadId
	OpportunityId
RelatedContact	Audit Fields
	ContactRelationshipId
	RelatedContactId
Revenue	Audit Fields
	ExternalId
	IntegrationId
	RevenueId
ServiceRequest	Audit Fields
	AccountId
	ContactID
	ExternalSystemId
	IntegrationId
	ServiceRequestId

Table 140. Status Key for the Contact Object

Child Component	Field Name
Team	Audit Fields
	ContactTeamId
	UserExternalSystemId
	UserId
	UserIntegrationId

Table 141 details the pick map fields for the contact object.

Table 141. Pick Map Fields for the Contact Object

Child Component	Pick Map Field	Maps To
Contact (parent)	AccountName	AccountId
	Owner	AssignedToAlias
	SourceCampaignName	SourceCampaignId
	ManagerExternalSystemId	ManagerId
Account	AccountExternalId	AccountId
Activity	AccountIntegrationId	AccountId
	AccountExternalId	AccountId
	AssignedTo	AssignedToId
	CampaignExternalId	CampaignId
	CampaignIntegrationId	CampaignId
	DelegatedBy	DelegatedById
	Dealer	DealerId
	FundRequest	FundRequestId
	OpportunityIntegrationId	OpportunityId
	OpportunityExternalId	OpportunityId
	PrimaryContactIntegrationId	PrimaryContactId
	PrimaryContactExternalId	PrimaryContactId
	LeadIntegrationId	LeadId
	LeadExternalId	LeadId
	OwnerExternalId	OwnerId
	OwnerIntegrationId	OwnerId

Table 141. Pick Map Fields for the Contact Object

Child Component	Pick Map Field	Maps To
	SRIntegrationId	SRId
	SRExternalId	SRId
	CustomObject1Name	CustomObject1Id
	CustomObject2Name	CustomObject2Id
	CustomObject3Name	CustomObject3Id
Asset	ExternalSystemId	AssetId
	IntegrationId	AssetId
	AccountIntegrationId	AccountId
	AccountExternalId	AccountId
	AccountLocation	AccountId
	AccountName	AccountId
	CustomerContactExternalSystemId	ContactId
	Manufacturer	ManufacturerId
	ProductExternalSystemId	ProductId
	ProductPrimaryProductLine	ProductPrimaryProductLineId
	PreferredServiceDealer	PreferredServiceDealerId
Book	BookName	BookId
Campaign	CampaignExternalSystemId	CampaignId
	CampaignName	CampaignId
Contact Team	UserExternalSystemId	UserId
	UserIntegrationId	UserId
CustomObject3	Owner	OwnerId
Interests	InterestExternalSystemId	InterestId

Table 141. Pick Map Fields for the Contact Object

Child Component	Pick Map Field	Maps To
Lead	AccountIntegrationId	AccountId
	AccountLocation	AccountId
	CampaignExternalId	CampaignId
	CampaignIntegrationId	CampaignId
	CampaignName	CampaignId
	OpportunityIntegrationId	OpportunityId
	OpportunityExternalId	OpportunityId
	OwnerExternalId	OwnerId
	OwnerIntegrationId	OwnerId
	SalesRepAliasId	SalesRepId
	CustomObject1Name	CustomObject1Id
	CustomObject2Name	CustomObject2Id
	CustomObject3Name	CustomObject3Id
Opportunity	AccountIntegrationId	AccountId
	AccountExternalId	AccountId
	AccountLocation	AccountId
	Account	AccountId
	SourceCampaignExternalId	CampaignId
	SourceCampaign	CampaignId
	KeyContactExternalId	KeyContactId
	KeyContactLastName	KeyContactId
	DealerExternalId	DealerId
	OwnerAlias	OwnerId
	SalesStage	SalesStageId
	Territory	TerritoryId
	CustomObject1Name	CustomObject1Id
	CustomObject2Name	CustomObject2Id
	CustomObject3Name	CustomObject3Id
Related Contact	RelatedContactExternalId	RelatedContactId
	RelatedContactIntegrationId	RelatedContactId

Table 141. Pick Map Fields for the Contact Object

Child Component	Pick Map Field	Maps To
Revenue	Product	ProductId
	ProductExternalId	ProductId
	ProductIntegrationId	ProductId
	ProductCategory	ProductCategoryId
	ProductCategoryExternalId	ProductCategoryId
	ProductCategoryIntegrationId	ProductCategory
Service Request	AccountIntegrationId	AccountId
	AccountExternalId	AccountId
	AccountLocation	AccountId
	Account	AccountId
	AssetExternalId	AssetId
	Dealer	DealerId
	ContactExternalId	ContactId
	ContactIntegrationId	ContactId
	OwnerExternalId	OwnerId
	OwnerIntegrationId	OwnerId
	OwnerAlias	OwnerId
	Product	ProductId
	CustomObject1Name	CustomObject1Id
	CustomObject2Name	CustomObject2Id
	CustomObject3Name	CustomObject3Id

Table 142 provides a list of the filterable fields for the child components of the contact objects, and a list of user key combinations for each child component.

Table 142. Filterable Fields and User Key Fields on the Contact Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Contact (parent)	All	ContactId
		IntegrationId
		ExternalSystemId

Table 142. Filterable Fields and User Key Fields on the Contact Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Account	AccountId	AccountId
	AccountExternalId	AccountExternalId
	Location	Name and Location
	Name	
Activity	CallType	IntegrationId
		ActivityID
		ExternalSystemId
Address	ExternalId	AddressID
	IntegrationId	ExternalSystemId
	City	IntegrationId
	Country	
	ZipCode	
	StateProvince	
	Province	
Asset	AssetId	AssetId
	ExternalSystemId	ExternalSystemId
	IntegrationId	IntegrationId
	ProductId	
	SerialNumber	
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Book	None	BookId
		BookName
Campaign	CampaignContactId	CampaignContactId
	CampaignId	CampaignId
	CampaignExternalSystemId	CampaignExternalSystemId
	CampaignName	CampaignName
	DeliveryStatus	
	ResponseStatus	
	ModifiedDate	

Table 142. Filterable Fields and User Key Fields on the Contact Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Contact Note	Subject	Subject and Description
		IntegrationId
		ExternalSystemId
Contact Team	ContactTeamId	UserId
	UserId	UserIntegrationId
	UserExternalSystemId	UserExternalSystemId
	UserIntegrationId	
	UserFirstName	
	UserLastName	
	ContactAccess	
	UserRole	
CustomObject3	ContactCustomObject3ModifiedById	CustomObject3Id
	ContactCustomObject3ModifiedDate	ExternalSystemId
	CustomObject3Id	IntegrationId
	CustomObject3Name	
	ExternalSystemId	
	IntegrationId	
	Type	
Interests	InterestId	ExternalSystemId
	Category	InterestId
	Interests	

Table 142. Filterable Fields and User Key Fields on the Contact Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
Lead	Campaign	IntegrationId
	EstimatedCloseDate	ExternalSystemId
	Rating	LeadId
	Source	
	Status	
	LeadOwner	
	PotentialRevenue	
	ProductInterest	
	SalesPerson	
	LeadId	
	OpportunityId	
Opportunity	Opportunity	IntegrationId
	OpportunityId	ExternalSystemId
	Owner	OpportunityId
	Revenue	
	CloseDate	
	Forecast	
	ExpectedRevenue	
	Probability	
	Priority	
	ReasonWonLost	
	SalesStage	
	Status	
	Account	

Table 142. Filterable Fields and User Key Fields on the Contact Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Personal Address	PersonalAddressIntegrationId	ExternalSystemId
	PersonalAddressName	
	AlternateCity	
	AlternateCountry	
	AlternateZipCode	
	AlternateStateProvince	
	AlternateAddress	
	AlternateAddress2	
	AlternateAddress3	
	Id	
	IntegrationId	
	AddressName	
	City	
	Country	
	ZipCode	
	StateProvince	
	Address	
	ShippingAddress2	
Related Contact	ContactRelationshipId	IntegrationId
	RelatedContactId	RelatedContactExternalId
	RelatedContactExternalId	
	RelatedContactIntegrationId	
	RelationshipStatus	
	RelationshipType	
	StartDate	
	EndDate	
	Description	
	RelationshipRole	
	ReverseRelationshipRole	

Table 142. Filterable Fields and User Key Fields on the Contact Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
Revenue	ContactFullName	RevenueId
	Description	ExternalId
	ExternalId	IntegrationId
	Forecast	ExternalSystemId
	Frequency	
	IntegrationId	
	ModifiedDate	
	NumberOfPeriods	
	Product	
	ProductId	
	ProductCategoryId	
	ProductCategory	
	ProductCategoryExternalId	
	ProductCategoryIntegrationId	
	ProductExternalId	
	ProductIntegrationId	
	PurchasePrice	
	Quantity	
	Revenue	
	RevenueId	
	Status	
	StartCloseDate	
	Type	

Table 142. Filterable Fields and User Key Fields on the Contact Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Service Request	Subject	SRNumber
	Area	IntegrationId
	Owner	ExternalSystemId
	Priority	
	Type	
	Cause	
	Source	
	Status	
	ServiceRequestId	

Table 143 details the picklists available for the contact object.

Table 143. Picklists Available for the Contact Object

Child Component	Field Name
Contact (parent)	ContactType
	LeadSource
	MrMrs
	BestTimeToCall
	CallFrequency
	CurrentInvestmentMix
	Degree
	ExperienceLevel
	Gender
	InvestmentHorizon
	LifeEvent
	MaritalStatus
	MarketPotential
	Objective
	OwnOrRent
	PrimaryGoal
	RiskProfile
	Route
	Segment
	Tier
Account	Call Frequency
	Route
	Status
	Type
Contact Team	TeamRole
Interests	Category
	Subjects

Table 143. Picklists Available for the Contact Object

Child Component	Field Name
Related Contact	Relationship
	Status
Revenue	Type
	Status
	Frequency

Table 144 details a number of contact object fields that you must not use for customer integrations.

Table 144. Contact Object Fields That You Must Not Use

Field Name
PartyTypeCode
PartyUId
PersonUId

For more information on the fields accessible through the Contact Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the contact object.

Current User

The current user object stores information on the currently logged-in user.

Child Components

Login History and Quota

Methods Called

Table 145 details the method called by the Current user service, and its name as defined on the service.

Table 145. Methods Called by Current User Service

Method	Name as Defined in Service
"QueryPage" on page 91	CurrentUserQueryPage

Fields

Table 146 details the required and read-only fields for the current user object.

Table 146. Required and Read-Only Fields for the Current User Object

Child Component	Field Name	Type
Current User (parent)	FirstName	Required
	LastName	Required
	CreatedBy	Read-only
	ModifiedBy	Read-only
Login History	LastLoggedIn	Read-only

Table 147 provides a list of the filterable fields for the child components of the current user objects, and a list of user key combinations for each child component.

Table 147. Filterable Fields on the Current User Object's Child Components

Child Component	Filterable Fields	User Key Field Combinations
Current User (parent)	All	CurrentUserId
		ExternalSystemId
		FirstName and LastName and Middlename
Login History	Alias	None
	SourceIPAddress	
	SignInStatus	
	SignInTime	

For more information on the fields accessible through the Current user Web service, go to the Web Services Administration page within the Oracle CRM On Demand application and generate the WSDL file for the current user object.

Related Topic

User

CustomObject1 - CustomObject3

The CustomObject1, CustomObject2, and CustomObject3 services expose the functionality of the CustomObject1 - CustomObject3 objects to external applications.

TIP: The reference information for each of CustomObject1, CustomObject2, and CustomObject3 follows the same pattern. In this topic, the information for CustomObject1 is given as an example.

NOTE: To download the CustomObject1 - CustomObject3 WSDL file, you must be given access to the relevant CustomObject object. If you do not have access to the CustomObject object, it is not available to download from the Web Services Administration page or available to use Web service calls. For assistance in gaining access to the CustomObject1 - CustomObject3 objects, contact your Oracle CRM On Demand service provider.

Parent Objects

Account, Contact

Child Components

Account, Attachment, Book, Contact, Opportunity, Portfolio, and Team.

CustomObject1, CustomObject2, and CustomObject3 also have child components for the other CustomObjects as follow:

- **CustomObject1.** CustomObject2 and CustomObject3 child components.
- **CustomObject2.** CustomObject1 and CustomObject3 child components.
- **CustomObject3.** CustomObject1 and CustomObject2 child components.

For information about using attachments with these objects, see Appendix A, “Using Attachments With Web Services On Demand.”

NOTE: For CustomObject3 only, the following fields are accessible: CustomObject5Id, CustomObject5Name, CustomObject5ExternalSystemId.

Methods Called

Table 148 details the methods called by the CustomObject1 service. The methods for CustomObject2 and CustomObject3 follow the same pattern.

Table 148. Methods Called by CustomObJect1 Service

Method	Name as Defined in Service
“Delete” on page 85	CustomObject1Delete
“DeleteChild” on page 88	CustomObject1DeleteChild
“Insert” on page 89	CustomObject1Insert
“InsertChild” on page 90	CustomObject1InsertChild

Table 148. Methods Called by CustomObJect1 Service

Method	Name as Defined in Service
"InsertOrUpdate" on page 90	CustomObject1OrUpdate
"QueryPage" on page 91	CustomObject1Page
"Update" on page 101	CustomObject1Update
"UpdateChild" on page 102	CustomObject1UpdateChild

Fields

Table 149 details the required and read-only fields for the CustomObject1 object. The fields for CustomObject2 and CustomObject3 follow a similar pattern.

Table 149. Required and Read-Only Fields for the CustomObject1 Object

Child Component	Field Name	Type
CustomObject1 (parent)	ParentId	Required
	ParentExternalSystemID	Required
	ParentIntegrationId	Required
	ParentObject1Id	Read-only
Account	CObj1AccountCreatedById	Read-only
	CObj1AccountCreatedDate	Read-only
	CObj1AccountModifiedById	Read-only
	CObj1AccountModifiedDate	Read-only
	Region	Read-only
	AccountType	Read-only
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	CustomObjectId	Read-only
	Id	Read-only
	ModId	Read-only
	Audit Fields	Read-only

Table 149. Required and Read-Only Fields for the CustomObject1 Object

Child Component	Field Name	Type
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	Audit Fields	Read-only
Contact	CObj1ContactCreatedById	Read-only
	CObj1ContactCreatedDate	Read-only
	CObj1ContactModifiedById	Read-only
	CObj1ContactModifiedDate	Read-only
	ContactFirstName	Read-only
	ContactLastName	Read-only
	ContactType	Read-only
CustomObject2	CObj1CustomObject2CreatedById	Read-only
	CObj1CustomObject2CreatedDate	Read-only
	CObj1CustomObject2ModifiedById	Read-only
	CObj1CustomObject2ModifiedDate	Read-only
	CustomObject2Id	Read-only
CustomObject3	CObj1CustomObject3CreatedById	Read-only
	CObj1CustomObject3CreatedDate	Read-only
	CObj1CustomObject3ModifiedById	Read-only
	CObj1CustomObject3ModifiedDate	Read-only
	CustomObject3Id	Read-only
Opportunity	AccountName	Read-only
	CObj1OpportunityCreatedById	Read-only
	CObj1OpportunityCreatedDate	Read-only
	CObj1OpportunityModifiedById	Read-only
	CObj1OpportunityModifiedDate	Read-only
	OpportunityName	Read-only
	Revenue	Read-only
	SalesStage	Read-only

Table 149. Required and Read-Only Fields for the CustomObject1 Object

Child Component	Field Name	Type
Portfolio	AccountNumber	Read-only
	CObj1PortfolioCreatedById	Read-only
	CObj1PortfolioCreatedDate	Read-only
	CObj1PortfolioModifiedById	Read-only
	CObj1PortfolioModifiedDate	Read-only
	Revenue	Read-only
Team	CustomObject1TeamId	Read-only
	UserFirstName	Read-only
	UserLastName	Read-only

Table 150 details the status key for the CustomObject1 object. The status keys for CustomObject2 and CustomObject3 follow a similar pattern.

Table 150. Status Key for the CustomObject1 Object

Child Component	Field Name
CustomObject1 (parent)	Audit Fields
	CustomObject1Id
	ExternalSystemId
	IntegrationId
Account	CustomObject1AccountId
	CObj1AccountCreatedById
	CObj1AccountCreatedDate
	CObj1AccountModifiedById
	CObj1AccountModifiedDate
Attachment	Audit Fields
	Id
	CustomObjectId
Book	Audit Fields
	BookId
	ModId

Table 150. Status Key for the CustomObject1 Object

Child Component	Field Name
Contact	CustomObject1ContactId
	CObj1ContactCreatedById
	CObj1ContactCreatedDate
	CObj1ContactModifiedById
	CObj1ContactModifiedDate
CustomObject2	CustomObject2Id
	CObj1CustomObject2CreatedById
	CObj1CustomObject2CreatedDate
	CObj1CustomObject2ModifiedById
	CObj1CustomObject2ModifiedDate
CustomObject3	CustomObject3Id
	CObj1CustomObject3CreatedById
	CObj1CustomObject3CreatedDate
	CObj1CustomObject3ModifiedById
	CObj1CustomObject3ModifiedDate
Opportunity	OpportunityId
	CObj1OpportunityCreatedById
	CObj1OpportunityCreatedDate
	CObj1OpportunityModifiedById
	CObj1OpportunityCreatedDate
Portfolio	PortfolioId
	CObj1PortfolioCreatedById
	CObj1PortfolioCreatedDate
	CObj1PortfolioModifedById
	CObj1PortfolioModifiedDate
Team	Audit Fields
	CustomObject1TeamId

Table 151 details the pick map fields for the CustomObject1 object. The fields for CustomObject2 and CustomObject3 follow a similar pattern.

Table 151. Pick Map Fields for the CustomObject1 Object

Child Component	Pick Map Field	Maps To
CustomObject1 (parent)	AccountExternalId	AccountId
	AccountIntegrationId	AccountId
	AccountName	AccountId
	ActivityExternalId	ActivityId
	ActivityIntegrationId	ActivityId
	ActivityName	ActivityId
	CampaignExternalId	CampaignId
	CampaignIntegrationId	CampaignId
	CampaignName	CampaignId
	ContactExternalId	ContactId
	ContactFirstName	ContactId
	ContactFullName	ContactId
	ContactIntegrationId	ContactId
	ContactLastName	ContactId
	CustomObject2ExternalId	CustomObject2Id
	CustomObject2IntegrationId	CustomObject2Id
	CustomObject2Name	CustomObject2Id
	CustomObject3ExternalId	CustomObject3Id
	CustomObject3IntegrationId	CustomObject3Id
	CustomObject3Name	CustomObject3Id
	DealerName	DealerId
	HouseholdExternalId	HouseholdId
	HouseholdIntegrationId	HouseholdId
	HouseholdName	HouseholdId
	LeadExternalId	LeadId
	LeadFirstName	LeadId
	LeadFullName	LeadId
	LeadIntegrationId	LeadId

Table 151. Pick Map Fields for the CustomObject1 Object

Child Component	Pick Map Field	Maps To
CustomObject1 (cont.)	LeadLastName	LeadId
	Owner	OwnerId
	OpportunityExternalId	OpportunityId
	OpportunityIntegrationId	OpportunityId
	OpportunityName	OpportunityId
	ParentExternalSystemId	ParentId
	ParentIntegrationId	ParentId
	PortfolioAccountNumber	PortfolioId
	ProductExternalId	ProductId
	ProductIntegrationId	ProductId
	ProductName	ProductId
	SolutionExternalId	SolutionId
	SolutionIntegrationId	SolutionId
	SolutionTitle	SolutionId
	ServiceRequestExternalId	ServiceRequestId
	ServiceRequestIntegrationId	ServiceRequestId
	ServiceRequestName	ServiceRequestId
	VIN	VehicleId
Account	ExternalSystemId	AccountId
	IntegrationId	AccountId
	Location	AccountId
	Name	AccountId
Book	BookName	BookId
Contact	ExternalSystemId	ContactId
	IntegrationId	ContactId
CustomObject2	Owner	OwnerId
CustomObject3	Owner	OwnerId
Opportunity	ExternalSystemId	OpportunityId
	IntegrationId	OpportunityId
Portfolio	ExternalSystemId	PortfolioId
	IntegrationId	PortfolioId

Table 151. Pick Map Fields for the CustomObject1 Object

Child Component	Pick Map Field	Maps To
Team	UserExternalSystemId	UserId
	UserIntegrationId	UserId
	UserEmail	UserId

Table 152 provides a list of the filterable fields for the child components of the CustomObject1 object, and a list of user key combinations for each child component. The fields for CustomObject2 and CustomObject3 follow a similar pattern.

Table 152. Filterable Fields and User Key Fields on the CustomObject1 Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
CustomObject1 (parent)	All	CustomObject1Id
		ExternalSystemId
		IntegrationId
Account	AccountId	CustomObject1AccountId
	AccountType	ExternalSystemId
	CObj1AccountModifiedById	IntegrationId
	CObj1AccountModifiedDate	
	ExternalSystemId	
	IntegrationId	
	Location	
	Name	
	Region	
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Book	None	BookId
		BookName

Table 152. Filterable Fields and User Key Fields on the CustomObject1 Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Contact	ContactId	CustomObject1ContactId
	CObj1ContactModifiedById	ExternalSystemId
	CObj1ContactModifiedDate	IntegrationId
	ContactType	
	ExternalSystemId	
	IntegrationId	
CustomObject2	CustomObject2Id	CustomObject2Id
	CObj1CustomObject2ModifiedById	ExternalSystemId
	CObj1CustomObject2ModifiedDate	IntegrationId
	ExternalSystemId	
	IntegrationId	
	Name	
	Type	
CustomObject3	CustomObject3Id	CustomObject3Id
	CObj1CustomObject3ModifiedById	ExternalSystemId
	CObj1CustomObject3ModifiedDate	IntegrationId
	ExternalSystemId	
	IntegrationId	
	Name	
	Type	
Opportunity	CObj1OpportunityModifiedById	OpportunityId
	CObj1OpportunityModifiedDate	ExternalSystemId
	ExternalSystemId	IntegrationId
	IntegrationId	
	OpportunityId	
Portfolio	CObj1PortfolioModifiedById	PortfolioId
	CObj1PortfolioModfiedDate	ExternalSystemId
	ExternalSystemId	IntegrationId
	IntegrationId	
	PortfolioId	

Table 152. Filterable Fields and User Key Fields on the CustomObject1 Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Team	CustomObject1TeamId	CustomObject1TeamId
	UserEmail	UserExternalSystemId
	UserExternalSystemId	UserIntegrationId
	UserId	
	UserIntegrationId	

Table 153 details the picklists available for the CustomObject1 object. The fields for CustomObject2 and CustomObject3 follow a similar pattern.

Table 153. Picklists Available for the CustomObject1 Object

Child Component	Field Name
Dealer	Type
Household	Type
Portfolio	Type
Vehicle	Type

Lead

The lead object stores information on a company or individual with whom an opportunity can be created. It allows the user to identify the companies that might be interested in a product or service. Leads are usually generated as part of a marketing campaign.

Parent Objects

Account, Campaign, Contact, and Opportunity

Child Components

Activity, Attachment, Book, Campaign.

For information about using attachments with this object, see Appendix A, "Using Attachments With Web Services On Demand."

Methods Called

Table 154 details the methods called by the Lead service.

Table 154. Methods Called by Lead Service

Method	Name as Defined in Service
"Delete" on page 85	LeadDelete
"DeleteChild" on page 88	LeadDeleteChild
"Insert" on page 89	LeadInsert
"InsertChild" on page 90	LeadInsertChild
"InsertOrUpdate" on page 90	LeadInsertOrUpdate
"QueryPage" on page 91	LeadQueryPage
"Update" on page 101	LeadUpdate
"UpdateChild" on page 102	LeadUpdateChild

Fields

Table 155 details the required and read-only fields for the lead object.

Table 155. Required and Read-Only Fields for the Lead Object

Child Component	Field Name	Type
Lead (parent)	FirstName	Required
	LastName	Required
	LeadOwner	Required
	ContactFullName	Read-only
	Audit Fields	Read-only
	LastUpdated	Read-only
	LeadConcatField	Read-only
	LeadFullName	Read-only
	ReferredById	Read-only

Table 155. Required and Read-Only Fields for the Lead Object

Child Component	Field Name	Type
Activity	AccountLocation	Read-only
	CreatedDetail	Read-only
	Lead	Read-only
	LeadExternalId	Read-only
	MEEventName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	CODInteractionTime	Read-only
	CODWrapUpTime	Read-only
	CODHandleTime	Read-only
	CODIVRTime	Read-only
	CODQueueHoldTime	Read-only
	CODTotalHoldTime	Read-only
	DescriptionShadow	Read-only
	Duration	Read-only
	Audit Fields	Read-only
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	LeadId	Read-only
	Id	Read-only
	ModId	Read-only
	Audit Fields	Read-only
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	Audit Fields	Read-only

Table 156 details the status key for the lead object.

Table 156. Status Key for the Lead Object

Child Component	Field Name
Lead (parent)	Audit Fields
	AccountId
	CampaignId
	ContactId
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
	OpportunityId
Activity	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
	LeadId
Attachment	Audit Fields
	Id
	LeadId
Book	Audit Fields
	BookId
	ModId

Table 157 details the pick map fields for the lead object.

Table 157. Pick Map Fields for the Lead Object

Child Component	Pick Map Field	Maps To
Lead (parent)	Campaign	CampaignId
	OpportunityName	OpportunityId
	Owner	OwnerId
	AccountExternalSystemId	AccountId
	OpportunityExternalSystemId	OpportunityId
	ContactExternalSystemId	ContactId
	CampaignExternalSystemId	CampaignId
	ReferredByExternalSystemId	ReferredById
Activity	AccountIntegrationId	AccountId
	AccountExternalId	AccountId
	Address	AddressId
	AssignedTo	AssignedToId
	CampaignExternalId	CampaignId
	CampaignIntegrationId	CampaignId
	DelegatedBy	DelegatedById

Table 157. Pick Map Fields for the Lead Object

Child Component	Pick Map Field	Maps To
	Dealer	DealerId
	FundRequest	FundRequestId
	OpportunityIntegrationId	OpportunityId
	OpportunityExternalId	OpportunityId
	PrimaryContactIntegrationId	PrimaryContactId
	PrimaryContactExternalId	PrimaryContactId
	LeadIntegrationId	LeadId
	LeadExternalId	LeadId
	OwnerExternalId	OwnerId
	OwnerIntegrationId	OwnerId
	SRIntegrationId	SRId
	SRExternalId	SRId
	CustomObject1Name	CustomObject1Id
	CustomObject2Name	CustomObject2Id
	CustomObject3Name	CustomObject3Id
Book	BookName	BookId

Table 158 provides a list of the filterable fields for the child components of the lead object, and a list of user key combinations for each child component.

Table 158. Filterable Fields and User Key Fields on the Lead Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Lead (parent)	All	LeadId
		IntegrationId
		ExternalSystemId
		LeadFirstName and LeadLastName
		Description

Table 158. Filterable Fields and User Key Fields on the Lead Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Activity	Type	Type and Description
	Owner	IntegrationId
	Subject	ExternalSystemId
	DueDate	
	Priority	
	Status	
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Book	None	BookId
		BookName
Partner	LeadPartnerId	PartnerIntegrationId
	PartnerId	PartnerExternalSystemId
	PartnerExternalSystemId	
	CreatedDate	
	ModifiedDate	

Table 159 details the picklists available for the lead object.

Table 159. Picklists Available for the Lead Object

Field Name
Country
MrMrs
Rating
Source
StateProvince
Status

For more information on the fields accessible through the Lead Web service, go to the Web Services Administration page within the Oracle CRM On Demand application and generate the WSDL file for the lead object.

Note

The note object stores information about the notes available in the Message Center in the Oracle CRM On Demand application. The notes can be sent from users or can store extra information (as a note) on a parent object. This allows employees who are working on a particular record to add extra information as they see fit. For example, when talking to a contact, an employee might notice that the contact is not happy with a service provided. The employee can record this information in a note so that any other employees who talk to the contact are aware of the contact's dissatisfaction.

The note object has no child components.

Parent Objects

Account, Campaign, Contact, Opportunity, and Service Request

Methods Called

Table 160 details the methods called by the Note service.

Table 160. Methods Called by Note Service

Method	Name as Defined in Service
"Delete" on page 85	NoteDelete
"Insert" on page 89	NoteInsert
"InsertOrUpdate" on page 90	NoteInsertOrUpdate
"QueryPage" on page 91	NoteQueryPage
"Update" on page 101	NoteUpdate

Fields

Table 161 details the required and read-only fields for the note object.

Table 161. Required and Read-Only Fields for the Note Object

Child Component	Field Name	Type
Note (parent)	Subject	Required
	NoteId	Read-only
	OwnerId	Read-only
	OwnerAlias	Read-only
	ParentNoteId	Read-only
	SourceId	Read-only
	SourceName	Read-only
	Audit Fields	Read-only

Table 162 details the status key for the note object.

Table 162. Status Key for the Note Object

Child Component	Field Name
Note (parent)	Audit Fields
	NoteId

Table 163 provides a list of the filterable fields and a list of user key combinations for the note object.

Table 163. Filterable Fields and User Key Fields on the Note Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Note (parent)	All	NoteId

For more information on the fields accessible through the Note Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the note object.

Opportunity

The opportunity object allows employees to identify and record a potential revenue-generating event that has arisen with an account or contact. Opportunities can be generated from marketing campaigns when leads indicate that they are interested in a product or service that has been offered.

Parent Objects

Account, Campaign, Contact, CustomObject1 - CustomObject3

Child Components

Activity, Attachment, Book, Competitor, Contact, Lead, Note, and OpportunityTeam.

NOTE: The Revenue child object for Opportunity is actually called Product.

For information about using attachments with this object, see Appendix A, “Using Attachments With Web Services On Demand.”

Methods Called

Table 164 details the methods called by the Opportunity service.

Table 164. Methods Called by Opportunity Service

Method	Name as Defined in Service
“Delete” on page 85	OpportunityDelete
“DeleteChild” on page 88	OpportunityDeleteChild
“Insert” on page 89	OpportunityInsert
“InsertChild” on page 90	OpportunityInsertChild
“InsertOrUpdate” on page 90	OpportunityInsertOrUpdate
“QueryPage” on page 91	OpportunityQueryPage
“Update” on page 101	OpportunityUpdate
“UpdateChild” on page 102	OpportunityUpdateChild

Fields

Table 165 details the required and read-only fields for the opportunity object.

Table 165. Required and Read-Only Fields for the Opportunity Object

Child Component	Field Name	Type
Opportunity (parent)	AccountId	Required
	CloseDate	Required
	OpportunityName	Required
	SalesStage	Required
	Audit Fields	Read-only
	LastUpdated	Read-only
	OpportunityConcatField	Read-only
Activity	AccountLocation	Read-only
	CreatedDetail	Read-only
	MEEventName	Read-only
	Opportunity	Read-only
	OpportunityIntegrationId	Read-only
	OpportunityExternalId	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	CODInteractionTime	Read-only
	CODWrapUpTime	Read-only
	CODHandleTime	Read-only
	CODIVRTime	Read-only
	CODQueueHoldTime	Read-only
	CODTotalHoldTime	Read-only
	DescriptionShadow	Read-only
	Duration	Read-only
	Audit Fields	Read-only

Table 165. Required and Read-Only Fields for the Opportunity Object

Child Component	Field Name	Type
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	OpportunityId	Read-only
	Id	Read-only
	ModId	Read-only
	Audit Fields	Read-only
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	Audit Fields	Read-only
Competitor	CompetitorId	Required
	CompetitorExternalSystemId	Required
	ReverseRelationshipRole	Required
	RelationshipRole	Required
	StartDate	Required
	OpportunityCompetitorId	Read-only
Contact	Age	Read-only
	OwnerFullName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	LastActivityDate	Read-only
	FirstNameShadow	Read-only
	LastNameShadow	Read-only
OpportunityTeam	OpportunityAccess	Required
	UserId	Required

Table 165. Required and Read-Only Fields for the Opportunity Object

Child Component	Field Name	Type
Partner	OpportunityPartnerId	Read-only
	PartnerExternalSystemId	Required
	ReverseRelationshipRole	Required
	RelationshipRole	Required
	StartDate	Required
Product	ProductRevenueId	Read-only
	ProductCategoryId	Read-only
	ProductCategory	Read-only
	ProductPartNumber	Read-only
	ProductStatus	Read-only
	ProductType	Read-only
	OpportunityId	Read-only
	OpportunityName	Read-only
	OpportunityIntegrationID	Read-only
	OpportunityExternalSystemId	Read-only
	OpportunitySalesStage	Read-only
	OpportunityAccountId	Read-only
	OpportunityAccountName	Read-only
	OpportunityAccountLocation	Read-only
	OpportunityAccountExternalSystemId	Read-only
	OpportunityAccountIntegrationId	Read-only
	ContactFirstName	Read-only
	ContactLastName	Read-only

Table 166 details the status key for the opportunity object.

Table 166. Status Key for the Opportunity Object

Child Component	Field Name
Opportunity (parent)	Audit Fields
	AccountId
	ExternalSystemId
	Id
	IntegrationId
Activity	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
	OpportunityId
Attachment	Audit Fields
	Id
	OpportunityId
Book	Audit Fields
	BookId
	ModId
Competitor	Audit Fields
	OpportunityCompetitorId
	CompetitorId
	CompetitorExternalSystemId
Contact	Audit Fields
	AccountId
	ContactId
	ExternalSystemId
	IntegrationId
	OpportunityId

Table 166. Status Key for the Opportunity Object

Child Component	Field Name
Lead	Audit Fields
	AccountId
	ContactId
	ExternalSystemId
	IntegrationId
	LastUpdated
	LeadId
	OpportunityId
OpportunityNote	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
	OpportunityId
Partner	Audit Fields
	OpportunityPartnerId
	PartnerId
	PartnerExternalSystemId
Product	Audit Fields
	ExternalId
	ProductRevenueId
	IntegrationID

Table 167 details the pick map fields for the opportunity object.

Table 167. Pick Map Fields for the Opportunity Object

Child Component	Pick Map Field	Maps To
Opportunity (parent)	Owner	OwnerId
	AccountExternalSystemId	AccountId
	Territory	TerritoryId
	KeyContactIntegrationId	KeyContactId
	KeyContactExternalSystemId	KeyContactId

Table 167. Pick Map Fields for the Opportunity Object

Child Component	Pick Map Field	Maps To
Activity	AccountIntegrationId	AccountId
	AccountExternalId	AccountId
	Address	AddressId
	AssignedTo	AssignedToId
	CampaignExternalId	CampaignId
	CampaignIntegrationId	CampaignId
	Dealer	DealerId
	DelegatedBy	DelegatedById
	FundRequest	FundRequestId
	OpportunityIntegrationId	OpportunityId
	OpportunityExternalId	OpportunityId
	PrimaryContactIntegrationId	PrimaryContactId
	PrimaryContactExternalId	PrimaryContactId
	LeadIntegrationId	LeadId
	LeadExternalId	LeadId
	OwnerExternalId	OwnerId
	OwnerIntegrationId	OwnerId
	SRIntegrationId	SRId
	SRExternalId	SRId
	CustomObject1Name	CustomObject1Id
	CustomObject2Name	CustomObject2Id
	CustomObject3Name	CustomObject3Id
Book	BookName	BookId
Contact	AccountExternaSystemlId	AccountId
	AccountIntegrationId	AccountId
	ManagerExternalSystemId	ManagerId
	ManagerIntegrationId	ManagerId
	OwnerEmailAddress	OwnerId
	OwnerExternalId	OwnerId
	OwnerIntegrationId	OwnerId
	SourceCampaignName	SourceCampaignId

Table 167. Pick Map Fields for the Opportunity Object

Child Component	Pick Map Field	Maps To
	SourceCampaignExternalId	SourceCampaignId
	TimeZoneName	TimeZoneId
	CustomObject1Name	CustomObject1Id
	CustomObject2Name	CustomObject2Id
	CustomObject3Name	CustomObject3Id
Competitor	PrimaryContactName	ContactId
	PartnerExternalSystemId	PartnerId
	PartnerName	PartnerId
Partner	PrimaryContactName	ContactId
	CompetitorExternalSystemId	CompetitorId
	CompetitorName	CompetitorId
Product	ProductName	ProductId
	ProductExternalSystemId	ProductId
	ProductIntegrationId	ProductId
	ContactExternalSystemId	ContactId
	ContactIntegrationId	ContactId
	Owner	OwnerId
	CustomObject5Name	CustomObject5Id
	CustomObject5ExternalSystemId	CustomObject5Id
	CustomObject6Name	CustomObject6Id
	CustomObject6ExternalSystemId	CustomObject6Id

Table 168 provides a list of the filterable fields for the child components of the opportunity objects, and a list of user key combinations for each child component.

Table 168. Filterable Fields and User Key Fields on the Opportunity Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Opportunity (parent)	All	OpportunityId
		IntegrationId
		ExternalSystemId

Table 168. Filterable Fields and User Key Fields on the Opportunity Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
Activity	Type	Type and Description
	Owner	IntegrationId
	Subject	ExternalSystemId
	DueDate	
	Priority	
	Status	
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Book	None	BookId
		BookName
Competitor	OpportunityCompetitorId	OpportunityCompetitorId
	PrimaryContactId	CompetitorExternalSystemId
	EndDate	CompetitorId
	CompetitorId	
	CompetitorExternalSystemId	
	ReverseRelationshipRole	
	RelationshipRole	
	StartDate	
	ModifiedDate	
Contact	ContactType	AccountName and Private
	ContactFirstName	ContactFirstName and ContactLastName and Private
	JobTitle	IntegrationId
	ContactLastName	ExternalSystemId
	Owner	
	Id	

Table 168. Filterable Fields and User Key Fields on the Opportunity Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Lead	Campaign	IntegrationId
	EstimatedCloseDate	ExternalSystemId
	Rating	
	Source	
	Status	
	LeadOwner	
	PotentialRevenue	
	ProductInterest	
	SalesPerson	
	LeadId	
Note	Subject	Subject and Description
		IntegrationId
		ExternalSystemId
OpportunityTeam	UserFirstName	OpportunityTeamId
	UserLastName	UserId
	ModifiedDate	UserExternalSystemId
	OpportunityAccess	UserEmail
	OpportunityAccessId	UserAlias
	OpportunityTeamId	
	TeamRole	
	UserExternalSystemId	
	UserId	
Partner	OpportunityPartnerId	OpportunityPartnerId
	PrimaryContactId	PartnerExternalSystemId
	EndDate	PartnerId
	PartnerId	
	PartnerExternalSystemId	
	ReverseRelationshipRole	
	RelationshipRole	
	StartDate	
	ModifiedDate	

Table 168. Filterable Fields and User Key Fields on the Opportunity Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Product	OpportunityIntegrationId	ExternalSystemId
	OpportunityExternalSystemId	OpportunityIntegrationId
	OpportunitySalesStage	ProductRevenueId
	OpportunityAccountId	IntegrationId
	OpportunityAccountName	
	OpportunityAccountLocation	
	OpportunityAccountExternalSystemId	
	OpportunityAccountIntegrationId	
	ModifiedDate	
	ContactId	
	ContactExternalSystemId	
	ContactIntegrationId	
	Contract	
	OwnerId	
	Owner	
	SerialNumber	
	Revenue	
	ExpectedRevenue	
	Quantity	
	PurchasePrice	
	PurchaseDate	

Table 168. Filterable Fields and User Key Fields on the Opportunity Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
	StartCloseDate	
	NumberOfPeriods	
	Frequency	
	Probability	
	Forecast	
	AssetValue	
	Premium	
	ShipDate	
	Status	
	Type	
	Warranty	
	ProductRevenueId	
	ExternalSystemId	
	IntegrationId	
	ProductId	
	ProductName	
	ProductExternalSystemID	
	ProductionIntegrationId	
	ProductCategoryId	
	ProductCategory	
	ProductPartNumber	
	ProductStatus	
	ProductType	
	OpportunityId	
	OpportunityName	

Table 169 details the picklists available for the opportunity object.

Table 169. Picklists Available for the Opportunity Object

Child Component	Field Name
Opportunity (parent)	LeadSource
	Priority
	Probability
	ReasonWonLost
	Status
	Type
	Year
	Make
	Model
OpportunityTeam	TeamRole
Product	Frequency
	Probability
	Status
	Type
	Warranty
	Contract

For more information on the fields accessible through the Opportunity Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the opportunity object.

Product

The product object allows you to define and record details about a product or service that your company sells to its customers, including information on product price, category, and so on. The product object does not have any child objects.

Parent Objects

Account, Campaign and Contact

Methods Called

Table 170 details the methods called by the Product service.

Table 170. Methods Called by Product Service

Method	Name as Defined in Service
"Insert" on page 89	ProductInsert
"InsertOrUpdate" on page 90	ProductInsertOrUpdate
"QueryPage" on page 91	ProductQueryPage
"Update" on page 101	ProductUpdate

Fields

All fields on the product object are filterable.

Table 171 details the required and read-only fields for the product object.

Table 171. Required and Read-Only Fields for the Product Object

Child Component	Field Name	Type
Product (parent)	ProductName	Required
	Audit Fields	Read-only

Table 172 details the status key for the product object.

Table 172. Status Key for the Product Object

Child Component	Field Name
Product (parent)	Audit Fields
	Id
	IntegrationId

Table 173 details the pick map field for the product object.

Table 173. Pick Map Field for the Product Object

Pick Map Field	Maps To
ParentCategory	ParentCategoryId

Table 174 details the user keys for the product object.

Table 174. User Keys for the Product Object

Child Component	Field Name
Product (parent)	ProductId
	IntegrationId
	ExternalSystemId

Table 175 details the picklists available for the product object.

Table 175. Picklists Available for the Product Object

Field Name
BodyStyle
Category
Class
CurrencyCode
DoorStyle
Engine
Make
Model
PriceType
ProductType
Revision
Status
SubType
TherapeuticClass
Transmission
Trim

For more information on the fields accessible through the Product Web service, go to the Web Services Administration page within the Oracle CRM On Demand application and generate the WSDL file for the product object.

Related Topic

Product Category

Product Category

The product category object allows you to logically sort products into groups, where each product is in some way related to the other products in the category. The product category object does not have any child objects.

Parent Objects

None

Methods Called

Table 176 details the methods called by the Product category service.

Table 176. Methods Called by Product Category Service

Method	Name as Defined in Service
“Delete” on page 85	ProductCategoryDelete
“Insert” on page 89	ProductCategoryInsert
“InsertOrUpdate” on page 90	ProductCategoryInsertOrUpdate
“QueryPage” on page 91	ProductCategoryQueryPage
“Update” on page 101	ProductCategoryUpdate

Fields

All fields on the product category object are filterable.

Table 177 details the required and read-only fields for the product category object.

Table 177. Required and Read-Only Fields for the Product Category Object

Child Component	Field Name	Type
ProductCategory	CategoryName	Required
	Audit Fields	Read-only
	ModifiedByFullName	Read-only

Table 178 details the status key for the product category object.

Table 178. Status Key for the Product Category Object

Child Component	Field Name
ProductCategory	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
	Name

Table 179 details the pick map field for the product category object.

Table 179. Pick Map Field for the Product Category Object

Pick Map Field	Maps To
ParentCategory	ParentCategoryId

Table 180 details the user keys for the product category object.

Table 180. User Keys for the Product Category Object

Child Component	Field Name
ProductCategory	ProductCategoryId
	IntegrationId
	ExternalSystemId
	Name

For more information on the fields accessible through the Product category Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the product category object.

Related Topic

Automotive Edition Parent Objects (Web Services v1.0)

Service Request

The service request object allows customers to request information or assistance with a problem related to products or services purchased from your company. Service requests can be ranked for severity and prioritized accordingly.

Parent Objects

Account, Contact, and Solution

Child Components

Activity, Attachment, Audit Trail, Book, Note, and Solution.

For information about using attachments with this object, see Appendix A, "Using Attachments With Web Services On Demand."

Methods Called

Table 181 details the methods called by the Service request service.

Table 181. Methods Called by Service Request Service

Method	Name as Defined in Service
"Delete" on page 85	ServiceRequestDelete
"DeleteChild" on page 88	ServiceRequestDeleteChild
"Insert" on page 89	ServiceRequestInsert
"InsertChild" on page 90	ServiceRequestInsertChild
"InsertOrUpdate" on page 90	ServiceRequestInsertOrUpdate
"QueryPage" on page 91	ServiceRequestQueryPage
"Update" on page 101	ServiceRequestUpdate
"UpdateChild" on page 102	ServiceRequestUpdateChild

Fields

Table 182 details the required and read-only fields for the service request object.

Table 182. Required and Read-Only Fields for the Service Request Object

Child Component	Field Name	Type
ServiceRequest (parent)	ContactEmail	Read-only
	ContactFirstName	Read-only
	ContactFullName	Read-only
	ContactLastName	Read-only
	Audit Fields	Read-only
	LastUpdated	Read-only
	ServiceRequestConcatId	Read-only
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	SRId	Read-only
	Id	Read-only
	ModId	Read-only
	Audit Fields	Read-only
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	Audit Fields	Read-only

Table 183 details the status key for the service request object.

Table 183. Status Key for the Service Request Object

Child Component	Field Name
ServiceRequest (parent)	Audit Fields
	AccountId
	ContactId
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
Activity	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
Attachment	Audit Fields
	Id
	SRId
Book	Audit Fields
	BookId
	ModId
ServiceRequestNote	Audit Fields
	ExternalSystemId
	Id
	LastUpdated
	ServiceRequestId
Solution	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated

Table 184 details the pick map fields for the service request object.

Table 184. Pick Map Fields for the Service Request Object

Child Component	Pick Map Field	Maps To
Service Request (parent)	Owner	OwnerId
	AccountExternalSystemId	AccountId
	AssetIntegrationId	AssetId
	AssetExternalSystemId	AssetId
	ProductExternalSystemId	ProductId
Book	BookName	BookId

Table 185 provides a list of the filterable fields for the child components of the service request object, and a list of user key combinations for each child component.

Table 185. Filterable Fields and User Key Fields on the Service Request Object's Child Components

Child Component	Filterable Fields	User Key Field Combinations
Service Request (parent)	All	ServiceRequestId
		IntegrationId
		ExternalSystemId
		SRNumber
Activity	Type	Type and Description
	Owner	IntegrationId
	Subject	ExternalSystemId
	DueDate	
	Priority	
	Status	
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Audit Trail	Date	None
	User	
	FieldModified	

Table 185. Filterable Fields and User Key Fields on the Service Request Object's Child Components

Child Component	Filterable Fields	User Key Field Combinations
Book	None	BookId
		BookName
Service Request Note	Subject	Subject and Description
		ExternalSystemId
Solution	Title	Title
	Published	IntegrationId
	SolutionId	ExternalSystemId
	Status	
	Id	

Table 186 details the picklists available for the service request object.

Table 186. Picklists Available for the Service Request Object

Field Name
Area
Cause
Priority
Source
Status
Type

For more information on the fields accessible through the Service request Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the service request object.

Solution

The solution object stores information on solutions to customer problems or service requests. Solutions can be reused if the same problem is identified with a product or service. This prevents the duplication of work for customer service representatives.

Parent Object

Activity and Service Request

Child Component

Attachment and Service Request.

For information about using attachments with this object, see Appendix A, “Using Attachments With Web Services On Demand.”

Methods Called

Table 187 details the methods called by the Solution service.

Table 187. Methods Called by Solution Service

Method	Name as Defined in Service
“Delete” on page 85	SolutionDelete
“DeleteChild” on page 88	SolutionDeleteChild
“Insert” on page 89	SolutionInsert
“InsertChild” on page 90	SolutionInsertChild
“InsertOrUpdate” on page 90	SolutionInsertOrUpdate
“QueryPage” on page 91	SolutionQueryPage
“Update” on page 101	SolutionUpdate
“UpdateChild” on page 102	SolutionUpdateChild

Fields

Table 188 details the required and read-only fields for the solution object.

Table 188. Required and Read-Only Fields for the Solution Object

Child Component	Field Name	Type
Solution (parent)	Title	Required
	Audit Fields	Read-only
	CreatorId	Read-only
	LastUpdated	Read-only

Table 188. Required and Read-Only Fields for the Solution Object

Child Component	Field Name	Type
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	SolutionId	Read-only
	Id	Read-only
	ModId	Read-only
	Audit Fields	Read-only

Table 189 details the status key for the solution object.

Table 189. Status Key for the Solution Object

Child Component	Field Name
Solution (parent)	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
Attachment	Audit Fields
	Id
	SolutionId
ServiceRequest	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated

Table 190 provides a list of the filterable fields for the child components of the solution objects, and a list of user key combinations for each child component.

Table 190. Filterable Fields and User Key Fields on the Solution Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Solution (parent)	All	SolutionId
		IntegrationId
		ExternalSystemId
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Service Request	Subject	SRNumber
	Area	IntegrationId
	Owner	ExternalSystemId
	Priority	
	Type	
	Cause	
	Source	
	Status	

Table 191 details the picklists available for the solution object.

Table 191. Picklists Available for the Solution Object

Field Name
Area
Cause
Priority
Source
Status
Type

For more information on the fields accessible through the Solution Web service, go to the Web Services Administration page within the Oracle CRM On Demand application and generate the WSDL file for the solution object.

Related Topic

Service Request

Territory

The territory object allows you to store information about the sales territory that is assigned to a user. This information includes the territory name, a description, the currency code, and the sales quota for the territory. The territory object does not have any associated child objects or parent objects.

Parent Objects

None

Methods Called

Table 192 details the methods called by the Territory service.

Table 192. Methods Called by Territory Service

Method	Name as Defined in Service
“Delete” on page 85	TerritoryDelete
“Insert” on page 89	TerritoryInsert
“InsertOrUpdate” on page 90	TerritoryInsertOrUpdate
“QueryPage” on page 91	TerritoryQueryPage
“Update” on page 101	TerritoryUpdate

Fields

All fields on the territory object are filterable. The TerritoryName, TerritoryExternalSystemId, and TerritoryIntegrationId fields are user keys for the territory object.

Table 193 details the required and read-only fields for the territory object.

Table 193. Required and Read-Only Fields for the Territory Object

Child Component	Field Name	Type
Territory (parent)	TerritoryName	Required
	Territory	Read-only
	Audit Fields	Read-only

Table 194 details the status key for the territory object.

Table 194. Status Key for the Territory Object

Child Component	Field Name
Territory (parent)	Audit Fields
	TerritoryExternalSystemId
	TerritoryIntegrationId
	TerritoryId

Table 195 details the pick map field for the territory object.

Table 195. Pick Map Field for the Territory Object

Pick Map Field	Maps To
ParentTerritoryIntegrationId	ParentTerritoryId
ParentTerritoryExternalSystemId	ParentTerritoryId
ParentTerritoryId	ParentTerritoryId

Table 196 details the picklists available for the territory object.

Table 196. Picklists Available for the Territory Object

Field Name
ParentTerritoryExternalSystemId
ParentTerritoryIntegrationId

For more information on the fields accessible through the Territory Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the territory object.

User

The user object allows you to define and record details of all users in Oracle CRM On Demand, for example, name, position, contact details, manager, and so on. It is different from the Current User object in that it is not restricted only to the currently logged in user. It enables queries to be run on all users, and enables an administrator to insert and update a user's profile. The user object does not have any child components.

Usage

The UserLoginId and UserSignInId fields must be used as follows:

- **UserLoginId**. Used for creating user records through the User Web service.
- **UserSignInId**. Used as the user name for logging in and authenticating using Web services. Also, used for queries, as using UserLoginId is not allowed for queries.

Parent Object

User Group

Child Object

Address, DelegatedUser

Methods Called

Table 197 details the methods called by the User service.

Table 197. Methods Called by User Service

Method	Name as Defined in Service
“DeleteChild” on page 88	UserDeleteChild
“Insert” on page 89	UserInsert
“InsertChild” on page 90	UserInsertChild
“InsertOrUpdate” on page 90	UserInsertOrUpdate
“QueryPage” on page 91	UserQueryPage
“Update” on page 101	UserUpdate
“UpdateChild” on page 102	UserUpdateChild

Fields

All fields on the user object are filterable.

Table 198 details the required and read-only fields for the user object.

Table 198. Required and Read-Only Fields for the User Object

Child Component	Field Name	Type
User	FirstName	Required
	LastName	Required
	UserLoginId	Required
	UserSignInId	Required
	Alias	Required
	EmailAddr	Required
	Role	Required
	Status	Required
	Audit Fields	Read-only
	LastSignInDateTime	Read-only
	ManagerFullName	Read-only

Table 199 details the status key for the user object.

Table 199. Status Key for the User Object

Child Component	Field Name
User	ModifiedById
	ModifiedDate
	EMailAddr
	UserId
	IntegrationId

Table 200 details the pick map field for the user object.

Table 200. Pick Map Field for the User Object

Pick Map Field	Maps To
Role	RoleId

Table 201 provides a list of user key combinations for the user object.

Table 201. User Key Fields on the User Object

Child Components	User Key Field Combinations
User	UserId
	ExternalSystemId
	IntegrationId
	EmailAddr
Address	IntegrationId
DelegatedUser	ExternalSystemId

For more information on the fields accessible through the User Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the user object.

Related Topic

Current User

User Group

The User Group object allows you to create groups to which users can be added. Users can only be a member of one group, and groups can contain many users.

Parent Objects

None

Child Component

User

Methods Called

Table 202 details the methods called by the User group service.

Table 202. Methods Called by User Group Service

Method	Name as Defined in Service
"Delete" on page 85	UserGroupDelete
"DeleteChild" on page 88	UserGroupDeleteChild
"Insert" on page 89	UserGroupInsert
"InsertChild" on page 90	UserGroupInsertChild
"InsertOrUpdate" on page 90	UserGroupInsertOrUpdate
"QueryPage" on page 91	UserGroupQueryPage
"Update" on page 101	UserGroupUpdate
"UpdateChild" on page 102	UserGroupUpdateChild

Fields

Table 203 details the required and read-only fields for the user group object.

Table 203. Required and Read-Only Fields for the User Group Object

Child Component	Field Name	Type
User Group (parent)	Name	Required
	UserGroupId	Read-only
	Audit Fields	Read-only
User	UserGroupUserId	Read-only
	UserId	Read-only
	Alias	Read-only
	Email	Read-only
	Role	Read-only
	UserFirstName	Read-only
	UserLastName	Read-only
	Audit Fields	Read-only

Table 204 details the status key for the user group object.

Table 204. Status Key for the User Group Object

Child Component	Field Name
UserGroup (parent)	Audit Fields
	UserGroupId
	UserGroupIntegrationId
	UserGroupExternalSystemId
User	Audit Fields
	Members_UserId
	UserExternalSystemId
	UserIntegrationId

Table 205 details the pick map field for the user group object.

Table 205. Pick Map Field for the User Group Object

Child Component	Pick Map Field	Maps To
User	UserIntegrationId	UserId
	UserExternalSystemId	UserId

Table 206 provides a list of the filterable fields for the child components of the user group object, and a list of user key combinations for each child component.

Table 206. Filterable Fields and User Key Fields on the User Group Object's Child Components

Child Component	Filterable Fields	User Key Field Combinations
User Group (parent)	All	Name
		UserGroupIntegrationId
		UserGroupExternalSystemId

Table 206. Filterable Fields and User Key Fields on the User Group Object's Child Components

Child Component	Filterable Fields	User Key Field Combinations
User	UserGroupId	UserIntegrationId
	UserId	UserExternalSystemId
	UserIntegrationId	
	UserExternalSystemId	
	Alias	
	Email	
	Role	
	UserFirstName	
	UserLastName	

For more information on the fields accessible through the User group Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the user group object.

Life Sciences Edition Parent Objects (Web Services v1.0)

This topic includes the parent objects available with Oracle CRM On Demand Life Sciences Edition.

To download WSDL files for these objects, you must be given access to the object. If you do not have access to the object, it is not available to download from the Web Services Administration page or available to use the Web service calls. For assistance in gaining access to the object, contact your Oracle CRM On Demand service provider.

The following objects are detailed in this topic:

- "MedEd" on page 368

MedEd

The MedEd object allows you to plan and track medical education events. A medical education event can be as simple as a lunch-and-learn session in a physician's office or as complex as a seminar series or national sales meeting.

Child Component

Invitee

Methods Called

Table 207 details the methods called by the MedEd service.

Table 207. Methods Called by MedEd Service

Method	Name as Defined in Service
“Delete” on page 85	MedEdDelete
“DeleteChild” on page 88	MedEdDeleteChild
“Insert” on page 89	MedEdInsert
“InsertChild” on page 90	MedEdInsertChild
“InsertOrUpdate” on page 90	MedEdInsertOrUpdate
“QueryPage” on page 91	MedEdQueryPage
“Update” on page 101	MedEdUpdate
“UpdateChild” on page 102	MedEdUpdateChild

Fields

Table 208 details the read-only fields for the MedEd object and its child component.

Table 208. Read-Only Fields on the MedEd Object

Child Component	Field Name	Type
MedEd (parent)	EndDate	Required
	Name	Required
	Objective	Required
	StartDate	Required
	Audit Fields	Read-only
Invitee	Status	Required
	InviteeId	Read-only
	Audit Fields	Read-only

Table 209 details the status key for the MedEd object.

Table 209. Status Key for the MedEd Object

Child Component	Field Name
MedEd (parent)	Audit Fields
	ExternalId
	MedEdId
Invitee	Audit Fields
	InviteeExternalSystemId
	InviteeId

Table 210 details the pickmap fields for the MedEd object and its child objects.

Table 210. Pick Map Fields for the MedEd Object

Child Component	Pick Map Field	Maps To
MedEd (parent)	ProductExternalId	ProductId
	ProductIntegrationId	ProductId

Table 211 provides a list of the filterable fields and user key combinations for the child components of the MedEd object.

Table 211. Filterable Fields and User Key Fields on the MedEd Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
MedEd (parent)	ExternalSystemId	MedEdId
	ProductIntegrationId	ExternalSystemId
	ProductId	
	ProductId	
	ProductExternalId	
	PrimaryOwnerId	
Invitee	ContactId	InviteeId
	InviteeExternalSystemId	InviteeExternalSystemId
	Status	
	Type	
	ModifiedDate	

Table 212 details the picklists available for the MedEd object.

Table 212. Picklists Available for the MedEd Object

Child Component	Field Name
MedEd (parent)	EventStatusCode
	EventTypeCode
Invitee	InviteeStatus

For more information on the fields accessible through the MedEd Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the MedEd object.

Related Topic

Invitee

Financial Services Edition Parent Objects (Web Services v1.0)

This topic includes the parent objects available with Oracle CRM On Demand Financial Services Edition.

NOTE: To download WSDL files for these objects, you must be given access to the object. If you do not have access to the object, it is not available to download from the Web Services Administration page or available to use the Web service calls. For assistance in gaining access to the objects, contact your Oracle CRM On Demand service provider.

The following objects are detailed in this topic:

- “Household” on page 371
- “Portfolio” on page 376

Household

The household object allows you to define and record financial details about a group of contacts that live in the same household, for example, parents, brothers, sisters, spouses, and so on. These details include the assets of the household, the liabilities of the household, the net income of the household, and so on.

Child Component

Book, HouseholdTeam

Methods Called

Table 213 details the methods called by the Household service.

Table 213. Methods Called by Household Service

Method	Name as Defined in Service
"Delete" on page 85	HouseholdDelete
"DeleteChild" on page 88	HouseholdDeleteChild
"Insert" on page 89	HouseholdInsert
"InsertChild" on page 90	HouseholdInsertChild
"InsertOrUpdate" on page 90	HouseholdInsertOrUpdate
"QueryPage" on page 91	HouseholdQueryPage
"Update" on page 101	HouseholdUpdate
"UpdateChild" on page 102	HouseholdInsertChild

Fields

Table 214 details the required and read-only fields for the household object.

Table 214. Required and Read-Only Fields for the Household Object

Child Component	Field Name	Type
Household (parent)	HouseholdName	Required
	IntegrationID	Required
	ExternalSystemID	Required
	HouseholdId	Read-only
	PrimaryContactId	Read-only
	PrimaryContactExternalId	Read-only
	PrimaryContactIntegrationId	Read-only
	PrimaryContactFirstName	Read-only
	PrimaryContactLastName	Read-only
	Timezone	Read-only
	HouseholdCurrency	Read-only
	LastActivity	Read-only
	HeadDOB	Read-only
	TotalIncome	Read-only
	TotalAssets	Read-only
	TotalExpenses	Read-only
	TotalLiabilities	Read-only
	TotalNetWorth	Read-only
	RiskProfile	Read-only
	ExperienceLevel	Read-only
	InvestmentHorizon	Read-only
	CurrentInvestmentMix	Read-only
	Objective	Read-only
	PrimaryGoal	Read-only
	Audit Fields	Read-only

Table 214. Required and Read-Only Fields for the Household Object

Child Component	Field Name	Type
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	Audit Fields	Read-only
Contact	ContactId	Required
	ContactExternalId	Required
	ContactIntegrationId	Required
	ContactFirstName	Read-only
	ContactLastName	Read-only
	ContactId	Read-only
	ContactMrMrs	Read-only
	Audit Fields	Read-only
HouseholdTeam	HouseholdAccess	Required
	UserId	Read-only
	UserAlias	Read-only
	UserEmail	Read-only

Table 215 details the status key for the household object.

Table 215. Status Key for the Household Object

Child Component	Field Name
Household (parent)	Audit Fields
	ExternalSystemId
	HouseholdId
	IntegrationID
Book	Audit Fields
	BookId
	ModId
Contact	Audit Fields
	ContactId

Table 215. Status Key for the Household Object

Child Component	Field Name
HouseholdTeam	Audit Fields
	UserExternalSystemId
	HouseholdTeamId
	UserAlias
	UserEmail

Table 216 details the pick map fields for the household object.

Table 216. Pick Map Field for the Household Object

Child Component	Pick Map Field	Maps To
Contact	ContactExternalId	ContactId
	ContactIntegrationId	
Book	BookName	BookId
HouseholdTeam	UserEmail	UserId
	UserAlias	UserId
	UserExternalSystemId	UserId
	LastName	UserId
	FirstName	UserId

Table 217 provides a list of the filterable fields for the child components of the household object, and a list of user key combinations for each child component.

Table 217. Filterable Fields and User Key Fields on the Household Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Household (parent)	All	HouseholdId
		IntegrationID
		ExternalSystemID
Book	None	BookId
		BookName

Table 217. Filterable Fields and User Key Fields on the Household Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Contact	ContactID	ContactID
	ContactExternalId	ContactExternalId
	ContactIntegrationId	ContactIntegrationId
	ModifiedDate	
	RelationshipRole	
HouseholdTeam	UserId	HouseholdTeamId
	UserExternalSystemId	UserExternalSystemId
	LastName	UserAlias
	FirstName	UserEmail
	TeamRole	
	HouseholdAccess	
	HouseholdTeamId	

Table 218 details the picklists available for the household object.

Table 218. Picklists Available for the Household Object

Child Component	Field Name
Household (parent)	Segment
	Type
Contact	RelationshipRole
HouseholdTeam	TeamRole
	HouseholdAccess

For more information on the fields accessible through the Household Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the household object.

Portfolio

The portfolio object allows you to define and record details about the collection of financial services that you can provide to an account. Financial services include loans, credit cards, insurance, general banking, and so on.

Child Component

Book, Contact, PortfolioTeam

Methods Called

Table 219 details the methods called by the Portfolio service.

Table 219. Methods Called by Portfolio Service

Method	Name as Defined in Service
"Delete" on page 85	PortfolioDelete
"DeleteChild" on page 88	PortfolioDeleteChild
"Insert" on page 89	PortfolioInsert
"InsertChild" on page 90	PortfolioInsertChild
"InsertOrUpdate" on page 90	PortfolioInsertOrUpdate
"QueryPage" on page 91	PortfolioQueryPage
"Update" on page 101	PortfolioUpdate
"UpdateChild" on page 102	PortfolioUpdateChild

Fields

Table 220 details the read-only fields for the portfolio object and its child component.

Table 220. Read-Only Fields on the Portfolio Object

Child Component	Field Name	Type
Portfolio (parent)	PortfolioId	Read-only
	Owner	Read-only
	OwnerId	Read-only
	PrimaryContact	Read-only
	Audit Fields	Read-only
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	Audit Fields	Read-only

Table 220. Read-Only Fields on the Portfolio Object

Child Component	Field Name	Type
Contact	ContactId	Read-only
	ContactFirstName	Read-only
	ContactLastName	Read-only
	ContactHomePhone	Read-only
	ContactEmail	Read-only
	Audit Fields	Read-only
PortfolioTeam	PortfolioAccess	Required
	UserId	Read-only
	UserAlias	Read-only
	UserEmail	Read-only

Table 221 details the status key for the portfolio object.

Table 221. Status Key for the Portfolio Object

Child Component	Field Name
Portfolio (parent)	Audit Fields
	ExternalSystemId
	PortfolioId
	IntegrationId
Book	Audit Fields
	BookId
	ModId
Contact	Audit Fields
	ContactId
	Id
PortfolioTeam	Audit Fields
	UserId
	UserAlias
	UserEmail
	UserExternalSystemId

Table 222 details the pickmap fields for the portfolio object and its child objects.

Table 222. Pick Map Fields for the Portfolio Object

Child Component	Pick Map Field	Maps To
Portfolio (parent)	InstitutionExternalId	InstitutionId
	InstitutionIntegrationId	InstitutionId
	InstitutionName	InstitutionId
	InstitutionLocation	InstitutionId
	Product	ProductId
	ProductExternalId	ProductId
	ProductIntegrationId	ProductId
Book	BookName	BookId
Contact	ContactExternalId	ContactId
	ContactIntegrationId	ContactId
PortfolioTeam	UserEmail	UserId
	UserAlias	UserId
	UserExternalSystemId	UserId
	LastName	UserId
	FirstName	UserId
	FullName	UserId

Table 223 provides a list of the filterable fields and user key combinations for the child components of the portfolio object.

Table 223. Filterable Fields and User Key Fields on the Portfolio Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Portfolio (parent)	All	PortfolioId
		IntegrationId
		ExternalSystemId
Book	None	BookId
		BookName

Table 223. Filterable Fields and User Key Fields on the Portfolio Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Contact	ContactId	ContactId
	ContactExternalId	ContactExternalId
	ContactIntegrationId	ContactIntegrationId
	ContactFirstName	
	ContactLastName	
	ContactHomePhone	
	ContactEmail	
	PrimaryInsured	
	NamedInsured	
	PolicyOwner	
	Relationship	
PortfolioTeam	UserId	UserId
	UserExternalSystemId	UserExternalSystemId
	LastName	UserAlias
	FirstName	UserEmail
	TeamRole	
	PortfolioAccess	

Table 224 details the picklists available for the portfolio object.

Table 224. Picklists Available for the Portfolio Object

Child Component	Field Name
Portfolio (parent)	AccountType
	Status
	TermUnit
Contact	Relationship
PortfolioTeam	TeamRole
	PortfolioAccess

For more information on the fields accessible through the Portfolio Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the portfolio object.

Automotive Edition Parent Objects (Web Services v1.0)

This topic includes the parent objects available with Oracle CRM On Demand Automotive Edition.

NOTE: To download WSDL files for these objects, you must be given access to the object. If you do not have access to the object, it is not available to download from the Web Services Administration page or available to use the Web service calls. For assistance in gaining access to the objects, contact your Oracle CRM On Demand service provider.

The following objects are detailed in this topic:

- “Dealer” on page 381
- “Vehicle” on page 384

Dealer

The dealer object stores information about dealerships in the automotive industry, for example, the name of the dealership, the identity of the parent dealership, the site on which the dealership is based, and so on. The dealer object does not have any parent objects.

Child Objects

Attachment and Book.

For information about using attachments with this object, see Appendix A, “Using Attachments With Web Services On Demand.”

Methods Called

Table 225 details the methods called by the Dealer service.

Table 225. Methods Called by Dealer Service

Method	Name as Defined in Service
“Delete” on page 85	DealerDelete
“DeleteChild” on page 88	DealerDeleteChild
“Insert” on page 89	DealerInsert
“InsertChild” on page 90	DealerInsertChild
“InsertOrUpdate” on page 90	DealerInsertOrUpdate
“QueryPage” on page 91	DealerQueryPage
“Update” on page 101	DealerUpdate
“UpdateChild” on page 102	DealerUpdateChild

Fields

All fields on the dealer object are filterable.

Table 226 details the required and read-only fields for the dealer object.

Table 226. Required and Read-Only Fields for the Dealer Object

Child Component	Field Name	Type
Dealer	DealerId	Required
	DealerIntegrationId	Required
	DealerExternalSystemID	Required
	DealerId	Read-only
	DealerType	Read-only
	Audit Fields	Read-only
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	DealerId	Read-only
	Id	Read-only
	ModId	Read-only
	Audit Fields	Read-only
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	Audit Fields	Read-only

Table 227 details the status key for the dealer object.

Table 227. Status Key for the Dealer Object

Child Component	Field Name
Dealer (parent)	Audit Fields
	DealerId
	DealerIntegrationID
	DealerExternalSystemId
Attachment	Audit Fields
	Id
	DealerId
Book	Audit Fields
	BookId
	ModId

Table 228 details the pick map fields for the dealer object.

Table 228. Pick Map Fields for the Dealer Object

Child Components	Pick Map Field	Maps To
Dealer (parent)	Owner	OwnerId
	ParentDealerExternalSystemId	ParentDealerId
	ParentDealerIntegrationId	ParentDealerId
	ParentDealerName	ParentDealerId
	ParentDealerSite	ParentDealerId
Book	BookName	BookId

Table 229 provides a list of the filterable fields for the child components of the dealer object, and a list of user key combinations for each child component.

Table 229. Filterable Fields and User Key Fields on the Dealer Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
Dealer (parent)	All	DealerId
		DealerIntegrationID
		DealerExternalSystemID

Table 229. Filterable Fields and User Key Fields on the Dealer Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Book	None	BookId
		BookName

Table 230 details the picklists available for the dealer object.

Table 230. Picklists Available for the Dealer Object

Field Name
ParentDealerName
ParentDealerSite

For more information on the fields accessible through the Dealer Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the dealer object.

Vehicle

The vehicle object allows you to create and store information about a vehicle, for example, a car, a truck, a van, and so on, that your company would like to sell to a contact or account. This information includes the vehicle's current mileage, the invoice price, the dealership, the make, and so on.

Parent Objects

None

Child Component

Contact

Methods Called

Table 231 details the methods called by the Vehicle service.

Table 231. Methods Called by Vehicle Service

Method	Name as Defined in Service
"Delete" on page 85	VehicleDelete
"DeleteChild" on page 88	VehicleDeleteChild
"Insert" on page 89	VehicleInsert
"InsertChild" on page 90	VehicleInsertChild
"InsertOrUpdate" on page 90	VehicleInsertOrUpdate
"QueryPage" on page 91	VehicleQueryPage
"Update" on page 101	VehicleUpdate
"UpdateChild" on page 102	VehicleUpdateChild

Fields

Table 232 details the required and read-only fields for the vehicle object.

Table 232. Required and Read-Only Fields for the Vehicle Object

Child Component	Field Name	Type
Vehicle (parent)	VehicleId	Read-only
	Contact	Read-only
	ProductType	Read-only
	SellingDealer	Read-only
	ServicingDealer	Read-only
	Audit Fields	Read-only
Contact	ContactId	Required
	ContactExternalSystemId	Required
	ContactIntegrationId	Required
	Audit Fields	Read-only

Table 233 details the status key for the vehicle object.

Table 233. Status Key for the Vehicle Object

Child Component	Field Name
Vehicle (parent)	Audit Fields
	ExternalSystemId
	IntegrationId
	VehicleId
Contact	Audit Fields
	ContactId

Table 234 details the pick map fields for the vehicle object.

Table 234. Pick Map Fields for the Vehicle Object

Child Component	Pick Map Field	Maps To
Vehicle (parent)	AccountName	AccountId
	AccountSite	AccountId
	AccountIntegrationId	AccountId
	AccountExternalID	AccountId
	SellingDealerExternalId	SellingDealerId
	SellingDealerIntegrationId	SellingDealerId
	ServicingDealerExternalId	ServicingDealerId
	ServicingDealerIntegrationId	ServicingDealerId
Contact	ContactExternalSystemId	ContactId
	ContactIntegrationId	ContactId

Table 235 provides a list of the filterable fields for the child components of the vehicle object, and a list of user key combinations for each child component.

Table 235. Filterable Fields and User Key Fields on the Vehicle Object's Child Components

Child Component	Filterable Fields	User Key Field Combinations
Vehicle (parent)	All	VehicleId
		ExternalSystemId
		IntegrationId

Table 235. Filterable Fields and User Key Fields on the Vehicle Object's Child Components

Child Component	Filterable Fields	User Key Field Combinations
Contact	ContactId	ContactId
	ContactExternalSystemId	ContactExternalSystemId
	ContactIntegrationId	ContactIntegrationId
	ContactFirstName	

Table 236 details the picklists available for the vehicle object.

Table 236. Picklists Available for the Vehicle Object

Field Name
Body
Door
Engine
ExteriorColor
InteriorColor
Location
Make
Model
VehicleOwnedBy
Status
Transmission
Trim
UsedNew
WarrantyType
Year

For more information on the fields accessible through the Vehicle Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the vehicle object.

Child Objects (Web Services v1.0)

The following is a list of child objects that are used in Oracle On Demand Web Services. These are objects that are child objects only and are not themselves parent objects:

Address

The address object stores information on the different addresses that are associated with accounts and contacts. It is used to store billing and shipping addresses for accounts. It is also used to store the personal addresses for contacts.

Parent Objects

Account, Contact

Attachment

The attachment object stores information about a file or URL that is attached to a record in Oracle CRM On Demand.

Parent Objects

Account, Activity, Campaign, Contact, CustomObject1 - CustomObject3, Lead, Lead, Opportunity, Service Request, and Solution.

Audit Trail

The audit trail object stores information about how a service request object is modified from the moment that it is created until a solution for the service request has been found. The audit trail object stores information, such as the created and modified dates for the service request, and also the users who created and updated the service request.

Parent Object

Service Request

BookUser

The BookUser object stores information about a book user.

For more information on the BookUser fields accessible, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the book object.

Parent Object

Book

Competitor

The competitor object stores the information on competitors for your accounts.

Fields

Table 237 details the picklists available for the competitor object.

Table 237. Picklists Available for the Competitor Object

Field Name
RelationshipRole
ReverseRelationshipRole

For more information on the competitor fields accessible, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the account object.

Parent Objects

Account, Opportunity

DelegatedUser

The DelegatedUser object stores information about a user that can impersonate another user. The delegated user is added so that he or she can access a user's data, for example, when a person is on vacation or leaves the company.

For more information on the DelegatedUser fields accessible, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the user object.

Parent Object

User

HouseholdTeam

The HouseholdTeam object stores the information on a team that shares household records.

For more information on the household team fields accessible, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the household object.

Parent Object

Household

Interests

The Interests object stores information about things in which a contact is interested, such as products, services, or hobbies.

Parent Object

Contact

Invitee

The Invitee object stores information about invitees to medical education events, including feedback about the invitation.

Parent Object

MedEd

Login History

The login history object stores information about the currently logged in user, such as the amount of times that the user has logged in, and the dates and times at which the current user logged in.

Parent Object

Current User

Multiple Contact Roles

The multiple contact roles object stores information on the different roles that a contact can hold within an account. It stores information on the different types of jobs that one contact can hold within your organization. For example, the customer relations manager can also have a role within the sales team to provide valuable feedback to the sales representatives.

Parent Object

Account

OpportunityTeam

The OpportunityTeam object stores information about a team that shares opportunity records.

For more information on the opportunity team fields accessible, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the opportunity object.

Parent Object

Opportunity

PortfolioTeam

The PortfolioTeam object stores information about a team that shares portfolio records

For more information on the portfolio team fields accessible, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the portfolio object.

Parent Objects

Account, Product Category

ProductsDetailed

The ProductsDetailed object stores the information on product details for an activity. This is used, for example, to record information about products discussed on sales calls to customers.

For more information on the product detail fields accessible, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the activity object.

Parent Objects

Activity

Quota

The quota object stores information about the sales targets of and sales made by the current user.

Parent Object

Current User

Recipient

The recipient object stores information about a recipient associated with a campaign.

Parent Object

Campaign

Related Account

The related account object stores information on an account that has a relationship with the parent account in question. The details of the related account child object are inherited from a particular account parent object.

Parent Object

Account

Related Contact

The related contact object stores information about a contact that has a relationship with the parent contact in question. The details of the related contact child object are inherited from a particular contact parent object.

Parent Object

Contact

Revenue

The revenue object stores monetary information about accounts, contacts, and their associated opportunities. This includes information on the revenue available, expected revenue, and also information about the products associated with the accounts, contacts, opportunities, and so on.

NOTE: The Revenue child object of Opportunity is actually called Product.

Parent Objects

Account, Contact and Opportunity

SampleDropped

The SampleDropped object stores the information on samples for an activity. This is used, for example, to record information about samples left with the customer on sales calls to customers.

For more information on the SampleDropped fields accessible, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the activity object.

Parent Object

Activity

SubBook

The Subbook object stores information about a subbook.

For more information on the SubBook fields accessible, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the book object.

Parent Object

Book

Team

The team object stores information on the team that is assigned to a particular account or contact. In this way, a team of employees can be dedicated to an account or contact, ensuring that the activities, service requests, leads, and opportunities surrounding that account or contact are always kept up-to-date and are attended to regularly.

Parent Objects

Account, Contact, and CustomObject1 - CustomObject3.

Core Parent Objects (Web Services v2.0)

This topic include the standard objects of Oracle CRM On Demand. Objects that are only available with industry-specific editions of the application are covered in the following topics:

- "Partner Relationship Management Edition Objects (Web Services v2.0)" on page 436
- "Life Sciences Edition Parent Objects (Web Services v2.0)" on page 470
- "Financial Services Edition Parent Objects (Web Services v2.0)" on page 511
- "Automotive Edition Parent Objects (Web Services v2.0)" on page 538

The following Oracle CRM On Demand objects are detailed in this topic:

Objects accessible through the Web Services v2.0 API can reference other objects through a number of reference fields, which are foreign key fields for those other objects. You can determine the objects that are referenced by examining the WSDL file for the referencing object.

NOTE: Information about status keys is not given in this topic because a standard set of status key fields is returned for all parent and child objects accessible through the Web Services v2.0 API. For more information about status keys, see "Oracle CRM On Demand Status Keys" on page 253.

Account

The account object stores information about the companies that you do business with and is also used to track partners and competitors. The methods called on the account object require a list (array) of account objects as an input argument. This list of accounts identifies the records on which the operation is to be carried out.

Adding and Updating Account Addresses

The account object has three types of address:

1. Billing address
2. Shipping address
3. Simple address

The billing and shipping address correspond to fields in the account object. The address child object can correspond to a simple address, but may also correspond to a billing address or a shipping address.

You can use either AccountInsert, AccountUpdate or AccountExecute calls to insert or update billing and shipping addresses for an account. With AccountInsert or AccountUpdate calls, the account parent node with billing and shipping address fields must be present and there must not be any child node. If an address child is present, only a simple address can be added, and not billing and shipping addresses. If the address child or any other child node is not present, only billing and shipping addresses can be added, and not simple addresses.

With the AccountExecute call, both the account parent node and address child node can be present along with billing and shipping address fields at the parent level.

To distinguish whether the address in an address child node is a billing or shipping address:

- If the PrimaryBillToAddressId and AddressId fields have the same value, then the address is a billing address.
- If the PrimaryShipToAddressId and AddressId have the same values, the address is a shipping address.

NOTE: A *PrimaryAddressLine1* field is used to dynamically map the primary address field from an external application to the primary address field in Oracle CRM On Demand. For more information, see "Mapping Primary Address Fields Using Web Services" on page 35.

Child Objects

Account Contact, Account Opportunity, Account Team, Activity, Address, Asset, Attachment, Business Plan, Claim, Competitor, Contact, Contact Role, Custom Object 1 through 3, Deal Registration, Financial Account, Financial Account Holder, Financial Plan, Lead, Message Response, Note, Objective, Partner, Policy, Policy Holder, Revenue, Service Request, Special Pricing Request.

Methods Called

Table 238 details the methods called by the Account service.

Table 238. Methods Called by Account Service

Method	Name as Defined in Service
"Delete" on page 105	AccountDelete
"Execute" on page 108	AccountExecute

Table 238. Methods Called by Account Service

Method	Name as Defined in Service
“Insert” on page 112	AccountInsert
“QueryPage” on page 115	AccountQueryPage
“Update” on page 136	AccountUpdate

Fields

Table 239 details the required and read-only fields for the account object.

Table 239. Required and Read-Only Fields for the Account Object

Field Name	Type
AccountName	Required
AccountConcatField	Read-only
Audit Fields	Read-only

Table 240 details the pick map fields for the account object.

Table 240. Pick Map Fields for the Account Object

Pick Map Field	Maps To
Owner	OwnerId
ParentAccount, ParentAccountLocation	ParentAccountId
ParentAccountIntegrationId	ParentAccountId
ParentAccountExternalSystemId	ParentAccountId

Table 241 provides a list of the user key combinations for the account object.

Table 241. User Key Fields on the Account Object

User Key Field Combinations
Id
ExternalSystemId
AccountName and Location

Table 242 details the picklists available for the account object.

Table 242. Picklists Available for the Account Object

Field Name
AccountType
Priority
Region
CallFrequency
InfluenceType
Route
Status
MarketPotential
MarketSegment

Activity

The activity object stores information on an activity that a user must carry out, for example, a call-back activity for an account. When an activity is created, the user must set the Activity field explicitly to Task or Appointment.

Usage

Oracle On Demand Web Services uses activities to organize, track, and resolve a variety of tasks, from finding and pursuing opportunities to closing service requests. If a task requires multiple steps that one or more people can carry out, activities greatly simplify the job. Activities can help to:

- Define and assign the task
- Provide information to complete the task
- Track the progress of the task
- Track costs and bill for the task

NOTE: The Owner field is not accessible for the activity object in Web Services v2.0.

Support for Recurring Events

The Activity service allows access to the iCRMId field. This field is used to support the synchronization of recurring appointments between Oracle CRM On Demand Desktop and Oracle CRM On Demand.

Microsoft Outlook supports a variety of recurring appointments (for example, meetings) where an appointment can occur daily, weekly, monthly, and so on. In addition, Outlook also supports exceptions within a recurring series of appointments, for example, a meeting that occurs at the same time each day in the week apart from one day where the meeting is at a different time. However, Oracle CRM On Demand does not support recurring activities, so each occurrence of a recurring appointment is created as an individual activity in Oracle CRM On Demand.

The iCRMId field contains a representation of the recurrence parameters and is required to re-assemble the recurring appointment and its exceptions within Outlook if there are ny changes to the individual activities in Oracle CRM On Demand. The field is 450 characters in length, is indexed, and contains:

- The unique Id of the recurring appointment generated by Outlook.
- The recurrence rule (RRULE), for example:

  ```
  FREQ=MONTHLY;UNTIL=20110207T144325;INTERVAL=2;BYDAY=MO,TU,WE,TH,FR;BYSETPOS=-1
  ```

- VTIMEZONE, the timezone specified for each Outlook appointment, for example:

  ```
  BEGIN:VTIMEZONE
  TZID:(UTC+02:00)
  BEGIN:STANDARD
  DTSTART:16011028T040000
  TZOFFSETFROM:+0300
  TZOFFSETTO:+0200
  END:STANDARD
  BEGIN:DAYLIGHT
  DTSTART:16010325T030000
  TZOFFSETFROM:+0200
  TZOFFSETTO:+0300
  END:DAYLIGHT
  END:VTIMEZONE
  ```

- The offset of the occurrence from the start of the series (int).
- The hash value of the original series fields (int). This includes the hash value of location, appt startdate, appt enddate, subject, and description.

For more information about the synchronization of recurring appointments between Oracle CRM On Demand Desktop and Oracle CRM On Demand, see *Oracle CRM On Demand Desktop Administration Guide*.

Child Objects

Activity Assessment, Activity Assessment Value, Attachment, Contact, Custom Object 1 through 3, Message Response, Products Detailed, Promotional Items Dropped, Sample Dropped, Sample Transaction, Signature, User.

Methods Called

Table 243 details the methods called by the Activity service.

Table 243. Methods Called by Activity Service

Method	Name as Defined in Service
"Delete" on page 105	ActivityDelete
"Execute" on page 108	ActivityExecute
"Insert" on page 112	ActivityInsert
"QueryPage" on page 115	ActivityQueryPage
"Update" on page 136	ActivityUpdate

Fields

Table 244 details the required and read-only fields for the activity object.

Table 244. Required and Read-Only Fields for the Activity Object

Field Name	Type
Subject	Required
Activity	Required
ActivityId	Read-only
AddressId	Read-only
CallType	Read-only
ContactFirstName	Read-only
ContactLastName	Read-only
LeadFirstName	Read-only
LeadLastName	Read-only
MedEdEventName	Read-only
OpportunityName	Read-only
FundRequest	Read-only
SmartCall	Read-only

Table 244. Required and Read-Only Fields for the Activity Object

Field Name	Type
AssignedQueue	Read-only
QueueHoldTime	Read-only
QueueStartTime	Read-only
TotalHoldTime	Read-only
ResolutionCode	Read-only
Audit Fields	Read-only

Table 245 details the pick map fields for the activity object.

Table 245. Pick Map Fields for the Activity Object

Pick Map Field	Maps To
OwnerAlias	OwnerId
AccountName	AccountId
AccountLocation	AccountId
AccountExternalSystemId	AccountId
AccountIntegrationId	AccountId
CampaignExternalSystemId	CampaignId
CampaignIntegrationId	CampaignId
CampaignName	CampaignId
LeadExternalSystemId	LeadId
LeadIntegrationId	LeadId
MedEdEventExternalSystemId	MedEdEventId
MedEdEventIntegrationId	MedEdEventId
OpportunityExternalSystemId	OpportunityId
OpportunityIntegrationId	OpportunityId
PortfolioExternalSystemId	PortfolioId
PortfolioIntegrationId	PortfolioId
ServiceRequestNumber	ServiceRequestId
ServiceRequestExternalSystemId	ServiceRequestId
ServiceRequestIntegrationId	ServiceRequestId

Table 245. Pick Map Fields for the Activity Object

Pick Map Field	Maps To
FundRequestExternalSystemId	FundRequestId
FundRequestIntegrationId	FundRequestId

Table 246 provides a list of the user key combinations for the activity object.

Table 246. User Key Fields on the Activity Object

User Key Field Combinations
Description
Id
ExternalSystemId

Table 247 details the picklists available for the activity object.

Table 247. Picklists Available for the Activity Object

Field Name
AccountName
AccountLocation
AccountIntegrationId
AccountExternalSystemId
OpportunityName
ServiceRequestNumber
ServiceRequestIntegrationId
ServiceRequestExternalSystemId
DelegatedByExternalSystemId
PrimaryContactIntegrationId
PrimaryContactExternalSystemId
MedEdEventIntegrationId
MedEdEventExternalSystemId
FundRequestExternalId
LeadIntegrationId

Table 247. Picklists Available for the Activity Object

Field Name
LeadExternalSystemId
CampaignIntegrationId
CampaignExternalSystemId
ActivitySubtype
ResolutionCode
PublishInternal
Status
Issue
Indication

Allotment Usage

The allotment usage object stores information about the historical allotment usage for a company.

Usage

The Allotment Usage service is only accessible to users with the Manage Company privilege.

There are various elements of the ListOfAllotmentUsage argument that specify the service allotment usage data:

- **AggregationType**. The aggregation type.
- **AggregationPeriodTotalUsage**. The total usage in the aggregation period.
- **AggregationPeriodTimestamp**. The aggregation period.
- **AllotmentLimitId**. The value set for the allotment by Customer Care.
- **AllotmentUnit**. The allotment unit of measure.
- **AllotmentType**. The allotment type.
- **AllotmentLimitName**. The allotment name.
- **AllotmentLimitDisplayName**. The allotment name in the language currently active for the company.

There are also various audit fields returned.

For more information about administration of service allotments, see the Oracle CRM On Demand online help.

Methods Called

Table 248 details the methods called by the Allotment Usage service.

Table 248. Methods Called by Allotment Usage Service

Method	Name as Defined in Service
"QueryPage" on page 115	AllotmentUsageQueryPage

Fields

Table 249 details the required and read-only fields for the allotment usage object.

Table 249. Required and Read-Only Fields for the Allotment Usage Object

Field Name	Type
AllotmentLimitName	Required
Audit Fields	Read-only

Table 250 details the user keys for the allotment usage object.

Table 250. User Keys for the Allotment Usage Object

Field Name
AllotmentLimitName

Asset

The asset object stores information on the assets held by your accounts, for example, the products that an account has purchased.

Child Objects

None

Usage

Oracle On Demand Web Services uses assets to manage products through their life cycle. It is also used by your accounts to register products, receive product news and literature, track warranty agreements, and receive recommendations on scheduled services.

Methods Called

Table 251 details the methods called by the Asset service.

Table 251. Methods Called by Asset Service

Method	Name as Defined in Service
"Delete" on page 105	AssetDelete
"Execute" on page 108	AssetExecute
"Insert" on page 112	AssetInsert
"QueryPage" on page 115	AssetQueryPage
"Update" on page 136	AssetUpdate

Fields

Table 252 details the required and read-only fields for the asset object.

Table 252. Required and Read-Only Fields for the Asset Object

Field Name	Type
ProductId	Required
ProductCategory	Read-only
PartNumber	Read-only
Type	Read-only
Status	Read-only
Audit Fields	Read-only

Table 253 details the pick map fields for the asset object.

Table 253. Pick Map Fields for the Asset Object

Pick Map Field	Maps To
AccountIntegrationId	AccountId
AccountExternalSystemId	AccountId
Account, AccountLocation	AccountId
ProductIntegrationId	ProductId
ProductExternalSystemId	ProductId
Product	ProductId

Table 254 provides a list of the user key combinations for the asset object.

Table 254. User Key Fields on the Asset Object

User Key Field Combinations
Id
ExternalSystemId
ProductId
ProductName
ProductExternalId

Table 255 details the picklists available for the asset object.

Table 255. Picklists Available for the Asset Object

Field Name
Warranty
Contract

Book

The book object provides a way of segmenting data according to the organizational units of your business, such as territories or products. Administrators can create book hierarchies based on how they want to organize your information, and then set up users to have the appropriate level of access to books.

NOTE: To download the Book WSDL, you must be given access to the Book object. If you do not have access to the Book object, it is not available to download from the Web Services Administration pages. For assistance in gaining access to the Book object, contact your Oracle CRM On Demand service provider.

Methods Called

Table 256 details the methods called by the Book service.

Table 256. Methods Called by Book Service

Method	Name as Defined in Service
“Delete” on page 105	BookDelete
“Execute” on page 108	BookExecute

Table 256. Methods Called by Book Service

Method	Name as Defined in Service
“Insert” on page 112	BookInsert
“QueryPage” on page 115	BookQueryPage
“Update” on page 136	BookUpdate

Fields

Table 257 details the required and read-only fields for the book object.

Table 257. Required and Read-Only Fields for the Book Object

Field Name	Type
BookName	Required
BookId	Read-only
Audit Fields	Read-only

Table 258 details the pick map fields for the book object.

Table 258. Pick Map Fields for the Book Object

Pick Map Field	Maps To
BookName	BookId

Table 259 provides a list of the user key combinations for the book object.

Table 259. User Key Fields on the Book Object

User Key Field Combinations
Id
BookName

Table 260 details the picklists available for the book object.

Table 260. Picklists Available for the Book Object

Field Name
BookType
ParentBookName

Campaign

The campaign object provides a mechanism for marketing products and services to customers and prospects. The campaign object is the primary way in which new products and services are marketed to customers and prospects.

Child Objects

Activity, Application, Attachment, Book, Financial Account, Financial Plan, Lead, MDF Request, Note, Policy

Methods Called

Table 261 details the methods called by the Campaign service.

Table 261. Methods Called by Campaign Service

Method	Name as Defined in Service
“Delete” on page 105	CampaignDelete
“Execute” on page 108	CampaignExecute
“Insert” on page 112	CampaignInsert
“QueryPage” on page 115	CampaignQueryPage
“Update” on page 136	CampaignUpdate

Fields

Table 262 details the required and read-only fields for the campaign object.

Table 262. Required and Read-Only Fields for the Campaign Object

Field Name	Type
CampaignName	Required
SourceCode	Required
Audit Fields	Read-only
CreatedByFullName	Read-only
LastUpdated	Read-only

Table 263 details the pick map field for the campaign object.

Table 263. Pick Map Field for the Campaign Object

Pick Map Field	Maps To
Owner	OwnerId

Table 264 provides a list of the user key combinations for the campaign object.

Table 264. User Key Fields on the Campaign Object

User Key Field Combinations
Id
ExternalSystemId
SourceCode
CampaignName

Table 265 details the picklists available for the campaign object.

Table 265. Picklists Available for the Campaign Object

Field Name
CampaignType
Status

Related Topic

Current User and Opportunity

Category

The category object allows you to logically sort products into groups, where each product is in some way related to the other products in the category.

The category object is equivalent to the Web Services V1.0 product category object.

Methods Called

Table 266 details the methods called by the Category service.

Table 266. Methods Called by Category Service

Method	Name as Defined in Service
"Delete" on page 105	CategoryDelete
"Execute" on page 108	CategoryExecute
"Insert" on page 112	CategoryInsert
"QueryPage" on page 115	CategoryQueryPage
"Update" on page 136	CategoryUpdate

Fields

Table 267 details the required and read-only fields for the category object.

Table 267. Required and Read-Only Fields for the Category Object

Field Name	Type
CategoryName	Required
Audit Fields	Read-only
ModifiedByFullName	Read-only

Table 268 details the pick map field for the category object.

Table 268. Pick Map Field for the Category Object

Pick Map Field	Maps To
ParentCategory	ParentCategoryId

Table 269 details the user keys for the category object.

Table 269. User Keys for the Category Object

Field Name
Id

Table 269. User Keys for the Category Object

Field Name
ExternaSystemId
Name

Contact

The contact object stores information on individuals with whom your organization has a relationship. It allows the user to store information on individuals who are external to your company, but who are associated with the business process. Contacts stored in the Oracle CRM On Demand database can also be associated with an account.

Adding and Updating Contact Addresses

The contact object has three types of address:

1. Primary address
2. Alternate address
3. Simple address

If the contact is associated with an account, the account's billing address becomes the contact's primary address, that is, the account's billing address fields populate the contact's primary address fields. The primary address fields are read-only in the contact record. The primary address may also be called the account address.

The alternate address is simply the contact's address. Alternate address fields at the parent level are editable and this address is added to address children with the Primary checkbox in the UI selected. The alternate address may also be called the contact address.

You can use either ContactInsert, ContactUpdate or ContactExecute calls to insert or update primary or alternate addresses for a contact. With ContactInsert or ContactUpdate calls, the contact parent node with primary and alternate address fields must be present, and there must not be any child node. If an address child is present, only a simple address can be added, and not primary and alternate addresses. If the address child or any other child node is not present, only primary and alternate addresses can be added, and not simple addresses.

With the ContactExecute call, both the contact parent node and address child node can be present along with primary and alternate address fields at the parent level.

NOTE: A *PrimaryAddressLine1* field is used to dynamically map the primary address field from an external application to the primary address field in Oracle CRM On Demand. For more information, see "Mapping Primary Address Fields Using Web Services" on page 35.

Child Objects

Account, Activity, Address, Attachment, Business Plan, Claim, Contact Best Time, Contact State License, Coverage, Custom Object 1 through 3, Deal Registration, Financial Account, Financial Account Holder, Financial Plan, Interests, Involved Party, Lead, Message Response, Note, Objective, Opportunity, Plan Contact, Policy, Policy Holder, Related Contact, Revenue, Service Request, Signature, Special Pricing Request, Contact Team.

Methods Called

Table 270 details the methods called by the Contact service.

Table 270. Methods Called by Contact Service

Method	Name as Defined in Service
"Delete" on page 105	ContactDelete
"Execute" on page 108	ContactExecute
"Insert" on page 112	ContactInsert
"QueryPage" on page 115	ContactQueryPage
"Update" on page 136	ContactUpdate

Fields

Table 271 details the required and read-only fields for the contact object.

Table 271. Required and Read-Only Fields for the Contact Object

Field Name	Type
FirstName	Required
LastName	Required
AlternateAddressId	Read-only
ContactConcatField	Read-only
ContactFullName	Read-only
Audit Fields	Read-only
Manager	Read-only
PrimaryAddressId	Read-only

Table 272 details the pick map fields for the contact object.

Table 272. Pick Map Fields for the Contact Object

Pick Map Field	Maps To
AccountName	AccountId
Owner	AssignedToAlias
SourceCampaignName	SourceCampaignId
ManagerExternalSystemId	ManagerId

Table 273 provides a list of the user key combinations for the contact object.

Table 273. User Key Fields on the Contact Object

User Key Field Combinations
FirstName and LastName
Id
ExternalSystemId

Table 274 details the picklists available for the contact object.

Table 274. Picklists Available for the Contact Object

Field Name
ContactType
LeadSource
MrMrs
BestTimeToCall
CallFrequency
CurrentInvestmentMix
Degree
ExperienceLevel
Gender
InvestmentHorizon
LifeEvent

Table 274. Picklists Available for the Contact Object

Field Name
MaritalStatus
MarketPotential
Objective
OwnOrRent
PrimaryGoal
RiskProfile
Route
Segment
Tier

Table 275 details a number of contact object fields that you must not use for customer integrations.

Table 275. Contact Object Fields That You Must Not Use

Field Name
PartyTypeCode
PartyUId
PersonUId

CustomObject

The CustomObject services expose the functionality of the CustomObject objects to external applications.

TIP: The reference information for each of Custom Objects follows the same pattern. In this topic, the information for CustomObject1 is given as an example.

The name of some CustomObject fields in the WSDL files for Web Services v2.0 differ from the names used in the WSDL files for Web Services v1.0. For example, the Web Services v1.0 fields ParentExternalSystemId and ParentName correspond to CustomObject1ExternalSystemId and CustomObject1Name for Web Services v2.0.

NOTE: To download the CustomObject WSDL file, you must be given access to the relevant CustomObject object. If you do not have access to the CustomObject object, it is not available to download from the Web Services Administration page or available to use Web service calls. For assistance in gaining access to the CustomObject objects, contact your Oracle CRM On Demand service provider.

Child Objects

Attachment.

Methods Called

Table 276 details the methods called by the CustomObject1 service. The methods for the other Custom Objects follow the same pattern.

Table 276. Methods Called by CustomObJect1 Service

Method	Name as Defined in Service
"Delete" on page 105	CustomObject1Delete
"Execute" on page 108	CustomObject1Execute
"Insert" on page 112	CustomObject1Insert
"QueryPage" on page 115	CustomObject1QueryPage
"Update" on page 136	CustomObject1Update

Fields

Table 277 details the required and read-only fields for the CustomObject1 object.

Table 277. Required and Read-Only Fields for the CustomObject1 Object

Field Name	Type
Id	Required
CustomObject1ExternalSystemID	Required
CustomObject1IntegrationId	Required

Table 278 details the pick map fields for the CustomObject1 object.

Table 278. Pick Map Fields for the CustomObject1 Object

Pick Map Field	Maps To
AccountExternalId	AccountId
AccountIntegrationId	AccountId
AccountName	AccountId
ActivityExternalId	ActivityId
ActivityIntegrationId	ActivityId
ActivityName	ActivityId

Table 278. Pick Map Fields for the CustomObject1 Object

Pick Map Field	Maps To
CampaignExternalId	CampaignId
CampaignIntegrationId	CampaignId
CampaignName	CampaignId
ContactExternalId	ContactId
ContactFirstName	ContactId
ContactFullName	ContactId
ContactIntegrationId	ContactId
ContactLastName	ContactId
CustomObject*n*ExternalId	CustomObjectnId
CustomObject*n*IntegrationId	CustomObject*n*Id
CustomObject*n*Name	CustomObject*n*Id
DealerName	DealerId
HouseholdExternalId	HouseholdId
HouseholdIntegrationId	HouseholdId
HouseholdName	HouseholdId
LeadExternalId	LeadId
LeadFirstName	LeadId
LeadFullName	LeadId
LeadIntegrationId	LeadId
LeadLastName	LeadId
Owner	OwnerId
OpportunityExternalId	OpportunityId
OpportunityIntegrationId	OpportunityId
OpportunityName	OpportunityId
ParentExternalSystemId	ParentId
ParentIntegrationId	ParentId
PortfolioAccountNumber	PortfolioId
ProductExternalId	ProductId
ProductIntegrationId	ProductId
ProductName	ProductId
SolutionExternalId	SolutionId

Table 278. Pick Map Fields for the CustomObject1 Object

Pick Map Field	Maps To
SolutionIntegrationId	SolutionId
SolutionTitle	SolutionId
ServiceRequestExternalId	ServiceRequestId
ServiceRequestIntegrationId	ServiceRequestId
ServiceRequestName	ServiceRequestId
VIN	VehicleId

Table 279 provides a list of the user key combinations for the CustomObject1 object.

Table 279. User Key Fields on the CustomObject1 Object

User Key Field Combinations
Id
ExternalSystemId

Exchange Rate

The Exchange rate object stores information about exchange rates for the currencies used by a company.

Usage

Administrators can create new exchange rates, update, or delete existing exchange rates, and query the exchange rate history for their company.

All fields are required when inserting new exchange rate records. For insert operations, the ToCurrencyCode value must be the company's default currency value.

Only the FromCurrencyCode, ToCurrencyCode, and ExchangeDate fields are required for update delete operations. However, these fields are not editable for an update operation.

The date value in the ExchangeDate field must be specified in the format: *yyyy-mm-dd*. For example, a value of 2011-10-12 specifies the date as 12th October 2011.

The ExchangeRateHistory child object can be queried to give the history of exchange rates for the company.

Methods Called

Table 280 details the methods called by the Exchange Rate service.

Table 280. Methods Called by Exchange Rate Service

Method	Name as Defined in Service
"Delete" on page 105	ExchangeRateDelete
"Execute" on page 108	ExchangeRateExecute
"Insert" on page 112	ExchangeRateInsert
"QueryPage" on page 115	ExchangeRateQueryPage
"Update" on page 136	ExchangeRateUpdate

Fields

Table 281 details the required and read-only fields for the exchange rate object.

Table 281. Required and Read-Only Fields for the Exchange Rate Object

Field Name	Type
FromCurrencyCode	Required
ToCurrencyCode	Required
ExchangeDate	Required
ExchangeRate	Required

Table 282 provides a list of the user key combinations for the exchange rate object.

Table 282. User Key Fields on the Exchange Rate Object

User Key Field Combinations
FromCurrencyCode
ToCurrencyCode
ExchangeDate

Group

The Group object allows you to create groups to which users can be added. Users can only be a member of one group, and groups can contain many users.

The group object is equivalent to the Web Services V1.0 user group object.

Methods Called

Table 283 details the methods called by the Group service.

Table 283. Methods Called by Group Service

Method	Name as Defined in Service
“Delete” on page 105	GroupDelete
“Execute” on page 108	GroupExecute
“Insert” on page 112	GroupInsert
“QueryPage” on page 115	GroupQueryPage
“Update” on page 136	GroupUpdate

Fields

Table 284 details the required and read-only fields for the group object.

Table 284. Required and Read-Only Fields for the Group Object

Field Name	Type
Name	Required
UserGroupId	Read-only
Audit Fields	Read-only

Table 285 details the pick map field for the group object.

Table 285. Pick Map Field for the Group Object

Pick Map Field	Maps To
UserIntegrationId	UserId
UserExternalSystemId	UserId

Table 286 provides a list of the user key combinations for the group object.

Table 286. User Key Fields on the Group Object

User Key Field Combinations
Id
ExternalSystemId

Table 286. User Key Fields on the Group Object

User Key Field Combinations
Name

Lead

The lead object stores information on a company or individual with whom an opportunity can be created. It allows the user to identify the companies that might be interested in a product or service. Leads are usually generated as part of a marketing campaign.

NOTE: A *PrimaryAddressLine1* field is used to dynamically map the primary address field from an external application to the primary address field in Oracle CRM On Demand. For more information, see "Mapping Primary Address Fields Using Web Services" on page 35.

Child Objects

Activity, Attachment, Custom Object 1 through 3.

Methods Called

Table 287 details the methods called by the Lead service.

Table 287. Methods Called by Lead Service

Method	Name as Defined in Service
"Delete" on page 105	LeadDelete
"Execute" on page 108	LeadExecute
"Insert" on page 112	LeadInsert
"QueryPage" on page 115	LeadQueryPage
"Update" on page 136	LeadUpdate

Fields

Table 288 details the required and read-only fields for the lead object.

Table 288. Required and Read-Only Fields for the Lead Object

Field Name	Type
FirstName	Required
LastName	Required
LeadOwner	Required

Table 288. Required and Read-Only Fields for the Lead Object

Field Name	Type
ContactFullName	Read-only
Audit Fields	Read-only
LastUpdated	Read-only
LeadConcatField	Read-only
LeadFullName	Read-only
ReferredById	Read-only

Table 289 details the pick map fields for the lead object.

Table 289. Pick Map Fields for the Lead Object

Pick Map Field	Maps To
Campaign	CampaignId
OpportunityName	OpportunityId
Owner	OwnerId
AccountExternalSystemId	AccountId
OpportunityExternalSystemId	OpportunityId
ContactExternalSystemId	ContactId
CampaignExternalSystemId	CampaignId
ReferredByExternalSystemId	ReferredById

Table 290 provides a list of the user key combinations of the lead object.

Table 290. User Key Fields on the Lead Object

User Key Field Combinations
Id
ExternalSystemId
LeadFirstName and LeadLastName

Table 291 details the picklists available for the lead object.

Table 291. Picklists Available for the Lead Object

Field Name
Country
MrMrs
Rating
Source
StateProvince
Status

Note

The note object stores information about the notes available in the Message Center in the Oracle CRM On Demand application. The notes can be sent from users or can store extra information (as a note) on a parent object. This allows employees who are working on a particular record to add extra information as they see fit. For example, when talking to a contact, an employee might notice that the contact is not happy with a service provided. The employee can record this information in a note so that any other employees who talk to the contact are aware of the contact's dissatisfaction.

Methods Called

Table 292 details the methods called by the Note service.

Table 292. Methods Called by Note Service

Method	Name as Defined in Service
"Delete" on page 105	NoteDelete
"Execute" on page 108	NoteExecute
"Insert" on page 112	NoteInsert
"QueryPage" on page 115	NoteQueryPage
"Update" on page 136	NoteUpdate

Fields

Table 293 details the required and read-only fields for the note object.

Table 293. Required and Read-Only Fields for the Note Object

Field Name	Type
Subject	Required
NoteId	Read-only
OwnerId	Read-only
OwnerAlias	Read-only
ParentNoteId	Read-only
SourceId	Read-only
SourceName	Read-only
Audit Fields	Read-only

Table 294 provides a list of the user key combinations for the note object.

Table 294. User Key Fields on the Note Object

User Key Field Combinations
NoteId

Opportunity

The opportunity object allows employees to identify and record a potential revenue-generating event that has arisen with an account or contact. Opportunities can be generated from marketing campaigns when leads indicate that they are interested in a product or service that has been offered.

Child Objects

Activity, Attachment, Competitor, Contact, Contact Role, Custom Object 1 through 3, Deal Registration, Lead, Note, Opportunity Child, Opportunity Team, Partner, Plan Opportunity, Revenue, Special Pricing Request.

Methods Called

Table 295 details the methods called by the Opportunity service.

Table 295. Methods Called by Opportunity Service

Method	Name as Defined in Service
"Delete" on page 105	OpportunityDelete
"Execute" on page 108	OpportunityExecute
"Insert" on page 112	OpportunityInsert
"QueryPage" on page 115	OpportunityQueryPage
"Update" on page 136	OpportunityUpdate

Fields

Table 296 details the required and read-only fields for the opportunity object.

Table 296. Required and Read-Only Fields for the Opportunity Object

Field Name	Type
AccountId	Required
CloseDate	Required
OpportunityName	Required
SalesStage	Required
Audit Fields	Read-only
LastUpdated	Read-only
OpportunityConcatField	Read-only

Table 297 details the pick map fields for the opportunity object.

Table 297. Pick Map Fields for the Opportunity Object

Pick Map Field	Maps To
Owner	OwnerId
AccountExternalSystemId	AccountId
Territory	TerritoryId
KeyContactIntegrationId	KeyContactId
KeyContactExternalSystemId	KeyContactId

Table 298 provides a list of the user key combinations for the opportunity object.

Table 298. User Key Fields on the Opportunity Object

User Key Field Combinations
Id
ExternalSystemId
Name

Table 299 details the picklists available for the opportunity object.

Table 299. Picklists Available for the Opportunity Object

Field Name
LeadSource
Priority
Probability
ReasonWonLost
Status
Type
Year
Make
Model

Product

The product object allows you to define and record details about a product or service that your company sells to its customers, including information on product price, category, and so on.

Child Objects

Price List Line Item

Methods Called

Table 300 details the methods called by the Product service.

Table 300. Methods Called by Product Service

Method	Name as Defined in Service
"Execute" on page 108	ProductExecute
"Insert" on page 112	ProductInsert
"QueryPage" on page 115	ProductQueryPage
"Update" on page 136	ProductUpdate

Fields

Table 301 details the required and read-only fields for the product object.

Table 301. Required and Read-Only Fields for the Product Object

Field Name	Type
ProductName	Required
Audit Fields	Read-only

Table 302 details the pick map field for the product object.

Table 302. Pick Map Field for the Product Object

Pick Map Field	Maps To
ParentCategory	ParentCategoryId

Table 303 details the user keys for the product object.

Table 303. User Keys for the Product Object

Field Name
ProductName
Id
ExternalSystemId

Table 304 details the picklists available for the product object.

Table 304. Picklists Available for the Product Object

Field Name
BodyStyle
Category
Class
CurrencyCode
DoorStyle
Engine
Make
Model
PriceType
ProductType
Revision
Status
SubType
TherapeuticClass
Transmission
Trim

Service Allotment

The service allotment object stores information about the service allotments for a company.

Usage

The Service Allotment service is only accessible to users with the Manage Company privilege.

There is only one method, ServiceAllotmentQueryPage, which returns the service allotment data for the company.

There are various elements of the ListOfServiceAllotment argument that specify the service allotment data:

- **Name**. The allotment name.
- **Value**. The value set for the allotment by Customer Care

- **CurrentUsage**. The amount of the allotment currently used.
- **AlertThreshold.** The threshold value at which a warning email is sent.
- **AlertEnabled**. Whether an alert is enabled for the allotment.
- **EmailAlertList**.The email addresses of the users who will receive a warning email.
- **AllotmentEmailSentDate.** The date when an email notifying full usage of the allotment was sent.
- **ThresholdEmailSentDate**. The date when a warning email for the allotment was sent.
- **DisplayName**. The allotment name in the language currently active for the company.
- **AllotmentUnit**. The allotment unit of measure.
- **AllotmentType**. The type of allotment.
- **RemainingUsage**. The amount of the allotment remaining.
- **Description**. A description of the allotment.
- **ListOfAllotmentUsage**. The allotment usage data:
  - **AggregationType**. The aggregation type.
  - **AggregationPeriodTotalUsage**. The total usage in the aggregation period.
  - **AggregationPeriodTimestamp**. The aggregation period.
  - **AllotmentLimitId**. The value set for the allotment by Customer Care.
  - **AllotmentUnit**. The allotment unit of measure.
  - **AllotmentType**. The allotment type.
  - **AllotmentLimitName**. The allotment name.
  - **AllotmentLimitDisplayName**. The allotment name in the language currently active for the company.

There are also various audit fields returned.

For more information about administration of service allotments, see the Oracle CRM On Demand online help.

Methods Called

Table 305 details the methods called by the Service Allotment service.

Table 305. Methods Called by Service Allotment Service

Method	Name as Defined in Service
“QueryPage” on page 115	ServiceAllotmentQueryPage

Fields

Table 306 details the required and read-only fields for the service allotment object.

Table 306. Required and Read-Only Fields for the Service Allotment Object

Field Name	Type
Name	Required
Audit Fields	Read-only

Table 307 details the user keys for the service allotment object.

Table 307. User Keys for the Service Allotment Object

Field Name
Name

Service Request

The service request object allows customers to request information or assistance with a problem related to products or services purchased from your company. Service requests can be ranked for severity and prioritized accordingly.

Child Objects

Attachment, Note.

Methods Called

Table 308 details the methods called by the Service request service.

Table 308. Methods Called by Service Request Service

Method	Name as Defined in Service
“Delete” on page 105	ServiceRequestDelete
“Execute” on page 108	ServiceRequestExecute
“Insert” on page 112	ServiceRequestInsert
“QueryPage” on page 115	ServiceRequestQueryPage
“Update” on page 136	ServiceRequestUpdate

Fields

Table 309 details the required and read-only fields for the service request object.

Table 309. Required and Read-Only Fields for the Service Request Object

Field Name	Type
ContactEmail	Read-only
ContactFirstName	Read-only
ContactFullName	Read-only
ContactLastName	Read-only
Audit Fields	Read-only
LastUpdated	Read-only
ServiceRequestConcatId	Read-only

Table 310 details the pick map fields for the service request object.

Table 310. Pick Map Fields for the Service Request Object

Pick Map Field	Maps To
Owner	OwnerId
AccountExternalSystemId	AccountId
AssetIntegrationId	AssetId
AssetExternalSystemId	AssetId
ProductExternalSystemId	ProductId

Table 311 provides a list of the user key combinations for the service request object.

Table 311. User Key Fields on the Service Request Object

User Key Field Combinations
Id
ExternalSystemId
SRNumber

Table 312 details the picklists available for the service request object.

Table 312. Picklists Available for the Service Request Object

Field Name
Area
Cause
Priority
Source
Status
Type

Solution

The solution object stores information on solutions to customer problems or service requests. Solutions can be reused if the same problem is identified with a product or service. This prevents the duplication of work for customer service representatives.

Child Object

Attachment, Book, Message Response, Messaging Plan, Messaging Plan Item, Messaging Plan Item Relation.

Methods Called

Table 313 details the methods called by the Solution service.

Table 313. Methods Called by Solution Service

Method	Name as Defined in Service
“Delete” on page 105	SolutionDelete
“Execute” on page 108	SolutionExecute
“Insert” on page 112	SolutionInsert
“QueryPage” on page 115	SolutionQueryPage
“Update” on page 136	SolutionUpdate

Fields

Table 314 details the required and read-only fields for the solution object.

Table 314. Required and Read-Only Fields for the Solution Object

Field Name	Type
Title	Required
Audit Fields	Read-only
CreatorId	Read-only
LastUpdated	Read-only

Table 315 provides a list of the user key combinations for the solution object.

Table 315. User Key Fields on the Solution Object

User Key Field Combinations
Name
Id
ExternalSystemId

Table 316 details the picklists available for the solution object.

Table 316. Picklists Available for the Solution Object

Field Name
Area
Cause
Priority
Source
Status
Type

Territory

The territory object allows you to store information about the sales territory that is assigned to a user. This information includes the territory name, a description, the currency code, and the sales quota for the territory.

Methods Called

Table 317 details the methods called by the Territory service.

Table 317. Methods Called by Territory Service

Method	Name as Defined in Service
"Delete" on page 105	TerritoryDelete
"Execute" on page 108	TerritoryExecute
"Insert" on page 112	TerritoryInsert
"QueryPage" on page 115	TerritoryQueryPage
"Update" on page 136	TerritoryUpdate

Fields

Table 318 details the required and read-only fields for the territory object.

Table 318. Required and Read-Only Fields for the Territory Object

Field Name	Type
TerritoryName	Required
Territory	Read-only
Audit Fields	Read-only

Table 319 details the pick map field for the territory object.

Table 319. Pick Map Field for the Territory Object

Pick Map Field	Maps To
ParentTerritoryIntegrationId	ParentTerritoryId
ParentTerritoryExternalSystemId	ParentTerritoryId
ParentTerritoryId	ParentTerritoryId

Table 320 details the user key for the territory object.

Table 320. User Key for the Territory Object

Field Name
Id
ExternalSystemId
TerritoryName

Table 321 details the picklist fields available for the territory object.

Table 321. Picklists Available for the Territory Object

Field Name
ParentTerritoryExternalSystemId
ParentTerritoryIntegrationId

User

The user object allows you to define and record details of all users in Oracle CRM On Demand, for example, name, position, contact details, manager, and so on. It enables queries to be run on all users, and enables an administrator to insert and update a user's profile.

Usage

The UserLoginId and UserSignInId fields must be used as follows:

- **UserLoginId**. Used for creating user records through the User Web service.
- **UserSignInId**. Used as the user name for logging in and authenticating using Web services. Also, used for queries, as using UserLoginId is not allowed for queries.

Child Object

Delegated User

Methods Called

Table 322 details the methods called by the User service.

Table 322. Methods Called by User Service

Method	Name as Defined in Service
“Execute” on page 108	UserExecute
“Insert” on page 112	UserInsert
“QueryPage” on page 115	UserQueryPage
“Update” on page 136	UserUpdate

Fields

Table 323 details the required and read-only fields for the user object.

Table 323. Required and Read-Only Fields for the User Object

Field Name	Type
FirstName	Required
LastName	Required
UserLoginId	Required
UserSignInId	Required
Alias	Required
EmailAddr	Required
Role	Required
Status	Required
Audit Fields	Read-only
LastSignInDateTime	Read-only
ManagerFullName	Read-only

Table 324 details the pick map field for the user object.

Table 324. Pick Map Field for the User Object

Pick Map Field	Maps To
Role	RoleId

Table 325 provides a list of user key combinations for the user object.

Table 325. User Key Fields on the User Object

User Key Field Combinations
Id
ExternalSystemId
Alias

Partner Relationship Management Edition Objects (Web Services v2.0)

This topic includes the parent objects available with Oracle CRM On Demand Partner Relationship Management Edition. For these parent objects, all child objects are also accessible.

NOTE: To download WSDL files for these objects, you must be given access to the object. If you do not have access to the object, it is not available to download from the Web Services Administration page or available to use the Web service calls. For assistance in gaining access to the objects, contact your Oracle CRM On Demand service provider.

The following objects are detailed in this topic:

- "Accreditation" on page 437
- "Accreditation Request" on page 439
- "Application" on page 440
- "Certification" on page 442
- "Certification Request" on page 445
- "Course" on page 447
- "Course Enrollment" on page 449
- "Deal Registration" on page 451
- "Exam" on page 453
- "Exam Registration" on page 455
- "Fund" on page 457
- "MDF Request" on page 459
- "Partner" on page 461
- "Price List" on page 463
- "Price List Line Item" on page 464
- "Special Pricing Request" on page 466

- "Special Pricing Request Line Item" on page 468

Accreditation

The accreditation object stores information about a partner company's accreditation for selling products, providing services, or both. Accreditations are an acknowledgement of a partner company's skill, competence, and training in selling a particular line of products or offering services such as design, installation, upgrade, and so on. A partner company is awarded an accreditation upon satisfying several requirements; one of which is usually a minimum number (or percentage) of employees with specific certifications. In Oracle CRM On Demand, an accreditation can be held only by a partner company. Individuals cannot hold accreditations; instead, they hold certifications.

Child Objects

Accreditation, Accreditation Request, Attachment, Book, Category, Certification, Custom Object 01-03, Custom Object 4 through 15, Product, Solution.

Methods Called

Table 326 details the methods called by the Accreditation service.

Table 326. Methods Called by Accreditation Service

Method	Name as Defined in Service
"Delete" on page 105	AccreditationDelete
"Execute" on page 108	AccreditationExecute
"Insert" on page 112	AccreditationInsert
"QueryPage" on page 115	AccreditationQueryPage
"Update" on page 136	AccreditationUpdate

Fields

Table 327 details the required and read-only fields for the accreditation object.

Table 327. Required and Read-Only Fields for the Accreditation Object

Field Name	Type
Name	Required
Status	Required
OwnerId	Required
Audit Fields	Read-only

Table 328 details the pick map fields for the accreditation object.

Table 328. Pick Map Fields for the Accreditation Object

Pick Map Field	Maps To
PrimaryProductProductCategory, PrimaryProductExternalSystemId, PrimaryProductDescription, PrimaryProductName, PrimaryProductPartNumber, PrimaryProductStatus, PrimaryProductIntegrationId, PrimaryProductType	PrimaryProductId
PrimaryProductCategoryName	PrimaryProductCategoryId
ContactEmail, ContactFirstName, ContactLastName, ContactFullName, ContactIntegrationId, ContactAccountName, ContactExternalSystemId	ContactId
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId

Table 329 provides a list of the user key combinations for the accreditation object.

Table 329. User Key Fields on the Accreditation Object

User Key Field Combinations
Id
Name

Table 330 details the picklists available for the accreditation object.

Table 330. Picklists Available for the Accreditation Object

Field Name
PaymentOption
Status
Type

Accreditation Request

The accreditation request object stores information about a partner company's request for accreditation.

Child Objects

None

Methods Called

Table 331 details the methods called by the Accreditation Request service.

Table 331. Methods Called by Accreditation Request Service

Method	Name as Defined in Service
“Delete” on page 105	AccreditationRequestDelete
“Execute” on page 108	AccreditationRequestExecute
“Insert” on page 112	AccreditationRequestInsert
“QueryPage” on page 115	AccreditationRequestQueryPage
“Update” on page 136	AccreditationRequestUpdate

Fields

Table 332 details the required and read-only fields for the accreditation request object.

Table 332. Required and Read-Only Fields for the Accreditation Request Object

Field Name	Type
AccreditationId	Required
PartnerId	Required
OwnerId	Required
Audit Fields	Read-only

Table 333 details the pick map fields for the accreditation request object.

Table 333. Pick Map Fields for the Accreditation Request Object

Pick Map Field	Maps To
AccreditationName, AccreditationNum, AccreditationStatus, AccreditationType, AccreditationExternalSystemId	AccreditationId
PartnerChannelAccountManagerAlias, PartnerIntegrationId, PartnerExternalSystemId, PartnerLocation, PartnerMainPhone, PartnerName	PartnerId
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId
PlanName, PlanType, PlanStatus, PlanDescription, PlanExternalSystemId	PlanId

Table 334 provides a list of the user key combinations for the accreditation request object.

Table 334. User Key Fields on the Accreditation Request Object

User Key Field Combinations
Id
AccreditationId, PartnerId and RequestDate

Table 335 details the picklists available for the accreditation request object.

Table 335. Picklists Available for the Accreditation Request Object

Field Name
Status

Application

The application object is the means by which a new or existing partner applies for a partnership with the brand owner. The brand owner then reviews, approves, or rejects the application.

Child Objects

Activity, Book.

Methods Called

Table 336 details the methods called by the Application service.

Table 336. Methods Called by Application Service

Method	Name as Defined in Service
"Delete" on page 105	ApplicationDelete
"Execute" on page 108	ApplicationExecute
"Insert" on page 112	ApplicationInsert
"QueryPage" on page 115	ApplicationQueryPage
"Update" on page 136	ApplicationUpdate

Fields

Table 337 details the required and read-only fields for the application object.

Table 337. Required and Read-Only Fields for the Application Object

Field Name	Type
ContactFirstName	Required
ContactLastName	Required
SubmissionStatus	Required
CompanyName	Required
CurrencyCode	Read-only
Audit Fields	Read-only

Table 338 details the pick map fields for the application object.

Table 338. Pick Map Fields for the Application Object

Pick Map Field	Maps To
CurrentApproverAlias	CurrentApproverId
AssociatedPartnerName	PartnerId
PartnerProgramProgramName	PartnerProgramId

Table 338. Pick Map Fields for the Application Object

Pick Map Field	Maps To
CampaignName	CampaignId
OwnerAlias	OwnerId
ProcessedByAlias	ProcessedById

Table 339 provides a list of the user key combinations for the application object.

Table 339. User Key Fields on the Application Object

User Key Field Combinations
Id
ExternalSystemId
CompanyName

Table 340 details the picklists available for the application object.

Table 340. Picklists Available for the Application Object

Field Name
ApprovalStatus
PartnerType
ProgramLevel
RejectReason
Source
SubmissionStatus

Certification

The certification object stores information about an individual's competency in a specific set of skills. In Oracle CRM On Demand, certifications are always associated with an individual, in contrast to accreditations, which are always associated with a partner company. A certification is proof of an individual's competency in a specific set of skills.

Child Objects

Accreditation, Attachment, Book, Category, Certification, Certification Request, Course, Custom Object 01-03, Custom Object 4 through 15, Exam, Product, Solution.

Methods Called

Table 341 details the methods called by the Certification service.

Table 341. Methods Called by Certification Service

Method	Name as Defined in Service
"Delete" on page 105	CertificationDelete
"Execute" on page 108	CertificationExecute
"Insert" on page 112	CertificationInsert
"QueryPage" on page 115	CertificationQueryPage
"Update" on page 136	CertificationUpdate

Fields

Table 342 details the required and read-only fields for the certification object.

Table 342. Required and Read-Only Fields for the Certification Object

Field Name	Type
Name	Required
Status	Required
OwnerId	Required
Audit Fields	Read-only

Table 343 details the pick map fields for the certification object.

Table 343. Pick Map Fields for the Certification Object

Pick Map Field	Maps To
PrimaryProductProductCategory, PrimaryProductExternalSystemId, PrimaryProductDescription, PrimaryProductName, PrimaryProductPartNumber, PrimaryProductStatus, PrimaryProductIntegrationId, PrimaryProductType	PrimaryProductId
PrimaryProductCategoryName	PrimaryProductCategoryId
ContactEmail, ContactFirstName, ContactLastName, ContactFullName, ContactIntegrationId, ContactAccountName, ContactExternalSystemId	ContactId
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId

Table 344 provides a list of the user key combinations for the certification object.

Table 344. User Key Fields on the Certification Object

User Key Field Combinations
Id
Name

Table 345 details the picklists available for the certification object.

Table 345. Picklists Available for the Certification Object

Field Name
PaymentOption
Status
Type

Certification Request

The certification request object stores information about an individual's request for certification.

Child Objects

None.

Methods Called

Table 346 details the methods called by the Certification Request service.

Table 346. Methods Called by Certification Request Service

Method	Name as Defined in Service
"Delete" on page 105	CertificationRequestDelete
"Execute" on page 108	CertificationRequestExecute
"Insert" on page 112	CertificationRequestInsert
"QueryPage" on page 115	CertificationRequestQueryPage
"Update" on page 136	CertificationRequestUpdate

Fields

Table 347 details the required and read-only fields for the certification request object.

Table 347. Required and Read-Only Fields for the Certification Request Object

Field Name	Type
CertificationId	Required
ContactId	Required
OwnerId	Required
Audit Fields	Read-only

Table 348 details the pick map fields for the certification request object.

Table 348. Pick Map Fields for the Certification Request Object

Pick Map Field	Maps To
CertificationName, CertificationNum, CertificationStatus, CertificationType, CertificationExternalSystemId	CertificationId
ContactEmail, ContactFirstName, ContactLastName, ContactFullName, ContactIntegrationId, ContactAccountName, ContactExternalSystemId	ContactId
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId
PlanName, PlanType, PlanStatus, PlanDescription, PlanExternalSystemId	PlanId

Table 349 provides a list of the user key combinations for the certification request object.

Table 349. User Key Fields on the Certification Request Object

User Key Field Combinations
Id
CertificationId, ContactId, and RequestDate

Table 350 details the picklists available for the certification request object.

Table 350. Picklists Available for the Certification Request Object

Field Name
Status

Course

The course object stores information about a training course. A course is a structured educational programme offered with the primary aim of imparting a specific set of skills to an individual. Many companies, especially those that deal with complex products or services, require their partners and resellers to be adequately trained in and familiar with what they sell. Typically, courses focus on a specific product or product family; however, at times, they might also deal with generic concepts or principles. Courses may either be theoretical or practical, or a blend of both.

Child Objects

Attachment, Book, Category, Certification, Course, Course Enrollment, Custom Object 01-03, Custom Object 4 through 15, Exam, Partner, Product, Solution.

Methods Called

Table 351 details the methods called by the Course service.

Table 351. Methods Called by Course Service

Method	Name as Defined in Service
"Delete" on page 105	CourseDelete
"Execute" on page 108	CourseExecute
"Insert" on page 112	CourseInsert
"QueryPage" on page 115	CourseQueryPage
"Update" on page 136	CourseUpdate

Fields

Table 352 details the required and read-only fields for the course object.

Table 352. Required and Read-Only Fields for the Course Object

Field Name	Type
Name	Required
Status	Required
OwnerId	Required
Audit Fields	Read-only

Table 353 details the pick map fields for the course object.

Table 353. Pick Map Fields for the Course Object

Pick Map Field	Maps To
PrimaryProductProductCategory, PrimaryProductExternalSystemId, PrimaryProductDescription, PrimaryProductName, PrimaryProductPartNumber, PrimaryProductStatus, PrimaryProductIntegrationId, PrimaryProductType	PrimaryProductId
ProductCategoryName	ProductCategoryId
ContactEmail, ContactFirstName, ContactLastName, ContactFullName, ContactIntegrationId, ContactAccountName, ContactExternalSystemId	ContactId
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId
ExamType, ExamStatus, ExamName, ExamID, ExamExternalSystemId	ExamId
OwnerFirstName, OwnerLastName, OwnerUserSignInId, OwnerAlias, OwnerFullName, OwnerIntegrationId, OwnerExternalSystemId, OwnerEMailAddr	OwnerId

Table 354 provides a list of the user key combinations for the course object.

Table 354. User Key Fields on the Course Object

User Key Field Combinations
Id
CourseId, CandidateId, and EnrollDate

Table 355 details the picklists available for the course object.

Table 355. Picklists Available for the Course Object

Field Name
Type
Status
Format
Medium
PaymentOption

Course Enrollment

The course enrollment object stores information about an individual's enrollment for a course.

Child Objects

None

Methods Called

Table 356 details the methods called by the Course Enrollment service.

Table 356. Methods Called by Course Enrollment Service

Method	Name as Defined in Service
"Delete" on page 105	CourseEnrollmentDelete
"Execute" on page 108	CourseEnrollmentExecute
"Insert" on page 112	CourseEnrollmentInsert
"QueryPage" on page 115	CourseEnrollmentQueryPage
"Update" on page 136	CourseEnrollmentUpdate

Fields

Table 357 details the required and read-only fields for the course enrollment object.

Table 357. Required and Read-Only Fields for the Course Enrollment Object

Field Name	Type
CourseId	Required
CandidateId	Required
OwnerId	Required
Audit Fields	Read-only

Table 358 details the pick map fields for the course enrollment object.

Table 358. Pick Map Fields for the Course Enrollment Object

Pick Map Field	Maps To
CourseName, CourseStatus, CourseType, CourseID, CourseExternalSystemId	CourseId
CandidateContactEmail, CandidateContactFirstName, CandidateContactLastName, CandidateContactFullName, CandidateIntegrationId, CandidateAccountName, CandidateExternalSystemId	CandidateId
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId
OfferingPartnerChannelAccountManagerAlias, OfferingPartnerIntegrationId, OfferingPartnerExternalSystemId, OfferingPartnerLocation, OfferingPartnerMainPhone, OfferingPartnerPartnerName	OfferingPartnerId
OwnerFirstName, OwnerLastName, OwnerUserSignInId, OwnerAlias, OwnerFullName, OwnerIntegrationId, OwnerExternalSystemId, OwnerEMailAddr	OwnerId

Table 359 provides a list of the user key combinations for the course enrollment object.

Table 359. User Key Fields on the Course Enrollment Object

User Key Field Combinations
CourseId
CandidateId
EnrollDate

Table 360 details the picklists available for the course enrollment object.

Table 360. Picklists Available for the Course Enrollment Object

Field Name
Status

Deal Registration

The deal registration object stores information about the process by which a partner company requests exclusive rights to an opportunity from the brand owner.

Child Objects

Activity, Book, Custom Object 01 -03, Custom Object 4 through 15, Special Pricing Request.

Methods Called

Table 361 details the methods called by the Deal Registration service.

Table 361. Methods Called by Deal Registration Service

Method	Name as Defined in Service
“Delete” on page 105	DealRegistrationDelete
“Execute” on page 108	DealRegistrationExecute
“Insert” on page 112	DealRegistrationInsert
“QueryPage” on page 115	DealRegistrationQueryPage
“Update” on page 136	DealRegistrationUpdate

Fields

Table 362 details the required and read-only fields for the deal registration object.

Table 362. Required and Read-Only Fields for the Deal Registration Object

Field Name	Type
DealRegistrationName	Required
Type	Required
PartnerId	Required
SubmissionStatus	Required
DealRegistrationId	Read-only
ConvertedtoOpportunity	Read-only
Audit Fields	Read-only

Table 363 details the pick map fields for the deal registration object.

Table 363. Pick Map Fields for the Deal Registration Object

Pick Map Field	Maps To
PartnerExternalSystemId	PartnerId
PartnerName	PartnerId

Table 364 provides a list of the user key combinations for the deal registration object.

Table 364. User Key Fields on the Deal Registration Object

User Key Field Combinations
ExternalSystemId
DealRegistrationName and PartnerId
Id

Table 365 details the picklists available for the deal registration object.

Table 365. Picklists Available for the Deal Registration Object

Field Name
Type
Status

Table 365. Picklists Available for the Deal Registration Object

Field Name
State
Country
Salutation
ApprovalStatus
RejectReason

Exam

The exam object stores information about a training examination. Exams measure the competency of an individual in a specific set of skills. They are usually the next logical step for an individual after completing a course. Typically, a course culminates in an exam (or exams) and an individual is expected to register and appear for the exams upon completing the course. In fact, many companies consider that a course has been successfully completed only if the exams are passed with at least the minimum passing score.

Child Objects

Attachment, Book, Category, Certification, Course, Custom Object 01-03, Custom Object 4 through 15, Exam, Exam Registration, Partner, Product, Solution.

Methods Called

Table 366 details the methods called by the Exam service.

Table 366. Methods Called by Exam Service

Method	Name as Defined in Service
"Delete" on page 105	ExamDelete
"Execute" on page 108	ExamExecute
"Insert" on page 112	ExamInsert
"QueryPage" on page 115	ExamQueryPage
"Update" on page 136	ExamUpdate

Fields

Table 367 details the required and read-only fields for the exam object.

Table 367. Required and Read-Only Fields for the Exam Object

Field Name	Type
Name	Required
Status	Required
OwnerId	Required
Audit Fields	Read-only

Table 368 details the pick map fields for the exam object.

Table 368. Pick Map Fields for the Exam Object

Pick Map Field	Maps To
PrimaryProductProductCategory, PrimaryProductExternalSystemId, PrimaryProductDescription, PrimaryProductName, PrimaryProductPartNumber, PrimaryProductStatus, PrimaryProductIntegrationId, PrimaryProductType	PrimaryProductId
ProductCategoryName	ProductCategoryId
ContactEmail, ContactFirstName, ContactLastName, ContactFullName, ContactIntegrationId, ContactAccountName, ContactExternalSystemId	ContactId
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId
OwnerFirstName, OwnerLastName, OwnerUserSignInId, OwnerAlias, OwnerFullName, OwnerIntegrationId, OwnerExternalSystemId, OwnerEMailAddr	OwnerId

Table 369 provides a list of the user key combinations for the exam object.

Table 369. User Key Fields on the Exam Object

User Key Field Combinations
Id
Name

Table 370 details the picklists available for the exam object.

Table 370. Picklists Available for the Exam Object

Field Name
Type
Status
Format
Medium
PaymentOption
Grade

Exam Registration

The exam registration object stores information about an individual's registration for an examination.

Child Objects

None.

Methods Called

Table 371 details the methods called by the Exam Registration service.

Table 371. Methods Called by Exam Registration Service

Method	Name as Defined in Service
"Delete" on page 105	ExamRegistrationDelete
"Execute" on page 108	ExamRegistrationExecute
"Insert" on page 112	ExamRegistrationInsert

Table 371. Methods Called by Exam Registration Service

Method	Name as Defined in Service
"QueryPage" on page 115	ExamRegistrationQueryPage
"Update" on page 136	ExamRegistrationUpdate

Fields

Table 372 details the required and read-only fields for the exam registration object.

Table 372. Required and Read-Only Fields for the Exam Registration Object

Field Name	Type
ExamId	Required
CandidateId	Required
ExamDate	Required
OwnerId	Required
Audit Fields	Read-only

Table 373 details the pick map fields for the exam registration object.

Table 373. Pick Map Fields for the Exam Registration Object

Pick Map Field	Maps To
ExamName, ExamStatus, ExamType, ExamID, ExamExternalSystemId	ExamId
ContactEmail, ContactFirstName, ContactLastName, ContactFullName, ContactIntegrationId, ContactAccountName, ContactExternalSystemId	ContactId
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId
PartnerChannelAccountManagerAlias, PartnerIntegrationId, PartnerExternalSystemId, PartnerLocation, PartnerMainPhone, PartnerName,	PartnerId
OwnerFirstName, OwnerLastName, OwnerUserSignInId, OwnerAlias, OwnerFullName, OwnerIntegrationId, OwnerExternalSystemId, OwnerEMailAddr.	OwnerId

Table 374 provides a list of the user key combinations for the exam registration object.

Table 374. User Key Fields on the Exam Registration Object

User Key Field Combinations
Id
CourseId, CandidateId, OwnerId, and ExamDate

Table 375 details the picklists available for the exam registration object.

Table 375. Picklists Available for the Exam Registration Object

Field Name
Status

Fund

The fund object stores information about a sum of money or a set of other resources that are set aside for a specific purpose.

Child Objects

Activity, Custom Object 01-03, Custom Object 4 through 15, MDF Request, Special Pricing Request.

Methods Called

Table 376 details the methods called by the Fund service.

Table 376. Methods Called by Fund Service

Method	Name as Defined in Service
“Delete” on page 105	FundDelete
“Execute” on page 108	FundExecute
“Insert” on page 112	FundInsert
“QueryPage” on page 115	FundQueryPage
“Update” on page 136	FundUpdate

Fields

Table 377 details the required and read-only fields for the fund object.

Table 377. Required and Read-Only Fields for the Fund Object

Field Name	Type
FundName	Required
FundId	Required
TargetAmount	Required
Audit Fields	Read-only

Table 378 details the pick map field for the fund object.

Table 378. Pick Map Field for the Fund Object

Pick Map Field	Maps To
Partner	PartnerId

Table 379 provides a list of the user key combinations for the fund object.

Table 379. User Key Fields on the Fund Object

User Key Field Combinations
FundId
Id

Table 380 details the picklists available for the fund object.

Table 380. Picklists Available for the Fund Object

Field Name
FundCode
Status
Type
CurrencyCode
Partner
Period

MDF Request

The MDF request object stores information about market development funds (MDF) requests.

Child Objects

Activity, Book, Custom Object 01-03, Custom Object 4 through 15.

Methods Called

Table 381 details the methods called by the MDF Request service.

Table 381. Methods Called by MDF Request Service

Method	Name as Defined in Service
"Delete" on page 105	MDFRequestDelete
"Execute" on page 108	MDFRequestExecute
"Insert" on page 112	MDFRequestInsert
"QueryPage" on page 115	MDFRequestQueryPage
"Update" on page 136	MDFRequestUpdate

Fields

Table 382 details the required and read-only fields for the MDF request object.

Table 382. Read-Only Fields on the MDF Request Object

Field Name	Type
RequestName	Required
PrincipalPartnerAccountId	Required
FundName	Required
DueDate	Required
SubmissionStatus	Required
MDFRequestId	Read-only
Audit Fields	Read-only

Table 383 details the pickmap fields for the MDF request object.

Table 383. Pick Map Fields for the MDF Request Object

Pick Map Field	Maps To
PrincipalPartnerAccountExternalSystemId	PrincipalPartnerAccountId
PrincipalPartnerAccountIntegrationId	PrincipalPartnerAccountId
PrincipalPartnerAccountPartnerName, PrincipalPartnerAccountLocation	PrincipalPartnerAccountId
PrincipalPartnerAccountChannelAccountManagerAlias	PrincipalPartnerAccountId

Table 384 provides a list of the user key combinations for the MDF request object.

Table 384. User Key Fields on the MDF Request Object

User Key Field Combinations
RequestName
Id
ExternalSystemId

Table 385 details the picklists available for the MDF request object.

Table 385. Picklists Available for the MDF Request Object

Field Name
Status
Type
SubmissionStatus
ApprovalStatus
MarketingPurpose
Region
Category
ReasonCode
ClaimStatus

Partner

The partner object allows you to define and record details about partner accounts.

Child Objects

Application, Book, Business Plan, Custom Object 4 through 15, Deal Registration, MDF Request, Partner, Policy, Special Pricing Request.

Methods Called

Table 386 details the methods called by the Partner service.

Table 386. Methods Called by Partner Service

Method	Name as Defined in Service
“Insert” on page 112	PartnerInsert
“QueryPage” on page 115	PartnerQueryPage
“Update” on page 136	PartnerUpdate

Fields

Table 387 details the read-only fields for the partner object.

Table 387. Read-Only Fields on the Partner Object

Field Name	Type
PartnerName	Required
CurrencyCode	Read-only
Audit Fields	Read-only

Table 388 details the pickmap fields for the partner object.

Table 388. Pick Map Fields for the Partner Object

Pick Map Field	Maps To
Owner	OwnerId
ParentAccountName, ParentAccountLocation	ParentAccountId
ChannelAccountManager	ChannelAccountManagerId

Table 388. Pick Map Fields for the Partner Object

Pick Map Field	Maps To
SourceCampaignName	SourceCampaignId
OwnerPartnerAccountPartnerName	OwnerPartnerId
PrincipalPartnerPartnerName	PrincipalPartnerId
OriginatingPartnerAccountPartnerName	OriginatingPartnerId
Territory	TerritoryId

Table 389 provides a list of the user key combinations for the partner object.

Table 389. User Key Fields on the Partner Object

User Key Field Combinations
Id
ExternalSystemId
Location
PartnerName

Table 390 details the picklists available for the partner object.

Table 390. Picklists Available for the Partner Object

Field Name
CallFrequency
ComplianceStatus
Expertise
InfluenceType
MarketPotential
MarketSegment
PartnerOrgStatus
PartnerLevel
Priority
Region

Table 390. Picklists Available for the Partner Object

Field Name
Route
Status

Price List

The price list object store information about a price list for products.

Child Objects

Account, Partner, Price List Line Item, Special Pricing Request.

Methods Called

Table 391 details the methods called by the Price List service.

Table 391. Methods Called by Price List Service

Method	Name as Defined in Service
“Delete” on page 105	PriceListDelete
“Execute” on page 108	PriceListExecute
“Insert” on page 112	PriceListInsert
“QueryPage” on page 115	PriceListQueryPage
“Update” on page 136	PriceListUpdate

Fields

Table 392 details the required and read-only fields for the price list object.

Table 392. Required and Read-Only Fields for the Price List Object

Field Name	Type
PriceListName	Required
EffectiveFrom	Required
Type	Required
Status	Required

Table 392. Required and Read-Only Fields for the Price List Object

Field Name	Type
CurrencyCode	Required
Audit Fields	Read-only

There are no pick map fields for the price list object.

Table 393 details the user keys for the price list object.

Table 393. User Keys for the Price List Object

Field Name
PriceListName and Type
Id
ExternalSystemId

Table 394 details the picklists available for the price list object.

Table 394. Picklists Available for the Price List Object

Field Name
Type
Status

Price List Line Item

The price list line item object store information about a product in a price list.

Child Objects

None

Methods Called

Table 395 details the methods called by the Price List Line Item service.

Table 395. Methods Called by Price List Line Item Service

Method	Name as Defined in Service
"Delete" on page 105	PriceListLineItemDelete
"Execute" on page 108	PriceListLineItemExecute
"Insert" on page 112	PriceListLineItemInsert
"QueryPage" on page 115	PriceListLineItemQueryPage
"Update" on page 136	PriceListLineItemUpdate

Fields

Table 396 details the required and read-only fields for the price list line item object.

Table 396. Required and Read-Only Fields for the Price List Line Item Object

Field Name	Type
PriceListId	Required
ProductId	Required
Type	Required
Audit Fields	Read-only

Table 397 details the pick map field for the price list line item object.

Table 397. Pick Map Field for the Price List Line Item Object

Pick Map Field	Maps To
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId
ProductName, ProductCategory	ProductId
PriceListPriceListName	PriceListId
PriceListIntegrationId	PriceListId
PriceListExternalSystemId	PriceListId

Table 398 details the user keys for the price list line item object.

Table 398. User Keys for the Price List Line Item Object

Field Name
Id
ExternalSystemId
ProductId and PriceListId and PriceType

Table 399 details the picklists available for the price list line item object.

Table 399. Picklists Available for the Price List Line Item Object

Field Name
PriceType

Special Pricing Request

The Special Pricing (SP) request object stores information about a special pricing request, which is a set of business processes where a partner asks the brand owner for a discount on a product.

Child Objects

Activity, Book, Custom Object 01 - 03, Custom Object 4 through 15, Special Pricing Request Line Item.

Methods Called

Table 400 details the methods called by the SP Request service.

Table 400. Methods Called by SP Request Service

Method	Name as Defined in Service
"Delete" on page 105	SPRequestDelete
"Execute" on page 108	SPRequestExecute
"Insert" on page 112	SPRequestInsert
"QueryPage" on page 115	SPRequestQueryPage
"Update" on page 136	SPRequestUpdate

Fields

Table 401 details the required and read-only fields for the SP request object.

Table 401. Read-Only Fields on the SP Request Object

Field Name	Type
SPRequestName	Required
PrincipalPartnerAccountId	Required
SubmissionStatus	Required
SPRequestId	Read-only
Audit Fields	Read-only

Table 402 details the pickmap fields for the SP request object.

Table 402. Pick Map Fields for the SP Request Object

Pick Map Field	Maps To
PrincipalPartnerAccountExternalSystemId	PrincipalPartnerAccountId
PrincipalPartnerAccountIntegrationId	PrincipalPartnerAccountId
PrincipalPartnerAccountPartnerName, PrincipalPartnerAccountLocation	PrincipalPartnerAccountId
PrincipalPartnerAccountChannelAccountManagerAlias	PrincipalPartnerAccountId

Table 403 provides a list of the user key combinations for the SP request object.

Table 403. User Key Fields on the SP Request Object

User Key Field Combinations
Id
ExternalSystemId
SPRequestName
FundId and RequestDate

Table 404 details the picklists available for the SP request object.

Table 404. Picklists Available for the SP Request Object

Field Name
SubmissionStatus
ApprovalStatus
RejectReason
Type
ClaimStatus

Special Pricing Request Line Item

The Special Pricing (SP) request line item object stores information about a product associated with a special pricing request.

Child Objects

None.

Methods Called

Table 405 details the methods called by the SP Request Line Item service.

Table 405. Methods Called by SP Request Line Item Service

Method	Name as Defined in Service
"Delete" on page 105	SPRequestLineItemDelete
"Execute" on page 108	SPRequestLineItemExecute
"Insert" on page 112	SPRequestLineItemInsert
"QueryPage" on page 115	SPRequestLineItemQueryPage
"Update" on page 136	SPRequestLineItemUpdate

Fields

Table 406 details the required and read-only fields for the SP request line item object.

Table 406. Read-Only Fields on the SP Request Line Item Object

Field Name	Type
ProductId	Required
SPRequestId	Read-only
Audit Fields	Read-only

Table 407 details the pickmap fields for the SP request line item object.

Table 407. Pick Map Fields for the SP Request Line Item Object

Pick Map Field	Maps To
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId
ProductName, ProductCategory	ProductId

Table 408 provides a list of the user key combinations for the SP request line item object.

Table 408. User Key Fields on the SP Request Line Item Object

User Key Field Combinations
Id
ExternalSystemId
ItemNumber

There are no picklists available for the SP request line item object.

Life Sciences Edition Parent Objects (Web Services v2.0)

This topic includes the parent objects available with Oracle CRM On Demand Life Sciences Edition. For these parent objects, all child objects are also accessible.

NOTE: To download WSDL files for these objects, you must be given access to the object. If you do not have access to the object, it is not available to download from the Web Services Administration page or available to use the Web service calls. For assistance in gaining access to the objects, contact your Oracle CRM On Demand service provider.

The following objects are detailed in this topic:

- "Transaction Item" on page 509

NOTE: While the Business Plan, Objective, Plan Account, Plan Contact, and Plan Opportunity functionality was developed in the context of Oracle CRM On Demand Life Sciences Edition, all Oracle CRM On Demand customers can access and leverage this functionality. Your administrator or customer advocate can enable the Business Plan, Objective, Plan Account, Plan Contact, and Plan Opportunity objects and make these pages available for your use.

Allocation

The allocation object determines the period of time in which sample products are provided for a sales representative to distribute to clinics, hospitals, or physician contacts during a sample drop call execution.

Child Objects

None.

Methods Called

Table 409 details the methods called by the Allocation service.

Table 409. Methods Called by Allocation Service

Method	Name as Defined in Service
"Delete" on page 105	AllocationDelete
"Execute" on page 108	AllocationExecute
"Insert" on page 112	AllocationInsert
"QueryPage" on page 115	AllocationQueryPage
"Update" on page 136	AllocationUpdate

Fields

Table 410 details the required and read-only fields for the allocation object.

Table 410. Required and Read-Only Fields for the Allocation Object

Field Name	Type
AllocationType	Required
OwnerId	Required
ProductName	Required
StopSampleFlag	Required

Table 410. Required and Read-Only Fields for the Allocation Object

Field Name	Type
ActivePeriod	Read-only
Audit Fields	Read-only

Table 411 details the pick map fields for the allocation object.

Table 411. Pick Map Fields for the Allocation Object

Pick Map Field	Maps To
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 412 provides a list of the user key combinations for the allocation object.

Table 412. User Key Fields on the Allocation Object

User Key Field Combinations
Id
ExternalSystemId
OwnerId and ProductId and StartDate and EndDate and AllocationType

Table 413 details the picklists available for the allocation object.

Table 413. Picklists Available for the Allocation Object

Field Name
AllocationType

Business Plan

The business plan object allows sales organizations to explicitly establish strategic goals (for example, revenue targets) and action plans to achieve those goals, in the form of objectives, activities, and opportunities.

Child Objects

Activity, Business Plan, Objective, Plan Account, Plan Contact, Plan Opportunity.

Methods Called

Table 414 details the methods called by the Business Plan service.

Table 414. Methods Called by BusinessPlan Service

Method	Name as Defined in Service
“Delete” on page 105	BusinessPlanDelete
“Execute” on page 108	BusinessPlanExecute
“Insert” on page 112	BusinessPlanInsert
“QueryPage” on page 115	BusinessPlanQueryPage
“Update” on page 136	BusinessPlanUpdate

Fields

Table 415 details the required and read-only fields for the business plan object.

Table 415. Required and Read-Only Fields for the Business plan Object

Field Name	Type
PlanName	Required
PeriodName	Required
Type	Required
Status	Required
Audit Fields	Read-only

Table 416 details the pick map fields for the business plan object.

Table 416. Pick Map Fields for the Business plan Object

Pick Map Field	Maps To
ParentPlanNamePlanName	ParentPlanId
ParentPlanNameExternalSystemId	ParentPlanId
ParentPlanNameIntegrationId	ParentPlanId
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId
PeriodName	PeriodId
PeriodExternalSystemId	PeriodId
PeriodIntegrationId	PeriodId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 417 provides a list of the user key combinations for the business plan object.

Table 417. User Key Fields on the Business plan Object

User Key Field Combinations
Id
ExternalSystemId
PeriodId and PlanName

Table 418 details the picklists available for the business plan object.

Table 418. Picklists Available for the Business plan Object

Field Name
Type
Status

Contact Best Time

The contact best time object records and tracks the best time and day of week to call on a contact (by typically, a physician or another health care provider).

Child Objects

None.

Methods Called

Table 419 details the methods called by the Contact Best Time service.

Table 419. Methods Called by Contact Best Time Service

Method	Name as Defined in Service
"Delete" on page 105	ContactBestTimeDelete
"Execute" on page 108	ContactBestTimeExecute
"Insert" on page 112	ContactBestTimeInsert
"QueryPage" on page 115	ContactBestTimeQueryPage
"Update" on page 136	ContactBestTimePlanUpdate

Fields

Table 420 details the required and read-only fields for the contact best time object.

Table 420. Required and Read-Only Fields for the Contact Best Time Object

Field Name	Type
StartDay	Required
StartTime	Required
EndTime	Required
FullAddress	Required
ContactFullName	Required
Audit Fields	Read-only

Table 421 details the pick map fields for the contact best time object.

Table 421. Pick Map Fields for the Contact Best Time Object

Pick Map Field	Maps To
AddressExternalSystemId	AddressId
AddressIntegrationId	AddressId
ContactFullName	ContactId
ContactExternalSystemId	ContactId
ContactIntegrationId	ContactId

Table 422 provides a list of the user key combinations for the contact best time object.

Table 422. User Key Fields on the Contact Best Time Object

User Key Field Combinations
ContactId
AddressId
ExternalSystemId

Table 423 details the picklists available for the contact best time object.

Table 423. Picklists Available for the Contact Best Time Object

Field Name
StartDay

Contact State License

The contact state license object stores the medical license information for a contact, typically a physician.

Child Objects

None.

Methods Called

Table 424 details the methods called by the contact state license service.

Table 424. Methods Called by Contact State License Service

Method	Name as Defined in Service
"Delete" on page 105	ContactLicenseDelete
"Execute" on page 108	ContactLicenseExecute
"Insert" on page 112	ContactLicenseInsert
"QueryPage" on page 115	ContactLicenseQueryPage
"Update" on page 136	ContactLicenseUpdate

Fields

Table 425 details the required and read-only fields for the contact state license object.

Table 425. Required and Read-Only Fields for the Contact State LicenseObject

Field Name	Type
ContactFullName	Required
LicenseNumber	Required
Audit Fields	Read-only

Table 426 details the pick map fields for the contact state license object.

Table 426. Pick Map Fields for the Contact State License Object

Pick Map Field	Maps To
ContactFullName	ContactId
ContactExternalSystemId	ContactId
ContactIntegrationId	ContactId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 427 provides a list of the user key combinations for the contact state license object.

Table 427. User Key Fields on the Contact State License Object

User Key Field Combinations
Id
ExternalSystemId
ContactId and State
LicenseNumber

Table 428 details the picklists available for the contact state license object.

Table 428. Picklists Available for the Contact State License Object

Field Name
State
Status

Inventory Audit Report

The inventory audit report object stores information about the reports generated by Oracle CRM On Demand when a sales representative chooses to create a new interim or final audit count for an inventory period.

Child Objects

Sample Inventory

Methods Called

Table 429 details the methods called by the Inventory Audit Report service.

Table 429. Methods Called by Inventory Audit Report Service

Method	Name as Defined in Service
"Delete" on page 105	InventoryAuditReportDelete
"Execute" on page 108	InventoryAuditReportExecute
"Insert" on page 112	InventoryAuditReportInsert

Table 429. Methods Called by Inventory Audit Report Service

Method	Name as Defined in Service
"QueryPage" on page 115	InventoryPeriodQueryPage
"Update" on page 136	InventoryPeriodUpdate

Fields

Table 430 details the required and read-only fields for the inventory audit report object.

Table 430. Required and Read-Only Fields for the Inventory Audit Report Object

Field Name	Type
Type	Required
Status	Required
Reason	Required
CountCompletedOn	Required
InventoryPeriodStartDate	Required
OwnerUserSignId	Required
Audit Fields	Read-only

Table 431 details the pick map fields for the inventory audit report object.

Table 431. Pick Map Field for the Inventory Audit Report Object

Pick Map Field	Maps To
InventoryPeriodExternalSystemId	InventoryPeriodId
InventoryPeriodIdIntegrationId	InventoryPeriodId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 432 provides a list of the user key combinations for the inventory audit report object.

Table 432. User Key Fields on the Inventory Audit Report Object

User Key Field Combinations
Id
ExternalSystemId
InventoryPeriodId and OwnerId and CountCompletedOn
Type

Table 433 details the picklists available for the inventory audit report object.

Table 433. Picklists Available for the Inventory Audit Report Object

Field Name
Type
Status
Reason

Inventory Period

The inventory period object represents a time frame within which all movement of samples or promotional items (sample drops, receipts, disbursements, and inventory adjustments) can be tracked.

Child Objects

Inventory Audit Report, Sample Inventory, Sample Transaction, Sample Transaction.

Methods Called

Table 434 details the methods called by the Inventory Period service.

Table 434. Methods Called by Inventory Period Service

Method	Name as Defined in Service
“Delete” on page 105	InventoryPeriodDelete
“Execute” on page 108	InventoryPeriodExecute
“Insert” on page 112	InventoryPeriodInsert

Table 434. Methods Called by Inventory Period Service

Method	Name as Defined in Service
“QueryPage” on page 115	InventoryPeriodQueryPage
“Update” on page 136	InventoryPeriodUpdate

Fields

Table 435 details the required and read-only fields for the inventory period object.

Table 435. Required and Read-Only Fields for the Inventory Period Object

Field Name	Type
StartDate	Required
OwnerId	Required
ActiveFlg	Required
ReconciledFlg	Required
Audit Fields	Read-only

Table 436 details the pick map fields for the inventory period object.

Table 436. Pick Map Field for the Inventory Period Object

Pick Map Field	Maps To
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 437 provides a list of the user key combinations for the inventory period object.

Table 437. User Key Fields on the Inventory Period Object

User Key Field Combinations
Id
ExternalSystemId
OwnerId and StartDate

MedEd

The MedEd object allows you to plan and track medical education events. A medical education event can be as simple as a lunch-and-learn session in a physician's office or as complex as a seminar series or national sales meeting.

Child Objects

Book, Invitee

Methods Called

Table 438 details the methods called by the MedEd service.

Table 438. Methods Called by MedEd Service

Method	Name as Defined in Service
"Delete" on page 105	MedEdDelete
"Execute" on page 108	MedEdExecute
"Insert" on page 112	MedEdInsert
"QueryPage" on page 115	MedEdQueryPage
"Update" on page 136	MedEdUpdate

Fields

Table 439 details the required and read-only fields for the MedEd object.

Table 439. Read-Only Fields on the MedEd Object

Field Name	Type
EndDate	Required
Name	Required
Objective	Required
StartDate	Required
Audit Fields	Read-only

Table 440 details the pickmap fields for the MedEd object.

Table 440. Pick Map Fields for the MedEd Object

Pick Map Field	Maps To
ProductExternalId	ProductId
ProductIntegrationId	ProductId

Table 441 provides a list of the user key combinations for the MedEd object.

Table 441. User Key Fields on the MedEd Object

User Key Field Combinations
MedEdId
ExternalSystemId
Name

Table 442 details the picklists available for the MedEd object.

Table 442. Picklists Available for the MedEd Object

Field Name
EventStatusCode
EventTypeCode

Message Response

The message response object stores feedback received from the audience during the presentation of a message plan during a particular period.

Child Objects

None

Methods Called

Table 443 details the methods called by the Message Response service.

Table 443. Methods Called by Message Response Service

Method	Name as Defined in Service
"Delete" on page 105	MessageResponseDelete
"Execute" on page 108	MessageResponseExecute
"Insert" on page 112	MessageResponseInsert
"QueryPage" on page 115	MessageResponseQueryPage
"Update" on page 136	MessageResponseUpdate

Fields

Table 444 details the required and read-only fields for the message response object.

Table 444. Read-Only Fields on the Message Response Object

Field Name	Type
EndTime	Required
FollowUp	Required
Response	Required
SequenceNumber	Required
SolutionId	Required
StartTime	Required
Message	Read-only
Duration	Read-only
Audit Fields	Read-only

Table 445 details the pickmap fields for the message response object.

Table 445. Pick Map Fields for the Message Response Object

Pick Map Field	Maps To
ContactFullName	ContactId
ContactExternalSystemId	ContactId
ContactIntegrationId	ContactId

Table 445. Pick Map Fields for the Message Response Object

Pick Map Field	Maps To
AccountName	AccountId
AccountExternalSystemId	AccountId
AccountIntegrationId	AccountId
SolutionTitle	SolutionId
SolutionExternalSystemId	SolutionId
SolutionIntegrationId	SolutionId
PlanName	PlanId
PlanExternalSystemId	PlanId
PlanIntegrationId	Planid
PlanItemName	PlanItemId
PlanItemExternalSystemId	PlanItemId
PlanItemIntegrationId	PlanItemId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 446 provides a list of the user key combinations for the message response object.

Table 446. User Key Fields on the Message Response Object

User Key Field Combinations
Id
ExternalSystemId
PlanName and SolutionName

Table 447 details the picklists available for the message response object.

Table 447. Picklists Available for the Message Response Object

Field Name
Response

Messaging Plan

The messaging plan object stores information about a presentation delivered using a personal computer or tablet computer for sales users to detail products and track customer feedback.

Child Objects

Book, Message Response, Messaging Plan Item.

Methods Called

Table 448 details the methods called by the Messaging Plan service.

Table 448. Methods Called by Messaging Plan Service

Method	Name as Defined in Service
"Delete" on page 105	MessagePlanDelete
"Execute" on page 108	MessagePlanExecute
"Insert" on page 112	MessagePlanInsert
"QueryPage" on page 115	MessagePlanQueryPage
"Update" on page 136	MessagePlanUpdate

Fields

Table 449 details the required and read-only fields for the messaging plan object.

Table 449. Read-Only Fields on the Messaging Plan Object

Field Name	Type
Name	Required
OwnerId	Required
Type	Required
ProductName	Required
Status	Required
LockSequence	Required
DisclosureMandatory	Required
EnableFollowUp	Required
VerificationStatus	Read-only
Audit Fields	Read-only

Table 450 details the pickmap fields for the messaging plan object.

Table 450. Pick Map Fields for the Messaging Plan Object

Pick Map Field	Maps To
SolutionTitle	SolutionId
SolutionExternalSystemId	SolutionId
SolutionIntegrationId	SolutionId
EffectivePeriodName	EffectivePeriodId
EffectivePeriodExternalSystemId	EffectivePeriodId
EffectivePeriodIntegrationId	EffectivePeriodId
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 451 provides a list of the user key combinations for the messaging plan object.

Table 451. User Key Fields on the Messaging Plan Object

User Key Field Combinations
Id
ExternalSystemId
Name and OwnerId

Table 452 details the picklists available for the messaging plan object.

Table 452. Picklists Available for the Messaging Plan Object

Field Name
Type
Status
Skin

Messaging Plan Item

The messaging plan item object stores information about the items that make up a message plan.

Child Objects

Messaging Plan Item Relation, Message Response.

Methods Called

Table 453 details the methods called by the Messaging Plan Item service.

Table 453. Methods Called by Messaging Plan Item Service

Method	Name as Defined in Service
"Delete" on page 105	MsgPlanItemDelete
"Execute" on page 108	MsgPlanItemExecute
"Insert" on page 112	MsgPlanItemInsert
"QueryPage" on page 115	MsgPlanItemQueryPage
"Update" on page 136	MsgPlanItemUpdate

Fields

Table 454 details the required and read-only fields for the messaging plan item object.

Table 454. Read-Only Fields on the Messaging Plan Item Object

Field Name	Type
SequenceNumber	Required
DisclosureMessage	Required
Type	Required
SolutionNameTitle	Required
ParentMessagePlanName	Required
Audit Fields	Read-only

Table 455 details the pickmap fields for the messaging plan item object.

Table 455. Pick Map Fields for the Messaging Plan Item Object

Pick Map Field	Maps To
ParentMessagePlanName	ParentMessagePlanId
ParentMessagePlanExternalSystemId	ParentMessagePlanId
ParentMessagePlanIntegrationId	ParentMessagePlanId
SolutionNameTitle	SolutionId
SolutionExternalSystemId	SolutionId
SolutionIntegrationId	SolutionId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 456 provides a list of the user key combinations for the messaging plan item object.

Table 456. User Key Fields on the Messaging Plan Item Object

User Key Field Combinations
Id
ExternalSystemId
ParentMessagePlanId and SolutionNameId and SequenceNumber and Type

Table 457 details the picklists available for the messaging plan item object.

Table 457. Picklists Available for the Messaging Plan Item Object

Field Name
Type
Status
Rating

Messaging Plan Item Relation

The messaging plan item relation object stores information about a presentation item that provides optional information to support the primary message plan item.

Child Objects

None

Methods Called

Table 458 details the methods called by the Messaging Plan Item Relation service.

Table 458. Methods Called by Messaging Plan Item Service

Method	Name as Defined in Service
"Delete" on page 105	MsgPlanItemRelationDelete
"Execute" on page 108	MsgPlanItemRelationExecute
"Insert" on page 112	MsgPlanItemRelationInsert
"QueryPage" on page 115	MsgPlanItemRelationQueryPage
"Update" on page 136	MsgPlanItemRelationUpdate

Fields

Table 459 details the required and read-only fields for the messaging plan item relation object.

Table 459. Read-Only Fields on the Messaging Plan Item Relation Object

Field Name	Type
Type	Required
ParentMPItemName	Required
SolutionNameTitle	Required
Audit Fields	Read-only

Table 460 details the pickmap fields for the messaging plan item relation object.

Table 460. Pick Map Fields for the Messaging Plan Item Relation Object

Pick Map Field	Maps To
ParentMPItemName	ParentMPItemId
ParentMPItemExternalSystemId	ParentMPItemId
ParentMPItemIntegrationId	ParentMPItemid
SolutionNameTitle	SolutionId
SolutionExternalSystemId	SolutionId
SolutionIntegrationId	SolutionId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 461 provides a list of the user key combinations for the messaging plan item relation object.

Table 461. User Key Fields on the Messaging Plan Item Relation Object

User Key Field Combinations
Id
ExternalSystemId
ParentMPId and SolutionNameId and Type

Table 462 details the picklists available for the messaging plan item relation object.

Table 462. Picklists Available for the Messaging Plan Item Relation Object

Field Name
Type
Status
Rating

Modification Tracking

The modification tracking object stores information about change events for a given set of Oracle CRM On Demand objects. When the modification tracking feature is enabled, for each event a modification record is written to the modification tracking log, which is used for synchronization of external mobile applications.

NOTE: The Modification Tracking Web service is for use with the CRM On Demand Offline Client for Life Sciences.

Child Objects

None.

Methods Called

Table 463 details the methods called by the Modification Tracking service.

Table 463. Methods Called by Modification Tracking Service

Method	Name as Defined in Service
"Delete" on page 105	ModificationTrackingDelete
"Execute" on page 108	ModificationTrackingExecute
"Insert" on page 112	ModificationTrackingInsert
"QueryPage" on page 115	ModificationTrackingQueryPage
"Update" on page 136	ModificationTrackingUpdate

Fields

Table 464 details the required and read-only fields for the modification tracking object.

Table 464. Required and Read-Only Fields for the Modification Tracking Object

Field Name	Type
ObjectId	Required
EventName	Required
ObjectName	Required
ModificationNumber	Required
Audit Fields	Read-only

Table 465 provides a list of the user key combinations for the modification tracking object.

Table 465. User Key Fields on the Modification Tracking Object

User Key Field Combinations
Id
ObjectName

Objective

The objective object stores information about a tactical goal you want to achieve to attain a higher level strategic goal in a business plan.

Child Objects

Activity, Objective, Opportunity.

Methods Called

Table 466 details the methods called by the objective service.

Table 466. Methods Called by Objective Service

Method	Name as Defined in Service
"Delete" on page 105	ObjectiveDelete
"Execute" on page 108	ObjectiveExecute
"Insert" on page 112	ObjectiveInsert
"QueryPage" on page 115	ObjectiveQueryPage
"Update" on page 136	ObjectiveUpdate

Fields

Table 467 details the required and read-only fields for the objective object.

Table 467. Required and Read-Only Fields for the Objective Object

Field Name	Type
ObjectiveName	Required
Type	Required
Status	Required

Table 467. Required and Read-Only Fields for the Objective Object

Field Name	Type
PeriodName	Required
Audit Fields	Read-only

Table 468 details the pick map fields for the objective object.

Table 468. Pick Map Fields for the Objective Object

Pick Map Field	Maps To
ParentObjectiveObjectiveName	ParentObjectiveId
ParentObjectiveExternalSystemId	ParentObjectiveId
ParentObjectiveIntegrationId	ParentObjectiveId
PlanNamePlanName	PlanNameId
PlanNameExternalSystemId	PlanNameId
PlanNameIntegrationId	PlanNameId
AccountNameAccountName	AccountId
AccountExternalSystemId	AccountId
AccountIntegrationId	AccountId
ContactNameContactFullName	ContactId
ContactExternalSystemId	ContactId
ContactIntegrationId	ContactId
ProductNameName	ProductId
ProductdExternalSystemId	ProductId
ProductIntegrationId	ProductId
PeriodName	PeriodId
PeriodExternalSystemId	PeriodId
PeriodIntegrationId	PeriodId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 469 provides a list of the user key combinations for the objective object.

Table 469. User Key Fields on the Objective Object

User Key Field Combinations
Id
ExternalSystemId
AccountNameId and ContactNameId and ProductNameId and PeriodId
ObjectiveName

Table 470 details the picklists available for the objective object.

Table 470. Picklists Available for the Objective Object

Field Name
Status
Type
ObjectiveUnits

Plan Account

The plan account object is a mapping of accounts and business plans.

Child Objects

None.

Methods Called

Table 471 details the methods called by the plan account service.

Table 471. Methods Called by Plan Account Service

Method	Name as Defined in Service
"Delete" on page 105	AccountBusinessPlanDelete
"Execute" on page 108	AccountBusinessPlanExecute
"Insert" on page 112	AccountBusinessPlanInsert

Table 471. Methods Called by Plan Account Service

Method	Name as Defined in Service
"QueryPage" on page 115	AccountBusinessPlanQueryPage
"Update" on page 136	AccountBusinessPlanUpdate

Fields

Table 472 details the required and read-only fields for the plan account object.

Table 472. Required and Read-Only Fields for the Plan Account Object

Field Name	Type
AccountIDAccountName	Required
BusinessPlanPlanName	Required
ContactId	Required
RelationshipCode	Read-only
Audit Fields	Read-only

Table 473 details the pick map fields for the plan account object.

Table 473. Pick Map Fields for the Plan Account Object

Pick Map Field	Maps To
AccountIDAccountName	AccountId
AccountExternalSystemId	AccountId
AccountIntegrationId	AccountId
BusinessPlanPlanName	BusinessPlanId
BusinessPlanExternalSystemId	BusinessPlanId
BusinessPlanIntegrationId	BusinessPlanId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 474 provides a list of the user key combinations for the plan account object.

Table 474. User Key Fields on the Plan Account Object

User Key Field Combinations
Id
ExternalSystemId
BusinessPlanId and AccountIDId
BusinessPlanPlanName

Plan Contact

The plan contact object is a mapping of contacts and business plans.

Child Objects

None

Methods Called

Table 475 details the methods called by the plan contact service.

Table 475. Methods Called by Plan Contact Service

Method	Name as Defined in Service
“Delete” on page 105	ContactBusinessPlanDelete
“Execute” on page 108	ContactBusinessPlanExecute
“Insert” on page 112	ContactBusinessPlanInsert
“QueryPage” on page 115	ContactBusinessPlanQueryPage
“Update” on page 136	ContactBusinessPlanUpdate

Fields

Table 476 details the required and read-only fields for the plan contact object.

Table 476. Required and Read-Only Fields for the Plan Contact Object

Field Name	Type
ContactId	Required
BusinessPlanPlanName	Required
Audit Fields	Read-only

Table 477 details the pick map fields for the plan contact object.

Table 477. Pick Map Fields for the Plan Contact Object

Pick Map Field	Maps To
ContactFullName	ContactId
ContactExternalSystemId	ContactId
ContactIntegrationId	ContactId
BusinessPlanPlanName	BusinessPlanId
BusinessPlanExternalSystemId	BusinessPlanId
BusinessPlanIntegrationId	BusinessPlanId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 478 provides a list of the user key combinations for the plan contact object.

Table 478. User Key Fields on the Plan Contact Object

User Key Field Combinations
Id
ExternalSystemId
BusinessPlanId and ContactId
ContactFullName

Plan Opportunity

The plan opportunity object is a mapping of opportunities and business plans.

Child Objects

None.

Methods Called

Table 479 details the methods called by the Plan Opportunity service.

Table 479. Methods Called by Plan Opportunity Service

Method	Name as Defined in Service
“Delete” on page 105	PlanOpportunityDelete
“Execute” on page 108	PlanOpportunityExecute
“Insert” on page 112	PlanOpportunityInsert
“QueryPage” on page 115	PlanOpportunityQueryPage
“Update” on page 136	PlanOpportunityUpdate

Fields

Table 480 details the required and read-only fields for the plan opportunity object.

Table 480. Required and Read-Only Fields for the Plan Opportunity Object

Field Name	Type
OpportunityName	Required
PlanName	Required
Audit Fields	Read-only

Table 481 details the pick map fields for the plan opportunity object.

Table 481. Pick Map Fields for the Plan Opportunity Object

Pick Map Field	Maps To
OpportunityName	OpportunityId
OpportunityExternalSystemId	OpportunityId
OpportunityIntegrationId	OpportunityId

Table 481. Pick Map Fields for the Plan Opportunity Object

Pick Map Field	Maps To
PlanName	PlanId
PlanExternalSystemId	PlanId
PlanIntegrationId	PlanId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 482 provides a list of the user key combinations for the plan opportunity object.

Table 482. User Key Fields on the Plan Opportunity Object

User Key Field Combinations
Id
ExternalSystemId
PlanId and OpportunityId
OpportunityName

Sample Disclaimer

The sample disclaimer object stores the text that is displayed to deny responsibility for certain adverse effects and reduce liability for a product sample that is dropped off to a physician or other health care professional.

Child Objects

None

Methods Called

Table 483 details the methods called by the Sample Disclaimer service.

Table 483. Methods Called by Sample Disclaimer Service

Method	Name as Defined in Service
"Delete" on page 105	SignatureDisclaimerDelete
"Execute" on page 108	SignatureDisclaimerExecute

Table 483. Methods Called by Sample Disclaimer Service

Method	Name as Defined in Service
“Insert” on page 112	SignatureDisclaimerInsert
“QueryPage” on page 115	SignatureDisclaimerQueryPage
“Update” on page 136	SignatureDisclaimerUpdate

Fields

Table 484 details the required and read-only fields for the sample disclaimer object.

Table 484. Required and Read-Only Fields for the Sample Disclaimer Object

Field Name	Type
DisclaimerText	Required
Status	Required
Audit Fields	Read-only

Table 485 details the pick map field for the sample disclaimer object.

Table 485. Pick Map Field for the Sample Disclaimer Object

Pick Map Field	Maps To
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 486 details the user keys for the sample disclaimer object.

Table 486. User Keys for the Sample Disclaimer Object

Field Name
Id
ExternalSystemId
DisclaimerNumber
Status and Created

Table 487 details the picklists available for the sample disclaimer object.

Table 487. Picklists Available for the Sample Disclaimer Object

Field Name
Status

Sample Inventory

The sample inventory object stores information about a product that has been categorized as qualifying for dropping a sample.

Child Objects

None

Methods Called

Table 488 details the methods called by the Sample Inventory service.

Table 488. Methods Called by Sample Inventory Service

Method	Name as Defined in Service
“Delete” on page 105	SampleInventoryDelete
“Execute” on page 108	SampleInventoryExecute
“Insert” on page 112	SampleInventoryInsert
“QueryPage” on page 115	SampleInventoryQueryPage
“Update” on page 136	SampleInventoryUpdate

Fields

Table 489 details the required and read-only fields for the sample inventory object.

Table 489. Required and Read-Only Fields for the Sample Inventory Object

Field Name	Type
InventoryPeriodStartDate	Required
ProductName	Required
OpeningBalance	Required

Table 489. Required and Read-Only Fields for the Sample Inventory Object

Field Name	Type
SystemCount	Read-only
Difference	Read-only
InventoryLot	Read-only
Audit Fields	Read-only

Table 490 details the pick map field for the sample inventory object.

Table 490. Pick Map Fields for the Sample Inventory Object

Pick Map Field	Maps To
LotNumLotId	LotNumId
LotNumExternalSystemId	LotNumId
LotNumIntegrationId	LotNumId
InventoryPeriodExternalSystemId	InventoryPeriodId
InventoryPeriodIntegrationId	InventoryPeriodId
AuditReportExternalSystemId	AuditReportId
AuditReportIntegrationId	AuditReportId
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 491 details the user keys for the sample inventory object.

Table 491. User Keys for the Sample Inventory Object

Field Name
SampleInventoryId
IntegrationId
ExternalSystemId

Table 491. User Keys for the Sample Inventory Object

Field Name
OwnerId and ProductId and InventoryPeriodId and LotNumId and AuditReportId
ProductName

Sample Lot

The sample lot object records information about lot numbers for sample products.

Child Objects

Sample Inventory, Transaction Item.

Methods Called

Table 492 details the methods called by the Sample Lot service.

Table 492. Methods Called by Sample Lot Service

Method	Name as Defined in Service
"Delete" on page 105	SampleLotDelete
"Execute" on page 108	SampleLotExecute
"Insert" on page 112	SampleLotInsert
"QueryPage" on page 115	SampleLotQueryPage
"Update" on page 136	SampleLotUpdate

Fields

Table 493 details the required and read-only fields for the sample lot object.

Table 493. Required and Read-Only Fields for the Sample Lot Object

Field Name	Type
LotId	Required
SampleName	Required
ExpirationDate	Required
Orderable	Required

Table 493. Required and Read-Only Fields for the Sample Lot Object

Field Name	Type
ParentSampleNameName	Required
Status	Read-only
CutoffDate	Read-only
Audit Fields	Read-only

Table 494 details the pick map field for the sample lot object.

Table 494. Pick Map Field for the Sample Lot Object

Pick Map Field	Maps To
ParentSampleNameName	ParentSampleNameId
ParentSampleNameExternalSystemId	ParentSampleNameId
ParentSampleNameIntegrationId	ParentSampleNameId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 495 details the user keys for the sample lot object.

Table 495. User Keys for the Sample Lot Object

Field Name
Id
ExternalSystemId
LotId

Sample Transaction

The sample transaction object is a record for a sample transfer, receipt, disbursement, or inventory adjustment. A sample transaction is created so that all movement of samples and promotional items can be reported and reconciled at the end of an inventory period, and a running count of inventory products can be maintained during the life of an inventory period.

Child Objects

Transaction Item

Methods Called

Table 496 details the methods called by the Sample Transaction service.

Table 496. Methods Called by Sample Transaction Service

Method	Name as Defined in Service
“Delete” on page 105	TransactionDelete
“Execute” on page 108	TransactionExecute
“Insert” on page 112	TransactionInsert
“QueryPage” on page 115	TransactionQueryPage
“Update” on page 136	TransactionUpdate

Fields

Table 497 details the required and read-only fields for the sample transaction object.

Table 497. Required and Read-Only Fields for the Sample Transaction Object

Field Name	Type
Name	Required
TransactionDate	Required
TransactionType	Required
InvoiceNumber	Required
Audit Fields	Read-only

Table 498 details the pick map field for the sample transaction object.

Table 498. Pick Map Field for the Sample Transaction Object

Pick Map Field	Maps To
SalesPersonIDId	SalesPersonIDId
SalesPersonIDExternalSystemId	SalesPersonIDId
SalesPersonIDIntegrationId	SalesPersonIDId
TransferToUserSignId	TransferToId

Table 498. Pick Map Field for the Sample Transaction Object

Pick Map Field	Maps To
TransferToExternalSystemId	TransferToId
TransferToIntegrationId	TransferToId
InventoryPeriodExternalSystemId	InventoryPeriodId
InventoryPeriodIntegrationId	InventoryPeriodId
ProductIntegrationId	ProductId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 499 details the user keys for the sample transaction object.

Table 499. User Keys for the Sample Transaction Object

Field Name
Id
ExternalSystemId
Name

Table 500 details the picklists available for the sample transaction object.

Table 500. Picklists Available for the Sample Transaction Object

Field Name
TransactionType
TransactionStatus
Reason

Signature

The signature object stores information about a signature associated with a call activity for samples that are dropped.

Child Objects

None

Methods Called

Table 501 details the methods called by the Signature service.

Table 501. Methods Called by Signature Service

Method	Name as Defined in Service
“Delete” on page 105	SignatureDelete
“Execute” on page 108	SignatureExecute
“Insert” on page 112	SignatureItemInsert
“QueryPage” on page 115	SignatureQueryPage
“Update” on page 136	SignatureUpdate

Fields

Table 502 details the required and read-only fields for the signature object.

Table 502. Required and Read-Only Fields for the Signature Object

Field Name	Type
ActivityId	Required
SignatureCtrl	Required
Audit Fields	Read-only

Table 503 details the pick map field for the signature object.

Table 503. Pick Map Field for the Signature Object

Pick Map Field	Maps To
ContactFullName	ContactId
ContactExternalSystemId	ContactId
ContactIntegrationId	ContactId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 504 details the user keys for the signature object.

Table 504. User Keys for the Signature Object

Field Name
Id
ExternalSystemId
ActivityId

Transaction Item

The transaction item object stores the details of the sample or promotional items associated with a sample transaction.

Child Objects

None

Methods Called

Table 505 details the methods called by the Transaction Item service.

Table 505. Methods Called by Transaction Item Service

Method	Name as Defined in Service
"Delete" on page 105	TransactionItemDelete
"Execute" on page 108	TransactionItemExecute
"Insert" on page 112	TransactionItemInsert
"QueryPage" on page 115	TransactionItemQueryPage
"Update" on page 136	TransactionItemUpdate

Fields

Table 506 details the required and read-only fields for the transaction item object.

Table 506. Required and Read-Only Fields for the Transaction Item Object

Field Name	Type
LineNumber	Required
SampleTransactionName	Required
ProductName	Required
Quantity	Required
Audit Fields	Read-only

Table 507 details the pick map field for the transaction item object.

Table 507. Pick Map Field for the Transaction Item Object

Pick Map Field	Maps To
SampleTransactionName	SampleTransactionId
SampleTransactionExternalSystemId	SampleTransactionId
SampleTransactionIntegrationId	SampleTransactionId
LotNumLotId	LotNumId
LotNumExternalSystemId	LotNumId
LotNumIntegrationId	LotNumId
InventoryPeriodExternalSystemId	InventoryPeriodId
InventoryPeriodIntegrationId	InventoryPeriodId
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 508 details the user keys for the transaction item object.

Table 508. User Keys for the Transaction Item Object

Field Name
Id
ExternalSystemId
SampleTransactionId and LineNumber
ProductName

Financial Services Edition Parent Objects (Web Services v2.0)

This topic includes the parent objects available with Oracle CRM On Demand Financial Services Edition. For these parent objects, all child objects are also accessible.

NOTE: To download WSDL files for these objects, you must be given access to the object. If you do not have access to the object, it is not available to download from the Web Services Administration page or available to use the Web service calls. For assistance in gaining access to the objects, contact your Oracle CRM On Demand service provider.

The following objects are detailed in this topic:

- "Claim" on page 512
- "Coverage" on page 514
- "Damage" on page 516
- "Financial Account" on page 517
- "Financial Account Holder" on page 519
- "Financial Account Holding" on page 520
- "Financial Plan" on page 522
- "Financial Product" on page 524
- "Financial Transaction" on page 525
- "Household" on page 527
- "Insurance Property" on page 529
- "Involved Party" on page 531
- "Policy" on page 532
- "Policy Holder" on page 534
- "Portfolio" on page 536

Claim

The claim object allows you to define and record details about an insurance policy claim. Claims are typically the claims of a contact or business that an insurance company manages.

Child Objects

Activity, Damage, Involved Party, Service Request.

Methods Called

Table 509 details the methods called by the Claim service.

Table 509. Methods Called by Claim Service

Method	Name as Defined in Service
"Delete" on page 105	ClaimDelete
"Execute" on page 108	ClaimExecute
"Insert" on page 112	ClaimInsert
"QueryPage" on page 115	ClaimQueryPage
"Update" on page 136	ClaimUpdate

Fields

Table 510 details the required and read-only fields for the claim object.

Table 510. Read-Only Fields on the Claim Object

Field Name	Type
ClaimNumber	Required
ReportDate	Required
LossDateandTime	Required
PolicyName	Required
CurrencyCode	Read-only
Audit Fields	Read-only

Table 511 details the pickmap fields for the claim object.

Table 511. Pick Map Fields for the Claim Object

Pick Map Field	Maps To
PolicyExternalSystemId	PolicyId
PolicyIntegrationId	PolicyId

Table 512 provides a list of the user key combinations for the claim object.

Table 512. User Key Fields on the Claim Object

User Key Field Combinations
Id
ExternalSystemId
ClaimNumber

Table 513 details the picklists available for the claim object.

Table 513. Picklists Available for the Claim Object

Field Name
AbilitytoWork
AtFault
CategoryofLoss
ClassofEmployee
InjuredDuringWorkingHours
KindofLoss
LiabilitySource
LineofBusiness
LocationofLoss
LossCode
LossType
MedicalInjuryCode
PartofBodyInjured
PlaceofInjury

Table 513. Picklists Available for the Claim Object

Field Name
RelationshiptoInsured
ReportedBy
State
Status
TypeofInjury

Coverage

The coverage object allows you to define and record details about an insurance policy coverage. Coverage is a term used to describe the monetary limits and risks covered as set out in an insurance policy.

Child Objects

None

Methods Called

Table 514 details the methods called by the Coverage service.

Table 514. Methods Called by Coverage Service

Method	Name as Defined in Service
"Delete" on page 105	CoverageDelete
"Execute" on page 108	CoverageExecute
"Insert" on page 112	CoverageInsert
"QueryPage" on page 115	CoverageQueryPage
"Update" on page 136	CoverageUpdate

Fields

Table 515 details the required and read-only fields for the coverage object.

Table 515. Read-Only Fields on the Coverage Object

Field Name	Type
CoverageName	Required
PolicyName	Required
CurrencyCode	Read-only
Owner	Read-only
OwnerId	Read-only
Audit Fields	Read-only

Table 516 details the pickmap fields for the coverage object.

Table 516. Pick Map Fields for the Coverage Object

Pick Map Field	Maps To
PolicyExternalSystemId	PolicyId
PolicyIntegrationId	PolicyId

Table 517 provides a list of the user key combinations for the coverage object.

Table 517. User Key Fields on the Coverage Object

User Key Field Combinations
CoverageName
IntegrationId
ExternalSystemId

Table 518 details the picklists available for the coverage object.

Table 518. Picklists Available for the Coverage Object

Field Name
Coverage
Status

Damage

The damage object allows you to define and record details about the damages to properties covered in an insurance policy.

Child Objects

None

Methods Called

Table 519 details the methods called by the Damage service.

Table 519. Methods Called by Damage Service

Method	Name as Defined in Service
"Delete" on page 105	DamageDelete
"Execute" on page 108	DamageExecute
"Insert" on page 112	DamageInsert
"QueryPage" on page 115	DamageQueryPage
"Update" on page 136	DamageUpdate

Fields

Table 520 details the required and read-only fields for the damage object.

Table 520. Read-Only Fields on the Damage Object

Field Name	Type
ClaimNumber	Required
DamageName	Required
CurrencyCode	Read-only
Owner	Read-only
OwnerId	Read-only
PrimaryContact	Read-only
Audit Fields	Read-only

Table 521 details the pickmap fields for the damage object.

Table 521. Pick Map Fields for the Damage Object

Pick Map Field	Maps To
ClaimExternalId	ClaimId
ClaimIntegrationId	ClaimId

Table 522 provides a list of the user key combinations for the damage object.

Table 522. User Key Fields on the Damage Object

User Key Field Combinations
Id
ExternalSystemId
DamageName

Financial Account

The financial account object stores information about the financial accounts of a contact or business that a financial institution manages, but can also track held away financial accounts.

Child Objects

Activity, Financial Account, Financial Account Holder, Financial Account Holding, Financial Plan, Financial Transaction, Lead, Contact Team, Policy, Service Request.

Methods Called

Table 523 details the methods called by the Financial Account service.

Table 523. Methods Called by Financial Account Service

Method	Name as Defined in Service
"Delete" on page 105	FinancialAccountDelete
"Execute" on page 108	FinancialAccountExecute
"Insert" on page 112	FinancialAccountInsert
"QueryPage" on page 115	FinancialAccountQueryPage
"Update" on page 136	FinancialAccountUpdate

Fields

Table 524 details the required and read-only fields for the financial account object.

Table 524. Required and Read-Only Fields for the Financial Account Object

Field Name	Type
FinancialAccount	Required
FinancialAccountNumber	Required
Type	Required
CurrencyCode	Read-only
Audit Fields	Read-only

Table 525 details the pick map fields for the financial account object.

Table 525. Pick Map Fields for the Financial Account Object

Pick Map Field	Maps To
ParentFinancialAccountIntegrationId	ParentFinancialAccountId
ParentFinancialAccountExternalSystemId	ParentFinancialAccountId

Table 526 provides a list of the user key combinations for the financial account object.

Table 526. User Key Fields on the Financial Account Object

User Key Field Combinations
FinancialAccount
Id
ExternalSystemId

Table 527 details the picklists available for the financial account object.

Table 527. Picklists Available for the Financial Account Object

Field Name
Type
HomeBranch

Financial Account Holder

The financial account holder object stores information about a financial account holder, which is any contact that has a relationship with the financial account, most commonly the legal owner of the financial account.

Child Objects

None

Methods Called

Table 528 details the methods called by the Financial Account Holder service.

Table 528. Methods Called by Financial Account Holder Service

Method	Name as Defined in Service
"Delete" on page 105	FinancialAccountHolderDelete
"Execute" on page 108	FinancialAccountHolderExecute
"Insert" on page 112	FinancialAccountHolderInsert
"QueryPage" on page 115	FinancialAccountHolderQueryPage
"Update" on page 136	FinancialAccountHolderUpdate

Fields

Table 529 details the required and read-only fields for the financial account holder object.

Table 529. Required and Read-Only Fields for the Financial Account Holder Object

Field Name	Type
FinancialAccountHolderName	Required
FinancialAccount	Required
Role	Required
CurrrencyCode	Read-only
Audit Fields	Read-only

Table 530 details the pick map fields for the financial account holder object.

Table 530. Pick Map Fields for the Financial Account Holder Object

Pick Map Field	Maps To
FinancialAccounIntegrationId	FinancialAccountId
FinancialAccounExternalSystemId	FinancialAccountId

Table 531 provides a list of the user key combinations for the financial account holder object.

Table 531. User Key Fields on the Financial Account Holder Object

User Key Field Combinations
FinancialAccountHolderName
Id
ExternalSystemId

Table 532 details the picklists available for the financial account holder object.

Table 532. Picklists Available for the Financial Account Holder Object

Field Name
Role

Financial Account Holding

The financial account holding object stores information about the financial account holdings, which are typically the total of all financial account transactions of a single financial product for a financial account.

Child Objects

None

Methods Called

Table 533 details the methods called by the Financial Account Holding service.

Table 533. Methods Called by Financial Account Holding Service

Method	Name as Defined in Service
"Delete" on page 105	FinancialAccountHoldingDelete
"Execute" on page 108	FinancialAccountHoldingExecute
"Insert" on page 112	FinancialAccountHoldingInsert
"QueryPage" on page 115	FinancialAccountHoldingQueryPage
"Update" on page 136	FinancialAccountHoldingUpdate

Fields

Table 534 details the required and read-only fields for the financial account holding object.

Table 534. Required and Read-Only Fields for the Financial Account Holding Object

Field Name	Type
FinancialAccountHoldingsName	Required
FinancialAccount	Required
FinancialProductFinancialProductName	Required
CurrencyCode	Read-only
Audit Fields	Read-only

Table 535 details the pick map fields for the financial account holding object.

Table 535. Pick Map Fields for the Financial Account Holding Object

Pick Map Field	Maps To
FinancialAccounIntegrationId	FinancialAccountId
FinancialAccounExternalSystemId	FinancialAccountId

Table 536 provides a list of the user key combinations for the financial account holding object.

Table 536. User Key Fields on the Financial Account Holding Object

User Key Field Combinations
FinancialAccountHoldingsName
Id
ExternalSystemId

Financial Plan

The financial plan object stores information about the plans of a contact or business for one or more specific financial accounts that a financial institution manages.

Child Objects

Activity, Lead, Contact Team, Service Request.

Methods Called

Table 537 details the methods called by the Financial Plan service.

Table 537. Methods Called by Financial Plan Service

Method	Name as Defined in Service
"Delete" on page 105	FinancialPlanDelete
"Execute" on page 108	FinancialPlanExecute
"Insert" on page 112	FinancialPlanInsert
"QueryPage" on page 115	FinancialPlanQueryPage
"Update" on page 136	FinancialPlanUpdate

Fields

Table 538 details the required and read-only fields for the financial plan object.

Table 538. Required and Read-Only Fields for the Financial Plan Object

Field Name	Type
FinancialPlan	Required
Status	Required
Type	Required
CurrencyCode	Read-only
Audit Fields	Read-only

Table 539 details the pick map fields for the financial plan object.

Table 539. Pick Map Fields for the Financial Plan Object

Pick Map Field	Maps To
FinancialAccounIntegrationId	FinancialAccountId
FinancialAccounExternalSystemId	FinancialAccountId

Table 540 provides a list of the user key combinations for the financial plan object.

Table 540. User Key Fields on the Financial Plan Object

User Key Field Combinations
FinancialPlan
Id
ExternalSystemId

Table 541 details the picklists available for the financial plan object.

Table 541. Picklists Available for the Financial Plan Object

Field Name
Type
Status

Financial Product

The financial product object stores information about the products and services that a financial institution offers to its customers.

Child Objects

Coverage, Financial Account Holding, Financial Product, Financial Transaction.

Methods Called

Table 542 details the methods called by the Financial Product service.

Table 542. Methods Called by Financial Product Service

Method	Name as Defined in Service
“Delete” on page 105	FinancialProductDelete
“Execute” on page 108	FinancialProductExecute
“Insert” on page 112	FinancialProductInsert
“QueryPage” on page 115	FinancialProductQueryPage
“Update” on page 136	FinancialProductUpdate

Fields

Table 543 details the required and read-only fields for the financial product object.

Table 543. Required and Read-Only Fields for the Financial Product Object

Field Name	Type
FinancialProductName	Required
CurrencyCode	Read-only
Audit Fields	Read-only

Table 544 details the pick map fields for the financial product object.

Table 544. Pick Map Fields for the Financial Product Object

Pick Map Field	Maps To
ParentFinancialProductIntegrationId	ParentFinancialProductId
ParentFinancialProductExternalSystemId	ParentFinancialProductId

Table 545 provides a list of the user key combinations for the financial product object.

Table 545. User Key Fields on the Financial Product Object

User Key Field Combinations
FinancialProductName
Id
ExternalSystemId

Table 546 details the picklists available for the financial product object.

Table 546. Picklists Available for the Financial Product Object

Field Name
Category
Type
Class
Sub-Class

Financial Transaction

The financial transaction object stores information about financial account transactions, which are typically the individual transactions of a financial account for all financial products.

Child Objects

Financial Transaction, Service Request.

Methods Called

Table 547 details the methods called by the Financial Transaction service.

Table 547. Methods Called by Financial Transaction Service

Method	Name as Defined in Service
"Delete" on page 105	FinancialTransactionDelete
"Execute" on page 108	FinancialTransactionExecute
"Insert" on page 112	FinancialTransactionInsert

Table 547. Methods Called by Financial Transaction Service

Method	Name as Defined in Service
“QueryPage” on page 115	FinancialTransactionQueryPage
“Update” on page 136	FinancialTransactionUpdate

Fields

Table 548 details the required and read-only fields for the financial transaction object.

Table 548. Required and Read-Only Fields for the Financial Transaction Object

Field Name	Type
FinancialAccount	Required
FinancialProductFinancialProductName	Required
TransactionType	Required
TransactionID	Required
Audit Fields	Read-only

Table 549 details the pick map fields for the financial transaction object.

Table 549. Pick Map Fields for the Financial Transaction Object

Pick Map Field	Maps To
FinancialTransactionParentIntegrationId	FinancialTransactionParentId
FinancialTransactionParentExternalSystemId	FinancialTransactionParentId

Table 550 provides a list of the user key combinations for the financial transaction object.

Table 550. User Key Fields on the Financial Transaction Object

User Key Field Combinations
TransactionID
Id
ExternalSystemId

Table 551 details the picklists available for the financial transaction object.

Table 551. Picklists Available for the Financial Transaction Object

Field Name
TransactionType

Household

The household object allows you to define and record financial details about a group of contacts that live in the same household, for example, parents, brothers, sisters, spouses, and so on. These details include the assets of the household, the liabilities of the household, the net income of the household, and so on.

Child Objects

Claim, Financial Account, Financial Account Holder, Financial Plan, Policy, Policy Holder.

Methods Called

Table 552 details the methods called by the Household service.

Table 552. Methods Called by Household Service

Method	Name as Defined in Service
“Delete” on page 105	HouseholdDelete
“Execute” on page 108	HouseholdExecute
“Insert” on page 112	HouseholdInsert
“QueryPage” on page 115	HouseholdQueryPage
“Update” on page 136	HouseholdUpdate

Fields

Table 553 details the required and read-only fields for the household object.

Table 553. Required and Read-Only Fields for the Household Object

Field Name	Type
HouseholdName	Required
IntegrationID	Required
ExternalSystemID	Required
HouseholdId	Read-only

Table 553. Required and Read-Only Fields for the Household Object

Field Name	Type
PrimaryContactId	Read-only
PrimaryContactExternalId	Read-only
PrimaryContactIntegrationId	Read-only
PrimaryContactFirstName	Read-only
PrimaryContactLastName	Read-only
Timezone	Read-only
HouseholdCurrency	Read-only
LastActivity	Read-only
HeadDOB	Read-only
TotalIncome	Read-only
TotalAssets	Read-only
TotalExpenses	Read-only
TotalLiabilities	Read-only
TotalNetWorth	Read-only
RiskProfile	Read-only
ExperienceLevel	Read-only
InvestmentHorizon	Read-only
CurrentInvestmentMix	Read-only
Objective	Read-only
PrimaryGoal	Read-only
Audit Fields	Read-only

Table 554 details the pick map fields for the household object.

Table 554. Pick Map Field for the Household Object

Pick Map Field	Maps To
ContactExternalId	ContactId
ContactIntegrationId	

Table 555 provides a list of the user key combinations for the household object.

Table 555. User Key Fields on the Household Object

User Key Field Combinations
HouseholdName
Id
ExternalSystemId

Table 556 details the picklists available for the household object.

Table 556. Picklists Available for the Household Object

Field Name
Segment
Type

Insurance Property

The insurance property object allows you to define and record details about an insurance property. Insurance properties are typically the properties of a contact that are included in an insurance policy or a claim.

Child Objects

Coverage, Damage.

Methods Called

Table 557 details the methods called by the Insurance Property service.

Table 557. Methods Called by Insurance Property Service

Method	Name as Defined in Service
"Delete" on page 105	InsurancePropertyDelete
"Execute" on page 108	InsurancePropertyExecute
"Insert" on page 112	InsurancePropertyInsert
"QueryPage" on page 115	InsurancePropertyQueryPage
"Update" on page 136	InsurancePropertyUpdate

Fields

Table 558 details the required and read-only fields for the insurance property object.

Table 558. Read-Only Fields on the Insurance Property Object

Field Name	Type
PolicyName	Required
Type	Required
CurrencyCode	Read-only
Owner	Read-only
OwnerId	Read-only
Audit Fields	Read-only

Table 559 details the pickmap fields for the insurance property object.

Table 559. Pick Map Fields for the Insurance Property Object

Pick Map Field	Maps To
PolicyExternalSystemId	PolicyId
PolicyIntegrationId	PolicyId

Table 560 provides a list of the user key combinations for the insurance property object.

Table 560. User Key Fields on the Insurance Property Object

User Key Field Combinations
Type
Id
ExternalSystemId

Table 561 details the picklists available for the insurance property object.

Table 561. Picklists Available for the Insurance Property Object

Field Name
Type

Involved Party

The involved party object allows you to define and record details about an involved party, which is typically a contact involved in an insurance claim.

Child Objects

None

Methods Called

Table 562 details the methods called by the Involved Party service.

Table 562. Methods Called by Involved Party Service

Method	Name as Defined in Service
“Delete” on page 105	InvolvedPartyDelete
“Execute” on page 108	InvolvedPartyExecute
“Insert” on page 112	InvolvedPartyInsert
“QueryPage” on page 115	InvolvedPartyQueryPage
“Update” on page 136	InvolvedPartyUpdate

Fields

Table 563 details the required and read-only fields for the involved party object.

Table 563. Read-Only Fields on the Involved Party Object

Field Name	Type
ClaimNumber	Required
Role	Required
InvolvedPartyName	Required
Contact	Required
CurrencyCode	Read-only
Owner	Read-only
OwnerId	Read-only
Audit Fields	Read-only

Table 564 details the pickmap fields for the involved party object.

Table 564. Pick Map Fields for the Involved Party Object

Pick Map Field	Maps To
ClaimExternalSystemId	ClaimId
ClaimIntegrationId	ClaimId

Table 565 provides a list of the user key combinations for the involved party object.

Table 565. User Key Fields on the Involved Party Object

User Key Field Combinations
InvolvedPartyName
Id
ExternalSystemId

Table 566 details the picklists available for the involved party object.

Table 566. Picklists Available for the Involved Party Object

Field Name
Role
RoleinAccident
Location

Policy

The policy object allows you to define and record details about an insurance policy.

Child Objects

Activity, Claim, Coverage, Insurance Property, Lead, Contact Team, Policy, Policy Holder, Service Request.

Methods Called

Table 567 details the methods called by the Policy service.

Table 567. Methods Called by Policy Service

Method	Name as Defined in Service
"Delete" on page 105	PolicyDelete
"Execute" on page 108	PolicyExecute
"Insert" on page 112	PolicyInsert
"QueryPage" on page 115	PolicyQueryPage
"Update" on page 136	PolicyUpdate

Fields

Table 568 details the required and read-only fields for the policy object.

Table 568. Read-Only Fields on the Policy Object

Field Name	Type
PolicyNumber	Required
PolicyType	Required
CurrencyCode	Read-only
Owner	Read-only
OwnerId	Read-only
Audit Fields	Read-only

Table 569 details the pickmap fields for the policy object.

Table 569. Pick Map Fields for the Policy Object

Pick Map Field	Maps To
ClaimExternalSystemId	ClaimId
ClaimIntegrationId	ClaimId

Table 570 provides a list of the user key combinations for the policy object.

Table 570. User Key Fields on the Policy Object

User Key Field Combinations
PolicyNumber
Id
ExternalSystemId

Table 571 details the picklists available for the policy object.

Table 571. Picklists Available for the Policy Object

Field Name
PolicyType
Status
SubStatus
BillingStatus
RateState
RatePlan
ReferralSource
PolicyPayMethod

Policy Holder

The policy holder object allows you to define and record details about an insurance policy holder. A policy holder is typically the contact that owns the policy that the insurance company manages.

Child Objects

None

Methods Called

Table 572 details the methods called by the Policy Holder service.

Table 572. Methods Called by Policy Holder Service

Method	Name as Defined in Service
"Delete" on page 105	PolicyHolderDelete
"Execute" on page 108	PolicyHolderExecute
"Insert" on page 112	PolicyHolderInsert
"QueryPage" on page 115	PolicyHolderQueryPage
"Update" on page 136	PolicyHolderUpdate

Fields

Table 573 details the required and read-only fields for the policy holder object.

Table 573. Read-Only Fields on the Policy Holder Object

Field Name	Type
PolicyName	Required
Role	Required
PolicyHolderName	Required
CurrencyCode	Read-only
Owner	Read-only
OwnerId	Read-only
Audit Fields	Read-only

Table 574 details the pickmap fields for the policy holder object.

Table 574. Pick Map Fields for the Policy Holder Object

Pick Map Field	Maps To
PolicyExternalSystemId	PolicyId
PolicyIntegrationId	PolicyId

Table 575 provides a list of the user key combinations for the policy holder object.

Table 575. User Key Fields on the Policy Holder Object

User Key Field Combinations
PolicyHolderName
Id
ExternalSystemId

Table 576 details the picklists available for the policy holder object.

Table 576. Picklists Available for the Policy Holder Object

Field Name
InsuredType
Role

Portfolio

The portfolio object allows you to define and record details about the collection of financial services that you can provide to an account. Financial services include loans, credit cards, insurance, general banking, and so on.

Child Objects

Financial Account, Financial Plan.

Methods Called

Table 577 details the methods called by the Portfolio service.

Table 577. Methods Called by Portfolio Service

Method	Name as Defined in Service
“Delete” on page 105	PortfolioDelete
“Execute” on page 108	PortfolioExecute
“Insert” on page 112	PortfolioInsert
“QueryPage” on page 115	PortfolioQueryPage
“Update” on page 136	PortfolioUpdate

Fields

Table 578 details the read-only fields for the portfolio object.

Table 578. Read-Only Fields on the Portfolio Object

Field Name	Type
PortfolioId	Read-only
Owner	Read-only
OwnerId	Read-only
PrimaryContact	Read-only
Audit Fields	Read-only

Table 579 details the pickmap fields for the portfolio object.

Table 579. Pick Map Fields for the Portfolio Object

Pick Map Field	Maps To
InstitutionExternalId	InstitutionId
InstitutionIntegrationId	InstitutionId
InstitutionName	InstitutionId
InstitutionLocation	InstitutionId
Product	ProductId
ProductExternalId	ProductId
ProductIntegrationId	ProductId

Table 580 provides a list of the user key combinations for the portfolio object.

Table 580. User Key Fields on the Portfolio Object

User Key Field Combinations
AccountNumber and ProductId
Id
ExternalSystemId

Table 581 details the picklists available for the portfolio object.

Table 581. Picklists Available for the Portfolio Object

Field Name
AccountType
Status
TermUnit

Automotive Edition Parent Objects (Web Services v2.0)

This topic includes the parent objects available with Oracle CRM On Demand Automotive Edition.

NOTE: To download WSDL files for these objects, you must be given access to the object. If you do not have access to the object, it is not available to download from the Web Services Administration page or available to use the Web service calls. For assistance in gaining access to the objects, contact your Oracle CRM On Demand service provider.

The following objects are detailed in this topic:

- “Dealer” on page 538
- “Vehicle” on page 540

Dealer

The dealer object stores information about dealerships in the automotive industry, for example, the name of the dealership, the identity of the parent dealership, the site on which the dealership is based, and so on. The dealer object does not have any parent objects.

Child Objects

Attachment

For information about using attachments with this object, see Appendix A, “Using Attachments With Web Services On Demand.”

Methods Called

Table 582 details the methods called by the Dealer service.

Table 582. Methods Called by Dealer Service

Method	Name as Defined in Service
"Delete" on page 105	DealerDelete
"Execute" on page 108	DealerExecute
"Insert" on page 112	DealerInsert
"QueryPage" on page 115	DealerQueryPage
"Update" on page 136	DealerUpdate

Fields

Table 583 details the required and read-only fields for the dealer object.

Table 583. Required and Read-Only Fields for the Dealer Object

Field Name	Type
DealerId	Required
DealerIntegrationId	Required
DealerExternalSystemID	Required
DealerId	Read-only
DealerType	Read-only
Audit Fields	Read-only

Table 584 details the pick map fields for the dealer object.

Table 584. Pick Map Fields for the Dealer Object

Pick Map Field	Maps To
Owner	OwnerId
ParentDealerExternalSystemId	ParentDealerId
ParentDealerIntegrationId	ParentDealerId
ParentDealerName	ParentDealerId
ParentDealerSite	ParentDealerId

Table 585 provides a list of the user key combinations for the dealer object.

Table 585. User Key Fields on the Dealer Object

User Key Field Combinations
DealerId
ExternalSystemID

Table 586 details the picklists available for the dealer object.

Table 586. Picklists Available for the Dealer Object

Field Name
ParentDealerName
ParentDealerSite

Vehicle

The vehicle object allows you to create and store information about a vehicle, for example, a car, a truck, a van, and so on, that your company would like to sell to a contact or account. This information includes the vehicle's current mileage, the invoice price, the dealership, the make, and so on.

Child Objects

None

Methods Called

Table 587 details the methods called by the Vehicle service.

Table 587. Methods Called by Vehicle Service

Method	Name as Defined in Service
"Delete" on page 105	VehicleDelete
"Execute" on page 108	VehicleExecute
"Insert" on page 112	VehicleInsert
"QueryPage" on page 115	VehicleQueryPage
"Update" on page 136	VehicleUpdate

Fields

Table 588 details the required and read-only fields for the vehicle object.

Table 588. Required and Read-Only Fields for the Vehicle Object

Field Name	Type
VehicleId	Read-only
Contact	Read-only
ProductType	Read-only
SellingDealer	Read-only
ServicingDealer	Read-only
Audit Fields	Read-only

Table 589 details the pick map fields for the vehicle object.

Table 589. Pick Map Fields for the Vehicle Object

Pick Map Field	Maps To
AccountName	AccountId
AccountSite	AccountId
AccountIntegrationId	AccountId
AccountExternalID	AccountId
SellingDealerExternalId	SellingDealerId
SellingDealerIntegrationId	SellingDealerId
ServicingDealerExternalId	ServicingDealerId
ServicingDealerIntegrationId	ServicingDealerId

Table 590 provides a list of the user key combinations for the vehicle object.

Table 590. User Key Fields on the Vehicle Object

User Key Field Combinations
Id
ExternalSystemId
ProductId

Table 590. User Key Fields on the Vehicle Object

User Key Field Combinations
SerialNumber
ProductExternalId

Table 591 details the picklists available for the vehicle object.

Table 591. Picklists Available for the Vehicle Object

Field Name
Body
Door
Engine
ExteriorColor
InteriorColor
Location
Make
Model
VehicleOwnedBy
Status
Transmission
Trim
UsedNew
WarrantyType
Year

Child Objects (Web Services v2.0)

This topic includes a list of child objects that are used in Oracle On Demand Web Services.

The list includes:

- Objects that are child objects only and are not themselves parent objects.
- Objects that are also parent objects.

Various parent objects are also child objects of other objects. This includes a number of industry-specific objects added in Release 16 and Release 17 of Oracle CRM On Demand. See the topics about the parent objects for information about the parent-child relationships. For these objects, the field information given for parent objects is the same for the objects when they are child objects, and is therefore not repeated in this topic.

For each child object, the methods that can be called are as shown in Table 592. The methods that you can use on a child object are the same as for its parent object.

Table 592. Methods Called On Child Objects

Method	Name as Defined in Service
"Delete" on page 105	<object>Delete
"Execute" on page 108	<object>Execute
"Insert" on page 112	<object>Insert
"QueryPage" on page 115	<object>QueryPage
"Update" on page 136	<object>Update

The following child objects are listed:

- "Account" on page 544
- "Account Contact" on page 545
- "Account Opportunity" on page 547
- "Account Team" on page 548
- "Activity" on page 550
- "Activity Assessment" on page 552
- "Activity Assessment Value" on page 552
- "Address" on page 553
- "Asset" on page 554
- "Attachment" on page 555
- "Book" on page 556
- "Competitor" on page 557
- "Contact" on page 559
- "Contact Role" on page 560
- "Contact Team" on page 562
- "Custom Object 1 through 3" on page 563
- "Custom Object 4 through 15" on page 563
- "Delegated User" on page 565

NOTE: For the Quote and Order child objects, access is included for use by the Oracle CRM On Demand to Oracle E-Business Suite integration and is not available for non-Oracle use.

Account

The account object stores information about the companies that you do business with and is also used to track partners and competitors.

Parent Object

Contact

Fields

Table 593 details the required and read-only fields for the account child object.

Table 593. Required and Read-Only Fields for the Account Child Object

Field Name	Type
AccountId	Read-only

Table 594 details the pick map fields for the account child object.

Table 594. Pick Map Fields for the Account Child Object

Pick Map Field	Maps To
ExternalSystemId	AccountId

Table 595 provides a list of the user key combinations for the account child object.

Table 595. User Key Fields on the Account Child Object

User Key Field Combinations
AccountId
ExternalSystemId
Name and Location

Table 596 details the picklists available for the account child object.

Table 596. Picklists Available for the Account Child Object

Field Name
Call Frequency
Route
Status
Type

Account Contact

The account contact object stores information about the contacts associated with an account.

Parent Object

Account

Fields

Table 597 details the required and read-only fields for the account contact child object.

Table 597. Required and Read-Only Fields for the Account Contact Child Object

Field Name	Type
AccountLocation	Read-only
AccountMainPhone	Read-only
AccountType	Read-only

Table 598 details the pick map fields for the account contact child object.

Table 598. Pick Map Fields for the Account Contact Child Object

Pick Map Field	Maps To
AccountExternalSystemId	AccountId
AccountName	AccountId
ContactExternalSystemId	ContactId
ContactFullName	ContactId
ContactLastName	ContactFirstName
ContactLastName	ContactId

Table 599 provides a list of the user key combinations for the account contact child object.

Table 599. User Key Fields on the Account Contact Child Object

User Key Field Combinations
AccountId
AccountExternalSystemId
AccountLocation and AccountName
ContactExternalSystemId
ContactFirstName and ContactLastName

Table 600 details the picklists available for the account contact child object.

Table 600. Picklists Available for the Account Contact Child Object

Field Name
Role

Account Opportunity

The opportunity object allows employees to identify and record a potential revenue-generating event that has arisen with an account or contact. Opportunities can be generated from marketing campaigns when leads indicate that they are interested in a product or service that has been offered.

Parent Object

Account

Fields

Table 601 details the required and read-only fields for the account opportunity child object.

Table 601. Required and Read-Only Fields for the Account Opportunity Child Object

Field Name	Type
OwnerFullName	Read-only
PrimaryRevenueExpectedValue	Read-only
CreatedbyEmailAddress	Read-only
ModifiedbyEmailAddress	Read-only
FuriganaAccountName	Read-only
NameShadow	Read-only
AssignmentStatus	Read-only
LastAssignmentCompletionDate	Read-only
LastAssignmentSubmissionDate	Read-only

Table 602 details the pick map fields for the account opportunity child object.

Table 602. Pick Map Fields for the Account Opportunity Child Object

Pick Map Field	Maps To
AccountIntegrationId	AccountId
AccountExternalId	AccountId
AccountLocation	AccountId
Account	AccountId
SourceCampaignExternalId	CampaignId
SourceCampaign	CampaignId
KeyContactExternalId	KeyContactId
KeyContactLastName	KeyContactId
DealerExternalId	DealerId
OwnerAlias	OwnerId
CustomObject1Name	CustomObject1Id
CustomObject2Name	CustomObject2Id
CustomObject3Name	CustomObject3Id

Table 603 provides a list of the user key combinations for the account opportunity child object.

Table 603. User Key Fields on the Account Opportunity Child Object

User Key Field Combinations
OpportunityId
Id

Account Team

The team object stores information on the team that is assigned to a particular account or contact. In this way, a team of employees can be dedicated to an account or contact, ensuring that the activities, service requests, leads, and opportunities surrounding that account or contact are always kept up-to-date and are attended to regularly.

Parent Object

Account

Fields

Table 604 details the required and read-only fields for the account team child object.

Table 604. Required and Read-Only Fields for the Account Team Child Object

Field Name	Type
AccountAccess	Required
UserId	Required
FirstName	Read-only

Table 605 details the pick map fields for the account team child object.

Table 605. Pick Map Fields for the Account Team Child Object

Pick Map Field	Maps To
AccountAccess	AccountAccessId
AccountName	AccountId
ContactAccess	ContactAccessId
LastName	FirstName
OpportunityAccess	OpportunityAccessId
UserAlias	UserId
UserEmail	UserId
UserExternalSystemId	UserId

Table 606 provides a list of the user key combinations for the account team child object.

Table 606. User Key Fields on the Account Team Child Object

User Key Field Combinations
UserId
UserExternalSystemId
UserEmail
UserAlias

Table 607 details the picklists available for the account team child object.

Table 607. Picklists Available for the Account Team Child Object

Field Name
TeamRole

Activity

The activity object stores information on an activity that a user must carry out, for example, a callback activity for an account.

Parent Objects

Account, Campaign, Contact, Lead, Opportunity.

Fields

Table 608 details the required and read-only fields for the activity child object.

Table 608. Required and Read-Only Fields for the Activity Child Object

Field Name	Type
AccountLocation	Read-only
CreatedDetail	Read-only
MEEventName	Read-only
CreatedbyEmailAddress	Read-only
ModifiedbyEmailAddress	Read-only
CODInteractionTime	Read-only
CODWrapUpTime	Read-only
CODHandleTime	Read-only
CODIVRTime	Read-only
CODQueueHoldTime	Read-only
CODTotalHoldTime	Read-only
DescriptionShadow	Read-only
DealerName	Read-only
Audit Fields	Read-only

Table 609 details the pick map fields for the activity child object.

Table 609. Pick Map Fields for the Activity Child Object

Pick Map Field	Maps To
AccountIntegrationId	AccountId
AccountExternalId	AccountId
CampaignExternalId	CampaignId
CampaignIntegrationId	CampaignId
DelegatedBy	DelegatedById
FundRequest	FundRequestId
OpportunityIntegrationId	OpportunityId
OpportunityExternalId	OpportunityId
PrimaryContactIntegrationId	PrimaryContactId
PrimaryContactExternalId	PrimaryContactId
LeadIntegrationId	LeadId
LeadExternalId	LeadId
OwnerExternalId	OwnerId
OwnerIntegrationId	OwnerId
SRIntegrationId	SRId
SRExternalId	SRId
CustomObject1Name	CustomObject1Id
CustomObject2Name	CustomObject2Id
CustomObject3Name	CustomObject3Id

Table 610 provides a list of the user key combinations for the activity child object.

Table 610. User Key Fields on the Activity Child Object

User Key Field Combinations
Id
ExternalSystemId
Description

Activity Assessment

The activity assessment object stores information about a sales assessment script associated with an activity.

Parent Object

Activity

Fields

Table 611 details the required and read-only fields for the activity assessment child object.

Table 611. Required and Read-Only Fields for the Activity Assessment Child Object

Field Name	Type
TemplateId	Required
Name	Required
Type	Required
Audit Fields	Read-only

Table 612 provides a list of the user key combinations for the activity assessment child object.

Table 612. User Key Fields on the Activity Assessment Child Object

User Key Field Combinations
Name
TemplateId

Activity Assessment Value

The activity assessment value object stores information about the responses for a sales assessment script associated with an activity, that is, the values related to the activity assessment child object.

Parent Object

Activity

Fields

Table 613 details the required and read-only fields for the activity assessment value child object.

Table 613. Required and Read-Only Fields for the Activity Assessment Value Child Object

Field Name	Type
AssessId	Required
AttribId	Required
Audit Fields	Read-only

Table 614 provides a list of the user key combinations for the activity assessment value child object.

Table 614. User Key Fields on the Activity Assessment Value Child Object

User Key Field Combinations
AssessId
AttribId

Address

The address object stores information on the different addresses that are associated with accounts and contacts.

For more information about addresses, see “Adding and Updating Account Addresses” on page 396, and “Adding and Updating Contact Addresses” on page 411.

Parent Objects

Account, Contact

Fields

Table 615 details the required and read-only fields for the address child object.

Table 615. Required and Read-Only Fields for the Address Child Object

Field Name	Type
AddressId	Read-only

Table 616 details the pick map fields for the address child object.

Table 616. Pick Map Fields for the Address Child Object

Pick Map Field	Maps To
None	

Table 617 provides a list of the user key combinations for the address child object.

Table 617. User Key Fields on the Address Child Object

User Key Field Combinations
AddressId
ExternalId

Asset

The asset object stores information on the assets held by your accounts, for example, the products that an account has purchased.

Parent Object

Account

Fields

Table 618 details the required and read-only fields for the asset child object.

Table 618. Required and Read-Only Fields for the Asset Child Object

Field Name	Type
AssetId	Required
ContactAssetId	Read-only
ExternalSystemId	Read-only
Product	Read-only
ProductId	Read-only
ProductPartNumber	Read-only
ProductType	Read-only
ProductStatus	Read-only

Table 618. Required and Read-Only Fields for the Asset Child Object

Field Name	Type
CreatedByandDate	Read-only
ModifiedByandDate	Read-only
SerialNumber	Read-only

Table 619 details the pick map fields for the asset child object.

Table 619. Pick Map Fields for the Asset Child Object

Pick Map Field	Maps To
AccountIntegrationId	AccountId
Manufacturer	ManufacturerId
PreferredServiceDealer	PreferredServiceDealerID

Table 620 provides a list of the user key combinations for the asset child object.

Table 620. User Key Fields on the Asset Child Object

User Key Field Combinations
AssetId
ExternalSystemId

Attachment

The attachment object stores information about a file or URL that is attached to a record in Oracle CRM On Demand.

Parent Objects

Account, Accreditation, Activity, Campaign, Certification, Contact, Course, CustomObject, Dealer, Exam, Lead, Opportunity, Service Request, and Solution.

Fields

Table 621 details the required and read-only fields for the attachment object.

Table 621. Required and Read-Only Fields for the Attachment Child Object

Field Name	Type
DisplayFileName	Required
FileNameOrURL	Required
FileDate	Read-only
FileSize	Read-only
<parent_object>Id	Read-only
Id	Read-only
ModId	Read-only
Audit Fields	Read-only

Table 622 provides a list of the user key combinations for the attachment child object.

Table 622. User Key Fields on the Attachment Child Object

User Key Field Combinations
Id
ExternalSystemId
FileNameOrURL and FileExtension

Book

The book object provides a way of segmenting data according to the organizational units of your business, such as territories or products. Administrators can create book hierarchies based on how they want to organize your information, and then set up users to have the appropriate level of access to books.

Parent Object

Accreditation, Application, Certification, Course, CustomObject, Deal Registration, Exam, MedEd, Messaging Plan, Partner, Solution, Special Pricing Request.

Fields

Table 623 details the required and read-only fields for the book child object.

Table 623. Required and Read-Only Fields for the Book Child Object

Field Name	Type
BookName	Required
BookId	Read-only
SystemAssociateFlag	Read-only
ModId	Read-only
Audit Fields	Read-only

Table 624 details the pick map fields for the book child object.

Table 624. Pick Map Fields for the Book Child Object

Pick Map Field	Maps To
BookName	BookId

Table 625 provides a list of the user key combinations for the book child object.

Table 625. User Key Fields on the Book Child Object

User Key Field Combinations
BookId
BookName

Competitor

The competitor object stores the information on competitors for your accounts and opportunities.

NOTE: This topic covers both the account competitor and opportunity competitor child objects. The Parent Object column in the tables in this topic indicates to which objects the information is applicable.

Parent Object

Account, Opportunity

Fields

Table 626 details the required and read-only fields for the competitor child object.

Table 626. Required and Read-Only Fields for the Competitor Child Object

Field Name	Type	Parent Object
CompetitorId	Required	Both
ReverseRole	Required	Both
Role	Required	Both
StartDate	Required	Both

Table 627 details the pick map fields for the competitor child object.

Table 627. Pick Map Fields for the Competitor Child Object

Pick Map Field	Maps To	Parent Object
CompetitorExternalId	CompetitorId	Both
CompetitorName	CompetitorId	Both
CompetitorFullName	CompetitorId	Both
CompetitorExternalId	AccountName	Opportunity
CompetitorName	AccountName	Opportunity

Table 628 provides a list of the user key combinations for the competitor child object.

Table 628. User Key Fields on the Competitor Child Object

User Key Field Combinations	Parent Object
Id	Both
Role & ReverseRole & CompetitorId	Both
Role & ReverseRole & CompetitorExternalId	Both
Role & ReverseRole & CompetitorName	Opportunity
Role & ReverseRole & AccountName	Opportunity

Table 629 details the picklists available for the competitor child object.

Table 629. Picklists Available for the Competitor Child Object

Field Name	Parent Object
ReverseRole	Both
Role	Both

Contact

The contact object stores information on individuals with whom your organization has a relationship. It allows the user to store information on individuals who are external to your company, but who are associated with the business process. Contacts stored in the Oracle CRM On Demand database can also be associated with an account.

Parent Object

Activity, Account, Opportunity

Fields

Table 630 details the required and read-only fields for the contact child object.

Table 630. Required and Read-Only Fields for the Contact Child Object

Field Name	Type
ContactId	Read-only
ContactFirstName	Read-only
ContactLastName	Read-only
ContactAccountId	Read-only
ContactAccountName	Read-only
ContactAccountLocation	Read-only
ContactAccountIntegrationId	Read-only
ContactAccountExternalSystemId	Read-only
Audit Fields	Read-only

Table 631 details the pick map fields for the contact child object.

Table 631. Pick Map Fields for the Contact Child Object

Pick Map Field	Maps To
ContactExternalId	ContactId
ContactIntegrationId	ContactId

Table 632 provides a list of the user key combinations for the contact child object.

Table 632. User Key Fields on the Contact Child Object

User Key Field Combinations
ActivityContactId
ContactExternalId

Contact Role

The contact role object stores information on the different roles that a contact can hold within an account or opportunity.

NOTE: This topic covers both the account contact role and opportunity contact role child objects. The Parent Object column in the tables in this topic indicates to which objects the information is applicable.

Parent Object

Account, Opportunity

Fields

Table 633 details the required and read-only fields for the contact role child object.

Table 633. Required and Read-Only Fields for the Contact Role Child Object

Field Name	Type	Parent Object
AccountId	Required	Both
ContactId	Required	Opportunity

Table 634 details the pick map fields for the contact role child object.

Table 634. Pick Map Fields for the Contact Role Child Object

Pick Map Field	Maps To	Parent Object
AccountExternalSystemId	AccountId	Account
AccountName	AccountId	Account
ContactExternalSystemId	ContactId	Both
ContactLastName	ContactId	Opportunity
ContactExternalSystemId	ContactFirstName	Opportunity
ContactLastName	ContactFirstName	Opportunity
ContactExternalSystemId	ContactLastName	Opportunity

Table 635 provides a list of the user key combinations for the contact role child object.

Table 635. User Key Fields on the Contact Role Child Object

User Key Field Combinations	Parent Object
Id	Account
Role, ContactId	Account
Role, ContactExternalSystemId	Account
Role, ContactFirstName, ContactLastName	Account
ContactExternalSystemId, BuyingRole	Opportunity
ContactFirstName, ContactLastName, BuyingRole	Opportunity

Table 636 details the picklists available for the contact role child object.

Table 636. Picklists Available for the Contact Role Child Object

Field Name	Parent Object
Role	Account
BuyingRole	Opportunity

Contact Team

The team object stores information on the team that is assigned to a particular account or contact. In this way, a team of employees can be dedicated to an account or contact, ensuring that the activities, service requests, leads, and opportunities surrounding that account or contact are always kept up-to-date and are attended to regularly.

Parent Object

Contact

Fields

Table 637 details the required and read-only fields for the contact team child object.

Table 637. Required and Read-Only Fields for the Contact Team Child Object

Field Name	Type
ContactTeamId	Read-only
UserFirstName	Read-only
UserLastName	Read-only
UserRole	Read-only

Table 638 details the pick map fields for the contact team child object.

Table 638. Pick Map Fields for the Contact Team Child Object

Pick Map Field	Maps To
UserExternalSystemId	UserId
UserIntegrationId	UserId

Table 639 provides a list of the user key combinations for the contact team child object.

Table 639. User Key Fields on the Contact Team Child Object

User Key Field Combinations
None

Table 640 details the picklists available for the contact team child object.

Table 640. Picklists Available for the Contact Team Child Object

Field Name
TeamRole

Custom Object 1 through 3

The custom objects 1, 2, and 3.

Parent Objects

Account, Activity, Contact, Lead, Opportunity

Fields

Table 641 details the required and read-only fields for the custom object child object.

Table 641. Required and Read-Only Fields for the Custom Object Child Object

Field Name	Type
Name	Required
Id	Read-only

Table 642 provides a list of the user key combinations for the custom object child object.

Table 642. User Key Fields on the Custom Object Child Object

User Key Field Combinations
Id
ExternalSystemId
Name

No pickmap or picklist fields are available.

Custom Object 4 through 15

The custom objects 4 through 15.

Parent Objects

All top-level objects.

Fields

Table 643 details the required and read-only fields for the custom object child object.

Table 643. Required and Read-Only Fields for the Custom Object Child Object

Field Name	Type
Name	Required
Currency	Read-only
Audit Fields	Read-only

Table 644 details the pick map fields for the custom object child object.

Table 644. Pick Map Fields for the Custom Object Child Object

Pick Map Field	Maps To
Owner	OwnerId

Table 645 provides a list of the user key combinations for the custom object child object.

Table 645. User Key Fields on the Custom Object Child Object

User Key Field Combinations
Id
ExternalSystemId

Table 646 details the picklists available for the custom object child object.

Table 646. Picklists Available for the Custom Object Child Object

Field Name
Type

Delegated User

The delegated user object stores information about a user that can impersonate another user. The delegated user is added so that he or she can access a user's data, for example, when a person is on vacation or leaves the company.

Parent Object

User

Fields

Table 647 details the required and read-only fields for the delegated user child object.

Table 647. Required and Read-Only Fields for the Delegated User Child Object

Field Name	Type
UserId	Required
UserRole	Required
Audit Fields	Read-only

Table 648 provides a list of the user key combinations for the delegated user child object.

Table 648. User Key Fields on the Delegated User Child Object

User Key Field Combinations
Id
ExternalSystemId

Table 649 details the picklists available for the delegated user child object.

Table 649. Picklists Available for the Delegated User Child Object

Field Name
UserRole

Exchange Rate History

The Exchange rate history object stores historical information about exchange rates for the currencies used by a company.

Parent Object

Exchange Rate

Usage

The ExchangeRateHistory child object can be queried to give the history of exchange rates for the company.

Fields

Table 650 details the required and read-only fields for the exchange rate object.

Table 650. Required and Read-Only Fields for the Exchange Rate History Object

Field Name	Type
FromCurrencyCode	Required
ToCurrencyCode	Required
ExchangeDate	Required
ExchangeRate	Required

Table 651 provides a list of the user key combinations for the exchange rate history object.

Table 651. User Key Fields on the Exchange Rate History Object

User Key Field Combinations
FromCurrencyCode
ToCurrencyCode
ExchangeDate

Interests

The interests object stores information about things in which a contact is interested, such as products, services, or hobbies.

Parent Object

Contact

Fields

Table 652 details the required and read-only fields for the interests child object.

Table 652. Required and Read-Only Fields for the Interests Child Object

Field Name	Type
Category	Required
Interests	Required
InterestId	Read-only
Audit Fields	Read-only

Table 653 details the pick map fields for the interests child object.

Table 653. Pick Map Fields for the Interests Child Object

Pick Map Field	Maps To
InterestExternalSystemId	InterestId

Table 654 provides a list of the user key combinations for the interests child object.

Table 654. User Key Fields on the Interests Child Object

User Key Field Combinations
InterestId
ExternalSystemId

Table 655 details the picklists available for the interests child object.

Table 655. Picklists Available for the Interests Child Object

Field Name
Category
Subjects

Invitee

The invitee object stores information about invitees to medical education events, including feedback about the invitation.

Parent Object

MedEd

Fields

Table 656 details the required and read-only fields for the invitee child object.

Table 656. Required and Read-Only Fields for the Invitee Child Object

Field Name	Type
InviteeStatus	Required
InviteeId	Read-only
Audit Fields	Read-only

Table 657 details the pick map fields for the invitee child object.

Table 657. Pick Map Fields for the Invitee Child Object

Pick Map Field	Maps To
InviteeExternalSystemId	InviteeId

Table 658 provides a list of the user key combinations for the invitee child object.

Table 658. User Key Fields on the Invitee Child Object

User Key Field Combinations
ContactIdExternalId
MedEdInviteeId

Table 659 details the picklists available for the invitee child object.

Table 659. Picklists Available for the Invitee Child Object

Field Name
InviteeStatus

Lead

The lead object stores information on a company or individual with whom an opportunity can be created. It allows the user to identify the companies that might be interested in a product or service. Leads are usually generated as part of a marketing campaign.

Parent Objects

Account, Campaign, Contact, Opportunity

Fields

Table 660 details the required and read-only fields for the lead child object.

Table 660. Required and Read-Only Fields for the Lead Child Object

Field Name	Type
OwnerFullName	Read-only
SalesRepFullName	Read-only
CreatedbyEmailAddress	Read-only
ModifiedbyEmailAddress	Read-only
ReferredByFullName	Read-only
FuriganaAccountName	Read-only
FuriganaContactFirstName	Read-only
FuriganaContactLastName	Read-only
FirstNameShadow	Read-only
LastNameShadow	Read-only
CompanyNameShadow	Read-only
AssignmentStatus	Read-only
LastAssignmentCompletionDate	Read-only
LastAssignmentSubmissionDate	Read-only

Table 661 details the pick map fields for the lead child object.

Table 661. Pick Map Fields for the Lead Child Object

Pick Map Field	Maps To
AccountIntegrationId	AccountId
AccountExternalId	AccountId

Table 661. Pick Map Fields for the Lead Child Object

Pick Map Field	Maps To
AccountLocation	AccountId
AccountName	AccountId
CampaignExternalId	CampaignId
CampaignIntegrationId	CampaignId
CampaignName	CampaignId
ContactExternalId	ContactId
ContactIntegrationId	ContactId
OpportunityIntegrationId	OpportunityId
OpportunityExternalId	OpportunityId
OwnerExternalId	OwnerId
OwnerIntegrationId	OwnerId
SalesRepAlias	SalesRepId
CustomObject1Name	CustomObject1Id
CustomObject2Name	CustomObject2Id
CustomObject3Name	CustomObject3Id

Table 662 provides a list of the user key combinations for the lead child object.

Table 662. User Key Fields on the Lead Child Object

User Key Field Combinations
LeadId
Id

Note

The note object stores information about the notes available in the Message Center in the Oracle CRM On Demand application.

Parent Objects

Account, Campaign, Contact, Opportunity, Service Allotment

Fields

Table 663 details the required and read-only fields for the note child object.

Table 663. Required and Read-Only Fields for the Note Child Object

Field Name	Type
Subject	Required
NoteId	Read-only
Audit Fields	Read-only

Table 664 provides a list of the user key combinations for the note child object.

Table 664. User Key Fields on the Note Child Object

User Key Field Combinations
NoteId

Opportunity

The opportunity object allows employees to identify and record a potential revenue-generating event that has arisen with an account or contact. Opportunities can be generated from marketing campaigns when leads indicate that they are interested in a product or service that has been offered.

Parent Object

Contact

Fields

Table 665 details the required and read-only fields for the opportunity child object.

Table 665. Required and Read-Only Fields for the Opportunity Child Object

Field Name	Type
OpportunityName	Required
CloseDate	Required

Table 666 provides a list of the user key combinations for the opportunity child object.

Table 666. User Key Fields on the Opportunity Child Object

User Key Field Combinations
Id
ExternalSystemId
OpportunityName

Opportunity Child

The opportunity child object is the child of a parent opportunity in an opportunity hierarchy. Such hierarchies are used for sales opportunities that have multiple components.

The same fields are accessible for the opportunity child object as for the opportunity parent object.

For more information about opportunity hierarchies, see the Oracle CRM On Demand online help.

Parent Object

Opportunity

Opportunity Team

The opportunity team object stores information on the team that is assigned to a particular opportunity.

Parent Object

Opportunity

Fields

Table 667 details the required and read-only fields for the opportunity team child object.

Table 667. Required and Read-Only Fields for the Opportunity Team Child Object

Field Name	Type
OpportunityAccess	Required
UserId	Required
UserFirstName	Read-only

Table 668 details the pick map fields for the opportunity team child object.

Table 668. Pick Map Fields for the Opportunity Team Child Object

Pick Map Field	Maps To
UserLastName	UserFirstName
OpportunityAccess	OpportunityAccessId
UserAlias	UserId
UserEmail	UserId
UserExternalSystemId	UserId

Table 669 provides a list of the user key combinations for the opportunity team child object.

Table 669. User Key Fields on the Opportunity Team Child Object

User Key Field Combinations
UserId
UserExternalSystemId
UserEmail
UserAlias

Table 670 details the picklists available for the opportunity team child object.

Table 670. Picklists Available for the Opportunity Team Child Object

Field Name
TeamRole

Products Detailed

The Products Detailed object stores the information on product details for an activity. This is used, for example, to record information about products discussed on sales calls to customers.

Parent Object

Activity

Fields

Table 671 details the required and read-only fields for the products detailed child object.

Table 671. Required and Read-Only Fields for the Products Detailed Child Object

Field Name	Type
ProductId	Required
Indication	Required
ProductDetailedId	Read-only
Audit Fields	Read-only

Table 672 details the pick map fields for the products detailed child object.

Table 672. Pick Map Fields for the Products Detailed Child Object

Pick Map Field	Maps To
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId

Table 673 provides a list of the user key combinations for the products detailed child object.

Table 673. User Key Fields on the Products Detailed Child Object

User Key Field Combinations
ProductDetailedId
ProductDetailedExternalSytemId
ProductExternalSytemId
ProductId
ProductExternalSytemId
Name
Indication

Promotional Items Dropped

The Promotional Items Dropped object stores the information on promotional items dropped for an activity.

Parent Object

Activity

Fields

Table 674 details the required and read-only fields for the promotional items dropped child object.

Table 674. Required and Read-Only Fields for the Promotional Items Dropped Child Object

Field Name	Type
ProductName	Required
Quantity	Required
ProductAllocationId	Required
Audit Fields	Read-only

Table 675 details the pick map fields for the promotional items dropped child object.

Table 675. Pick Map Fields for the Promotional Items Dropped Child Object

Pick Map Field	Maps To
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId

Table 676 provides a list of the user key combinations for the promotional items dropped child object.

Table 676. User Key Fields on the Promotional Items Dropped Child Object

User Key Field Combinations
Id

Recipient

The recipient object stores information about a recipient associated with a campaign.

Parent Object

Campaign

Fields

Table 677 details the required and read-only fields for the recipient child object.

Table 677. Required and Read-Only Fields for the Recipient Child Object

Field Name	Type
ContactId	Required
ModifiedDate	Read-only

Table 678 provides a list of the user key combinations for the recipient child object.

Table 678. User Key Fields on the Recipient Child Object

User Key Field Combinations
None

Related Contact

The related contact object stores information about a contact that has a relationship with the parent contact in question. The details of the related contact child object are inherited from a particular contact parent object.

Parent Object

Contact

Fields

Table 679 details the required and read-only fields for the related contact child object.

Table 679. Required and Read-Only Fields for the Related Contact Child Object

Field Name	Type
ContactRelationshipId	Read-only
ContactId	Read-only
RelatedContactFirstName	Read-only
RelatedContactLastName	Read-only

Table 680 details the pick map fields for the related contact child object.

Table 680. Pick Map Fields for the Related Contact Child Object

Pick Map Field	Maps To
RelatedContactExternalId	RelatedContactId
RelatedContactIntegrationId	RelatedContactId

Table 681 provides a list of the user key combinations for the related contact child object.

Table 681. User Key Fields on the Related Contact Child Object

User Key Field Combinations
None

Table 682 details the picklists available for the related contact child object.

Table 682. Picklists Available for the Related Contact Child Object

Field Name
Relationship
Status

Revenue

The revenue object stores monetary information about accounts, contacts, and their associated opportunities. This includes information on the revenue available, expected revenue, and also information about the products associated with the accounts, contacts, opportunities, and so on.

NOTE: The Revenue child object of Opportunity is actually called Product.

Parent Objects

Account, Contact, Opportunity

Fields

Table 683 details the required and read-only fields for the revenue child object.

Table 683. Required and Read-Only Fields for the Revenue Child Object

Field Name	Type
RevenueId	Required
PartNumber	Required
Revenue	Required
ContactFullName	Required
Audit Fields	Read-only

Table 684 details the pick map fields for the revenue child object.

Table 684. Pick Map Fields for the Revenue Child Object

Pick Map Field	Maps To
Product	ProductId
ProductExternalId	ProductId
ProductIntegrationId	ProductId
ProductCategory	ProductCategoryId
ProductCategoryExternalId	ProductCategoryId
ProductCategoryIntegrationId	ProductCategoryId

Table 685 provides a list of the user key combinations for the revenue child object.

Table 685. User Key Fields on the Revenue Child Object

User Key Field Combinations
RevenueId
ProductId
ProductExternalId

Table 686 details the picklists available for the revenue child object.

Table 686. Picklists Available for the Revenue Child Object

Field Name
Type
Status
Frequency

Sample Dropped

The SampleDropped object stores the information on samples for an activity. This is used, for example, to record information about samples left with the customer on sales calls to customers.

Parent Object

Activity

Fields

Table 687 details the required and read-only fields for the sample dropped child object.

Table 687. Required and Read-Only Fields for the Sample Dropped Child Object

Field Name	Type
ProductId	Required
Quantity	Required
SampleDroppedId	Read-only
Audit Fields	Read-only

Table 688 details the pick map fields for the sample dropped child object.

Table 688. Pick Map Fields for the Sample Dropped Child Object

Pick Map Field	Maps To
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId

Table 689 provides a list of the user key combinations for the sample dropped child object.

Table 689. User Key Fields on the Sample Dropped Child Object

User Key Field Combinations
SampleDroppedId
SampleDroppedExternalSystemId
ProductId
ProductExternalSystemId
Quantity
ProductName

Service Request

The service request object allows customers to request information or assistance with a problem related to products or services purchased from your company. Service requests can be ranked for severity and prioritized accordingly.

Parent Objects

Account, Contact

Fields

Table 690 details the required and read-only fields for the service request child object.

Table 690. Required and Read-Only Fields for the Service Request Child Object

Field Name	Type
Name	Read-only
OwnerFullName	Read-only
CreatedbyEmailAddress	Read-only
ModifiedbyEmailAddress	Read-only
LastAssessmentDate	Read-only
AssignmentStatus	Read-only
FuriganaAccountName	Read-only
FuriganaContactFirstName	Read-only
FuriganaContactLastName	Read-only

Table 690. Required and Read-Only Fields for the Service Request Child Object

Field Name	Type
SRNumberShadow	Read-only
AbstractShadow	Read-only
LastAssignmentCompletionDate	Read-only
LastAssignmentSubmissionDate	Read-only
Audit Fields	Read-only

Table 691 details the pick map fields for the service request child object.

Table 691. Pick Map Fields for the Service Request Child Object

Pick Map Field	Maps To
AccountIntegrationId	AccountId
AccountExternalId	AccountId
AccountLocation	AccountId
Account	AccountId
AssetExternalId	AssetId
Dealer	DealerId
ContactExternalId	ContactId
ContactIntegrationId	ContactId
DealerExternalId	DealerId
OwnerExternalId	OwnerId
OwnerIntegrationId	OwnerId
OwnerAlias	OwnerId
Product	ProductId
CustomObject1Name	CustomObject1Id
CustomObject2Name	CustomObject2Id
CustomObject3Name	CustomObject3Id

Table 692 provides a list of the user key combinations for the service request child object.

Table 692. User Key Fields on the Service Request Child Object

User Key Field Combinations
ServiceRequestId
Id

User

The user object allows you to define and record details of all users in Oracle CRM On Demand, for example, name, position, contact details, manager, and so on.

Parent Object

Activity

Fields

Table 693 details the required and read-only fields for the user child object.

Table 693. Required and Read-Only Fields for the User Child Object

Field Name	Type
UserId	Read-only
UserEmail	Read-only
UserFirstName	Read-only
UserLastName	Read-only
UserRole	Read-only
Audit Fields	Read-only

Table 694 details the pick map fields for the user child object.

Table 694. Pick Map Fields for the User Child Object

Pick Map Field	Maps To
UserExternalId	UserId
UserIntegrationId	UserId

Table 695 provides a list of the user key combinations for the user child object.

Table 695. User Key Fields on the User Child Object

User Key Field Combinations
UserId
UserExternalId

A Using Attachments With Web Services On Demand

This appendix describes how to exchange attachments using Web services On Demand. It contains the following topics:

About Attachments

Attachments can be exchanged for a number of objects using Web Services On Demand. The parent objects that have Attachment child objects are: Account, Accreditation, Activity, Campaign, Contact, Course, Custom Object 01 - 03, Dealer, Exam, Lead, Opportunity, Service Request, and Solution.

File Attachments can be exchanged as text or as binary data. In the case of binary data, the data must be encoded as inline Base64 encoded data, which ensures that the binary data is in a printable format. Binary files must therefore be encoded into Base64 format in the content provided in SOAP requests.

File attachments added through Web services requests are limited to 20MB in size just as they are for the Oracle CRM On Demand UI. Attachments greater than 20MB in size are therefore rejected. To accommodate Base64 encoding of binary files, the request size limit for SOAP requests with attachment files is 28800 KB.

You can develop client applications that allow attachment files to be added or updated for multiple records. SOAP requests contain multiple records; the first record contains the attachment as well as a Content ID value. However, other records in the SOAP request do not need to contain the same attachment, they need only reference the Content ID value from the first record.

The Attachment Element

In a Web services call, the content of an attachment must be contained in the <Attachment> element, which has the complex data type SiebelXmlAttachmentType, as shown in the following sample WSDL for an Attachment child object:

```
<xsd:complexType name="ListOfAttachment">

<xsd:sequence>

  <xsd:element name="Attachment" maxOccurs="unbounded" minOccurs="0"
type="xsdLocal1:Attachment" />
```

```
</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="Attachment">

<xsd:sequence>

  <xsd:element name="Id" maxOccurs="1" minOccurs="0" type="xsdLocal1:string30"/>

  <xsd:element name="DisplayFileName" maxOccurs="1" minOccurs="0"
type="xsdLocal1:string200" />

  <xsd:element name="FileNameOrURL" maxOccurs="1" minOccurs="0"
type="xsdLocal1:string200" />

  <xsd:element name="FileExtension" maxOccurs="1" minOccurs="0"
type="xsdLocal1:string10" />

  <xsd:element name="FileDate" maxOccurs="1" minOccurs="0" type="xsd:string" />

  <xsd:element name="FileSize" maxOccurs="1" minOccurs="0" type="xsd:string" />

  <xsd:element name="ContactId" maxOccurs="1" minOccurs="0" type="xsd:string" />

  <xsd:element name="ExternalSystemId" maxOccurs="1" minOccurs="0"
type="xsdLocal1:string30" />

  <xsd:element name="Description" maxOccurs="1" minOccurs="0" type="xsd:string" />

  <xsd:element name="CreatedDate" maxOccurs="1" minOccurs="0" type="xsd:string" />

  <xsd:element name="CreatedById" maxOccurs="1" minOccurs="0" type="xsd:string" />

  <xsd:element name="CreatedBy" maxOccurs="1" minOccurs="0" type="xsd:string" />

  <xsd:element name="ModId" maxOccurs="1" minOccurs="0" type="xsdLocal1:string30"/>

  <xsd:element name="ModifiedDate" maxOccurs="1" minOccurs="0" type="xsd:string" />

  <xsd:element name="ModifiedById" maxOccurs="1" minOccurs="0" type="xsd:string" />

  <xsd:element name="ModifiedBy" maxOccurs="1" minOccurs="0" type="xsd:string" />

  <xsd:element name="Attachment" maxOccurs="1" minOccurs="0"
type="xsdLocal1:SiebelXmlAttachmentType"/>

</xsd:sequence>

</xsd:complexType>

...

<xsd:complexType name="SiebelXmlAttachmentType">

<xsd:simpleContent>

<xsd:extension base="xsd:base64Binary">
```

```
  <xsd:attribute name="ContentId" type="xsd:string" />
  <xsd:attribute name="AttachmentIsTextData" type="xsd:string" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
```

As the WSDL sample indicates. the <Attachment> child element can have the following optional attributes:

- **ContentId.** Uniquely identifies the attachment content. If the attribute is not specified, the Web Services framework creates one for internal processing and to identify the content. This attribute is also used to achieve content reuse within a SOAP message, as described in "Attaching a File to Multiple Records" on page 589.
- **AttachmentIsTextData.** Indicates that the content is plain text and that no Base64 decoding needs to be performed on it. The default value for this attribute is false. This attribute must be specified when text content is being specified.

Specifying File Attachment Content

The following topics describe the different ways in which you can specify the attachment content and how you can attach the same file to multiple records.

Sample SOAP requests are included to illustrate the use of the attributes of the <Attachment> element.

Attaching Binary Data

The following SOAP sample shows how the Base64 encoded attachment file must be embedded within the SOAP document:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://
www.w3.org/2001/XMLSchema">
<soap:Body>
<AccountWS_AccountInsert_Input xmlns="urn:crmondemand/ws/account/10/2004">
<ListOfAccount>
  <Account>
     <Description>This is for missing fields test</Description>
     <Location>Bangalore</Location>
```

```
        <AccountName>Atach_Test1</AccountName>

        <NumberEmployees>300</NumberEmployees>

        <ListOfAttachment>

          <Attachment>

          <FileNameOrURL>Attached</FileNameOrURL>

          <FileExtension>doc</FileExtension>

          <DisplayFileName>A Doc Attachment</DisplayFileName>

            <Attachment ContentId="content_doc">0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAA

........

........

            AAAAAAAAAAAAAAAAAAAAAAAAAA=</Attachment>

        </Attachment>

        </ListOfAttachment>

      </Account>

    </ListOfAccount>

    </AccountWS_AccountInsert_Input>

    </soap:Body>

    </soap:Envelope>
```

The attachment content is specified within the <Attachment> element which has the data type SiebelXmlAttachmentType.

NOTE: It is not required to specify the ContentId attribute.

Attaching a Text File

When attaching a text file (a non-binary file in printable ASCII format), no Base64 encoding is required.

The following SOAP request illustrates how a text file can be attached:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://
www.w3.org/2001/XMLSchema">

<soap:Body>
```

```
<AccountWS_AccountInsert_Input xmlns="urn:crmondemand/ws/account/10/2004">
<ListOfAccount>
<Account>
  <Description>Account with text attachment</Description>
  <Location>Bangalore</Location>
  <AccountName>Attach Text</AccountName>
  <NumberEmployees>300</NumberEmployees>
  <ListOfAttachment>
    <Attachment>
    <FileNameOrURL>Attach Text</FileNameOrURL>
    <FileExtension>txt</FileExtension>
    <DisplayFileName>My Rantings</DisplayFileName>
    <Attachment AttachmentIsTextData="true">
    The quick brown fox jumps over the lazy dog..
    </Attachment>
    </Attachment>
  </ListOfAttachment>
</Account>
</ListOfAccount>
</AccountWS_AccountInsert_Input>
</soap:Body>
</soap:Envelope>
```

The AttachmentIsTextData attribute of the <Attachment> element has the value true to indicate that the content is plain text and that no Base64 encoding or decoding is required to be performed by the Web Services framework.

Attaching a File to Multiple Records

The ContentId attribute of the <Attachment> element is used to uniquely identify attachment content. The ContentID argument is optional and the Web services framework generate ones for its internal processing, if it is not provided.

The ContentID attribute allows you to reuse the content contained in another <Attachment> element so that you can specify the same attachment content for multiple records.

When reusing the content in this way, you only need to specify the ContentID value, and the attachment content must be empty. Values of the ContentID attribute must be unique.

The following SOAP request shows how a file can be attached to two records:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://
www.w3.org/2001/XMLSchema">

<soap:Body>

<AccountWS_AccountInsert_Input xmlns="urn:crmondemand/ws/account/10/2004">

<ListOfAccount>

  <Account>

     <Description>Account for attachment and resuse test</Description>

     <Location>Bangalore</Location>

     <AccountName>Attach Test Reuse</AccountName>

     <NumberEmployees>300</NumberEmployees>

     <ListOfAttachment>

        <Attachment>

           <FileNameOrURL>Attached</FileNameOrURL>

           <FileExtension>doc</FileExtension>

           <DisplayFileName>To be reused</DisplayFileName>

           <Attachment ContentId="reuse">0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/
CQAGAAAAAAAAAAAAAAAACAAAA+AAAAAAAAAAEAAA+gAAAAEAAAD+////AAAAAPYAAAD3AAAA/////////
/////////////////

           .......... The Base64 encoded stream goes here ..........

           AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=</Attachment>

        </Attachment>

     </ListOfAttachment>

  </Account>

  <Account>

     <Description>This copying the attachment</Description>

     <Location>San Mateo</Location>

     <AccountName>Attach Test Copy</AccountName>
```

```
        <NumberEmployees>300</NumberEmployees>
        <ListOfAttachment>
          <Attachment>
            <FileNameOrURL>Attach Reuse</FileNameOrURL>
            <FileExtension>doc</FileExtension>
            <DisplayFileName>Reused</DisplayFileName>
            <Attachment ContentId="reuse"></Attachment>
          </Attachment>
        </ListOfAttachment>
      </Account>
    </ListOfAccount>
    </AccountWS_AccountInsert_Input>
    </soap:Body>
    </soap:Envelope>
```

The SOAP request above shows how you can attach the same file to two different Account records.

- The ContentId value Content_01 is specified for the attachment child of the first Account.
- The same ContentId value is specified for the attachment child of the second account, with the result that the same file is attached to both accounts. Only the ContentId attribute needs to be specified for the second account record.

Retrieving or Querying File Attachments

When performing a query operation to retrieve attachment content, the response must contain Base64 encoded content. This is true even if the attached file is a simple text file. The attachment content must therefore be decoded back from Base64 to the binary or text format in all cases.

No content reuse is supported when performing a query. Even though the same file has been attached to multiple records, the response always generates unique ContentId values and the <Attachment> elements contain the attachment content.

The following SOAP response illustrates a query response for an attachment child:

```
<?xml version="1.0" encoding="UTF-8"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://
www.w3.org/2001/XMLSchema">

<SOAP-ENV:Body>
```

```
<ns:AccountWS_AccountQueryPage_Output mlns:ns="urn:crmondemand/ws/account/10/
2004">

<ns:LastPage>true</ns:LastPage>

<ListOfAccount xmlns="urn:/crmondemand/xml/account">

  <Account>

     <AccountId>1QA2-L9DZG</AccountId>

     <Description>Account with attachment</Description>

     <Location>Toronto</Location>

     <AccountName>Test Account</AccountName>

     <ListOfAttachment>

        <Attachment>

           <DisplayFileName>My sales analysis</DisplayFileName>

           <FileNameOrURL>Analysis</FileNameOrURL>

           <FileExtension>xml</FileExtension>

           <Description></Description>

           <Attachment AttachmentIsTextData="false" Extension="xml" ContentId="1QA2-
RTX3">PD94bWwgdiVVRGLTgiPz48P1NpZWJlbC1Qcm9wZXJ0eS1TZXQgRXNjYXBl

           <Base 64 encoded attachment data ........>

           </Attachment>

        </Attachment>

     </ListOfAttachment>


  </Account>

</ListOfAccount>

</ns:AccountWS_AccountQueryPage_Output>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

There are two additional attributes of the <Attachment> element:

- **EndOfData**. Indicates that the data is the complete content. Always true.
- **TimedOut**. Indicates whether a time-out occurred when obtaining the data. Always false.

Specifying URL Attachments

Web Services also supports the exchange of URL attachments as well as file attachments.

The handling of URL attachments differs from file attachments in the following ways:

- The FileNameOrURL field must specify the actual URL. The URL must begin with one of the following: HTTP, HTTPS, FTP or WWW (case-insensitive).
- The FileNameOrURL and FileExtension fields form a user key for Attachment child objects, however, FileExtension is not a required field for URL attachments and is ignored.

The <Attachment> element is not required to have any content, and any content is ignored if provided.

The following example SOAP request shows a URL attachment.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://
www.w3.org/2001/XMLSchema">

<soap:Body>

<AccountWS_AccountInsert_Input xmlns="urn:crmondemand/ws/account/10/2004">

<ListOfAccount>

  <Account>

     <Description>Account with URL attachment</Description>

     <Location>Bangalore</Location>

     <AccountName>Atach URL</AccountName>

     <NumberEmployees>300</NumberEmployees>

     <ListOfAttachment>

        <Attachment>

           <FileNameOrURL>www.pingpong.com</FileNameOrURL>

        </Attachment>

     </ListOfAttachment>

  </Account>

</ListOfAccount>

</AccountWS_AccountInsert_Input>

</soap:Body>

</soap:Envelope>
```

B Mapping of Display Names to Service API Object Names

A number of the Service API methods, such as GetMapping use elements such as <ObjectName> or <RecordType> to identify the objects on which they operate. However, the object names specified in requests or returned may differ from the name displayed in the UI. For such cases, Table 696 shows the mapping between the default UI display name of objects and their programmatic names as used by Service API methods. For more information about the Service API methods, see "Service API Calls" on page 139.

Table 696. Mapping of Display Object Names to Service API Object Names

UI Name	Service API Name
Account Competitor	AccountCompetitor
Account Partner	AccountPartner
Account Relationship	AccountRelationship
Accreditation Request	AccreditationRequest
Address	CUT Address
Application Team	ApplicationTeam
Allocation	CRMOD_LS_Allocation
Assessment	Sales Assessment Value
Business Plan	CRMODLS_BusinessPlan
Call Product Detail	Call ProdDetail
Call Promotional Item	Call PromItemDrop
Call Sample Dropped	Call SampDrop
Campaign Recipient	ContactCampaign
Certification Request	CertificationRequest
Contact Best Times	ContactBestTimes
Contact Interests	ContactInterest
Contact Relationship	ContactRelationship
Contact State License	CRMOD_LS_ContactLicenses
Course Enrollment	CourseEnrollment
Custom Object 01 Team	CustObj1 Team
Custom Object 02 Team	CustObj2 Team

Table 696. Mapping of Display Object Names to Service API Object Names

UI Name	Service API Name
Custom Object 03 Team	CustObj3 Team
Dealer	Channel Partner
Dealer Territory	Channel Partner Territory
Deal Registration	DealRegistration
Deal Registration Product Revenue	DealRegistrationProductRevenue
Deal Registration Team	DealRegistrationTeam
Exam Registration	ExamRegistration
Financial Account	FinancialAccount
Financial Account Holder	FinancialAccountHolder
Financial Account Holding	FinancialAccountHolding
Financial Plan	FinancialPlan
Financial Product	FinancialProduct
Financial Transaction	FinancialTransaction
Household Team	HouseholdTeam
Inventory Audit Report	CRMODLS_InventoryAuditReport
Inventory Period	CRMODLS_InventoryPeriod
Insurance Property	InsuranceProperty
Involved Party	InvolvedParty
MDF Request	MDFRequest
MDF Request Team	MDFRequestTeam
MedEd Event	MedEdEvent
MedEd Invitee	MedEdInvitee
Message Response	CRMODLS_PCD_MSGRSP
Messaging Plan	CRMOD_LS_MessagingPlan
Messaging Plan Item	CRMOD_LS_MsgPlanItem
Messaging Plan Item Relation	CRMOD_LS_MsgPlnRel
Modification Tracking	CRMODLS_ModificationLog
Objective	CRMODLS_OBJECTIVE
Opportunity Competitor	OpportunityCompetitor
Opportunity Partner	OpportunityPartner
Partner Account	PartnerAccount

Table 696. Mapping of Display Object Names to Service API Object Names

UI Name	Service API Name
Partner Opportunity	PartnerOpportunity
Partner Program	PartnerProgram
Partner Team	PartnerTeam
Partner Type	PartnerType
Plan Account	CRMODLS_BPL_ACNT
Plan Contact	CRMODLS_BPL_CNTCT
Plan Opportunity	CRMODLS_PlanOpportunities
Policy Holder	Policy
Portfolio Team	PortfolioTeam
Portfolio Owner	PortfolioOwners
Price List	PriceList
Price List Line Item	PriceListLineItem
Program Membership	ProgramPartner
Sample Disclaimer	CRMODLS_SIGNDISC
Sample Inventory	CRMODLS_SampleInventory
Sample Lot	CRMODLS_SampleLot
Sample Transaction	CRMOD_LS_Transactions
Signature	CRMODLS_Signature
Special Pricing Request	SPRequest
Special Pricing Request Line Item	SPRequestLineItem
Special Pricing Request Team	SPRequestTeam
Transaction Item	CRMOD_LS_TransactionItems
Vehicle Financial Information	Vehicle FinInfo
Vehicle Sales History	Vehicle SalesHist
Vehicle Service History	Vehicle ServHist

C Web Services On Demand Limiters

This appendix contains the following topics:

- Web Services On Demand Limiters
- Web Services Request Rate on page 601

Web Services On Demand Limiters

The Oracle CRM On Demand's Web Services interface resources can be shared by multiple organizations. Oracle CRM On Demand provides a limiting infrastructure to make sure that some users do not consume a disproportionate share of those resources. These limiters constrain customer organizations' use of server-side resources to equitably share available resources among users and to minimize the possibility of denial-of-service incidents.

Table 697 summarizes the limiters, their default values, units, and error messages. For more information, contact Customer Care.

In addition to the limiters described in this appendix, there are a number of service allotments that promote the equitable use of resources. For more information, see "About Service Allotments" on page 45. Further information about the Web Services Request Rate limiter is provided in "Web Services Request Rate" on page 601.

Table 697. Web Services On Demand Limiters

Limiter Name	Description	Default Value	Units	Error Message
Concurrent Session Limit (Company)	Maximum number of concurrent sessions for a company	10	sessions	Not applicable
Lifetime of Audit records	Maximum number of days Web services utilization records are saved. Records older than 30 days are purged.	30	days	Not applicable

Table 697. Web Services On Demand Limiters

Limiter Name	Description	Default Value	Units	Error Message
Queued Request Limit (Stateful)	Maximum number of requests that a stateful Web services user may have queued waiting for a free OM session	5	sessions	The server is currently busy processing a prior request for you. Please try your request again later.
Queued Request Limit (Stateless)	Maximum number of requests that a stateless Web services user may have queued waiting for a free OM session.	5	sessions	The server is currently busy processing a prior request for you. Please try your request again later.
Web Service Request Rate	Number of Web service messages that can be sent in one second	20	requests /second	The maximum rate of requests was exceeded. Please try again in {0} ms.
Web Service Request Size	Maximum size, in kilobytes, of a Web services request.	28800	KB	Request exceeded the size limit of {0} KB.
Web Service Session Inactivity Time Limit	How long, in seconds, that a Web services session may remain idle before being closed.	600	seconds	Internal Error: Session is not available. Aborting.
Maximum Records Returned	For return messages the maximum number of records returned for each query. A response never returns more than the specified number of records for a parent object in a request.	100 (fixed maximum value)	records	PageSize method argument cannot be greater than 100, specified by the server parameter 'MaximumPageSize'.
Maximum Objects in a Web Services Request	The maximum number of objects that can be sent in a single SOAP request.	20 (fixed maximum value)	objects	Requests containing more than 20 objects are not allowed.

Web Services Request Rate

All integration requests (data exchange requests) in a session are subject to rate limiting. Rate limiting is implemented for the following reasons:

- A user can perform long-running operations on the server that result in complex and long-running queries on the database.
- A user can perform constant operations on the server that constantly use resources.

Rate limiting can alleviate the previous problems to some extent. Oracle CRM On Demand applies a restriction to each session to limit the number of requests for each second that clients can make. The rate limit is set to twenty requests for each second. This is measured as a minimum of 1/20th second wait time between requests.

If the rate limit is exceeded, the following error message is provided to subsequent SOAP requests:

```
The maximum rate of requests was exceeded. Please try again in <waitTime> ms.
```

There are two situations in which this error message is displayed:

- Where the client sends more than 20 requests every second
- Where the client sends requests at a rate faster than one request every 50 milliseconds.

For the second situation, a wait statement is recommended.

Index

B

E

F

G

H

I

K

recipient child object

S